EUROGRAPHICS 2018 / D. Gutierrez and A. Sheffer
(Guest Editors)

Volume 37 (2018), Number 2

Sequences with Low-Discrepancy Blue-Noise 2-D Projections

Hélène Perrier[1] David Coeurjolly[1] Feng Xie[2] Matt Pharr[3] Pat Hanrahan[2] Victor Ostromoukhov[1]

[1]Université de Lyon, CNRS, LIRIS, France [2]Stanford, USA [3] Google, USA

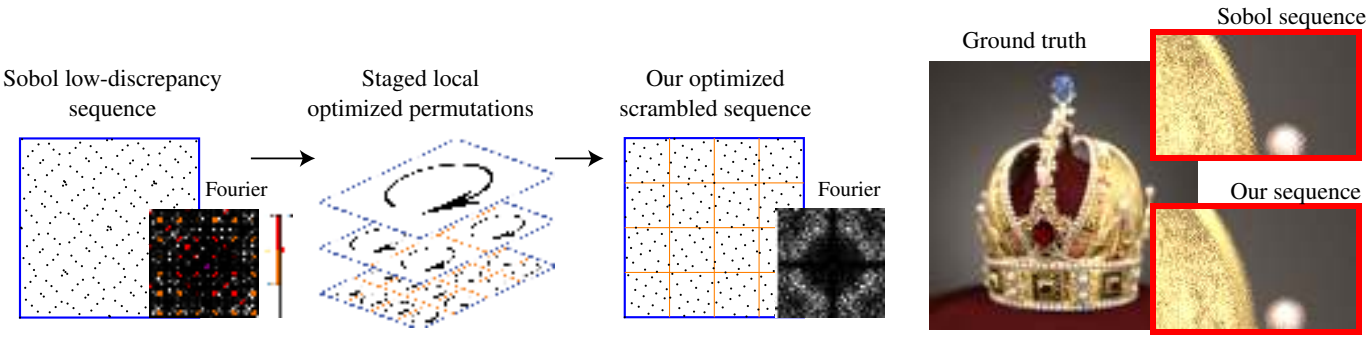

Figure 1: **Left:** our staged per-tile optimized scrambling, applied to a Sobol sequence of sampling points, produces a power spectrum close to Blue Noise. **Right:** Rendering of a challenging scene featuring depth of field and high specularity (jewels). The sampling was done with the Sobol sequence (top) and our sampler (bottom); both use 256 samples per pixel and 3 light bounces. Note the improvement in aliasing when using our method in comparison to the original Sobol sequence.

Abstract

Distributions of samples play a very important role in rendering, affecting variance, bias and aliasing in Monte-Carlo and Quasi-Monte Carlo evaluation of the rendering equation. In this paper, we propose an original sampler which inherits many important features of classical low-discrepancy sequences (LDS): a high degree of uniformity of the achieved distribution of samples, computational efficiency and progressive sampling capability. At the same time, we purposely tailor our sampler in order to improve its spectral characteristics, which in turn play a crucial role in variance reduction, anti-aliasing and improving visual appearance of rendering. Our sampler can efficiently generate sequences of multidimensional points, whose power spectra approach so-called Blue-Noise (BN) spectral property while preserving low discrepancy (LD) in certain 2-D projections.

In our tile-based approach, we perform permutations on subsets of the original Sobol LDS. In a large space of all possible permutations, we select those which better approach the target BN property, using pair-correlation statistics. We pre-calculate such "good" permutations for each possible Sobol pattern, and store them in a lookup table efficiently accessible in runtime. We provide a complete and rigorous proof that such permutations preserve dyadic partitioning and thus the LDS properties of the point set in 2-D projections. Our construction is computationally efficient, has a relatively low memory footprint and supports adaptive sampling. We validate our method by performing spectral/discrepancy/aliasing analysis of the achieved distributions, and provide variance analysis for several target integrands of theoretical and practical interest.

CCS Concepts

•***Computing methodologies*** → ***Rendering;***

1. Motivation

Monte Carlo and Quasi-Monte-Carlo (MCQMC) integration are widely used in various scientific fields to numerically approximate integrals of complex functions that cannot be expressed in closed form. In computer graphics, MCQMC MCQMCthods are fundamental for evaluation of the rendering equation [PJH16]. The choice of the sampling pattern used in MCQMC integration may affect the overall rendering quality in terms of noise and aliasing.

Among various MCQMC methods, low-discrepancy sequences (LDS) play a crucial role [Nie92, Lem09], because of their inherent virtues: guaranteed high degree of uniformity, computational efficiency, easy and compact implementation. Thanks to the well-

known Koksma-Hlawka inequality [Nie92], all LDS have guaranteed upper bound of the variance (and therefore, of the noise), which is the consequence of their high uniformity. Several computer graphics scientists advocated for their use [Shi91, Kel12, PJH16, APC*16]; nowadays LDS are effectively integrated in several rendering engines [PJH16].

On the other hand, according to the spectral approach to evaluate the rendering quality, noise and aliasing are tightly related to the presence of spectral peaks in Fourier spectra of the sampling patterns. This approach, pioneered by Dippé and Wold [DW85], Cook [Coo86], Mitchell [Mit91], has been strengthened through decades of intense research [MF92, ODJ04, KCODL06, DH06, LD08, BSD09, CYC*12, SGBW10, WW11, SHD11, Fat11, dGBOD12, ZHWW12, OG12, HSD13, EMP*12, EAG*14, WPC*14, ANHD17, QCHC17]. It is generally accepted that so-called blue-noise (BN) sampling [Uli88] is a good approach for reducing to reduce aliasing and noise. Researchers have recently devised a closed-form formulation for the variance as a function of the spectral content of the integrand and that of the sampling pattern [Dur11, PSC*15, Ö16].

The very first attempt to build a sampler with both low-discrepancy and blue-noise properties have been recently presented in [APC*16], which shows that such a sampler effectively reduces noise and aliasing in MCQMC integration, while preserving good asymptotic behavior when the number of samples grows. However, this method has several strong limitations. First, their construction is applicable to 2-D point sets only, which limits its application in rendering engines, where higher dimensional samples are required. Second, by contrast with traditional low-discrepancy *sequences*, the method of [APC*16], generates finite *point sets* only; the implementation of progressive sampling is therefore almost impossible. Finally, the method by Ahmed et al. [APC*16] does not support adaptive sampling schemes.

In this paper, we propose a more advanced sampling system, which addresses the limitations enumerated above. Our construction stems from the well-known Sobol sampler [Sob67], which generates low-discrepancy *sequences* by construction and thus allow our sampler to support progressive sampling. As a derivative of Sobol LDS, our construction naturally supports higher dimensions, while preserving almost-BN spectral properties in some 2-D projections, for which specially designed permutations have been pre-computed and stored in lookup tables. At runtime, our hierarchical approach uses lookup tables to obtain a very efficient sampling strategy with good spectral properties. While being a trade-off between memory efficiency and BN quality, we show that with lookup tables of 7kbytes, one can achieve good BN spectra for very large point sets. This scheme of having a s-D sampler with almost-BN 2-D projections is meaningful in rendering, where we need to solve several nested integrals. Designing a consistent s-D sampler with high quality 2-D projections (LD and BN) is a very challenging task. For example, the trivial approach that considers two 2-D LD and BN point sets (*e.g.* from [APC*16]) and a (stochastic) pairing procedure to generate a 4-D point set leads to unsatisfactory discrepancy measurements (see Section 8).

Contributions. In summary, this paper brings the following contributions to the state-of-the art:

(i) our approach scrambles samples from the multidimensional sequences;
(ii) it insures low-discrepancy (and therefore, highly uniform) distributions of sampling in 2-D projections, for which our local optimized scrambling has been applied;
(iii) it allows spectral properties close to that of *blue noise* in such optimized 2-D projections;
(iv) it allows adaptive sampling;
(v) it is relatively fast and has a low memory footprint. Its time complexity is $O(N)$, where N is the number of samples.

2. Related Work

Our method is built upon several existing approaches, namely on Sobol LDS construction, as well as Owen's scrambling technique. We also borrow some concepts from tile-based and optimization-based approaches. In this section we shall enumerate the most relevant prior work.

Low-discrepancy sequences. LDS uniformly fill the sampling domain by calculating multidimensional coordinates of the sampling point from its ordinal number. This number is interpreted as a sequence of digits. By performing manipulations on these digits (radical inversion followed by carefully chosen scrambling operator), the coordinates of the point are calculated. A sampling point with a given ordinal number is stationary, which means that its coordinates remain the same, no matter how many other points fill the sampling domain. This powerful and elegant concept is extremely popular in the MCQMC community, where very high dimensional functions need to be uniformly sampled. Among the most popular LDS are those of van der Corput, Halton, Sobol, Niederreiter, and their generalizations by permutations or digital scrambling [Sob67, Nie92, Lem09]).

Shirley was among the pioneers and the advocates for the use of discrepancy as a quality measure of computer graphics samplers [Shi91]. Keller and his collaborators further promote low-discrepancy sequences in computer graphics [Kel12, GRK12b]. [GHSK08] investigate a LD construction, related to (t,k,s)-nets, under the constraint of maximizing minimal distance between sampling points. They therefore alleviate the problem of arbitrarily close points, which may appear in classical Sobol LDS. Nevertheless, distributions of points as described in [GHSK08] suffer from strong spectral peaks, due to repetitive nature of their construction, which may lead to strong aliasing. Our construction is conceptually quite different; it explicitly addresses the problem of spectral peaks, and therefore it is more appropriate for anti-aliasing.

More recently, Ahmed et al. [APC*16] presented their original construction of two-dimensional low-discrepancy *point sets*, whose limitations have been enumerated in Section 1. It is important to stress a few key differences between the approach presented in [APC*16] and ours. The former is an inherently 2-D construction with no possibility of multidimensional extension, whereas our construction starts from multidimensional Sobol LDS and applies some local permutations on subsets; the resulting sequence is inherently multidimensional. Second, the approach presented in [APC*16] generates *point sets*, not sequences, which prevents progressive sampling, whereas our approach generates sequences

and therefore supports progressive sampling by construction. Finally, the approach presented in [APC*16] does not support adaptive sampling whereas our does support this important feature. In summary, our paper presents a considerable advance with respect to [APC*16] in building *multidimensional sequences* of points with controlled spectral and low-discrepancy properties in 2-D projections.

Scrambled Low-discrepancy sequences. All LDS present excellent asymptotic uniformity; their uniformity at low sample counts may suffer though, especially when the number of dimensions is high. To overcome this limitation, several scrambling techniques have been proposed [Owe95, Mat99, Lem09]. The most powerful of them, Owen's scrambling, consists of building pseudo-random permutation trees, one per dimension, and performing corresponding digit permutations on top of Sobol's construction. Owen's permutation does not affect the low discrepancy of the sequence, while it modifies the inter-sample pair-correlation distance function; the resulting pair-correlation function is close to that of stratified jitter. Note that as it is impossible to evaluate a permutation tree of infinite depth, for practical purposes Owen's scrambling cannot generate a number of samples higher than 2 to the power of the depth of the tree. This breaks the sequentiality property of the input LDS, generating a scrambled LD set instead. Efficient implementation of Owen's scrambling has been explored in [FK01]. In contrast to Owen's construction, which uses random permutations, we precalculate a set of *admissible* permutations which do not affect the discrepancy terms of the distribution. We select only those permutations which improve spectral properties by targeting those of BN.

Tile-based methods. Several methods have been proposed to map elementary samples on tiles of different shapes and thus ensure global uniformity of the achieved distributions. [CSHD03] and [KCODL06] propose to use Wang tiles to break regularity during texture synthesis and in distributions of sampling points. [ODJ04], followed by [Ost07] explore Penrose and polyomino-based tiling systems to build deterministic adaptive sampling systems with good spectral properties. Wachtel et al. [WPC*14] propose a tile-based method that incorporates spectral control over sample distributions. More recently, Ahmed et al. [ANHD17] proposed a 2-D square tile-based sampling method with one sample per tile and controllable Fourier spectra. Our approach is also a tile-based relying on a subdivision system. By contrast with all prior tile-based approaches, we provide LD and BN projections in a high-dimensional setting.

Optimization-based approaches. Thanks to the pioneering work by Dippé and Wold [DW85], Mitchell [Mit91], Cook [Coo86], Shirley [Shi91], the computer graphics community became sensitive to the fact that noise and aliasing are tightly coupled to sampling. They were the first to propose stochastic sampling as an alternative to regular grid-based sampling, arguing that unstructured noise is better perceived than visually perturbing aliasing. Since then, two optimization-based approaches have been developed and presented in numerous papers: (1) on-line optimization [MF92, DH06, LD08, BSD09, BWWM10, EMP*12, CYC*12, SGBW10, SHD11, Fat11, dGBOD12, ZHWW12, OG12, HSD13, RRSG16], and (2) off-line optimization [ODJ04, KCODL06, Ost07, WPC*14, APC*16, ANHD17], where the near-optimal solution is prepared in form of lookup tables, used in runtime. The present work uses the off-line optimization approach.

Figure 2: Illustration of dyadic partitioning, applied on first 16 points of Sobol LDS, dimensions 1 and 2 (all five sub-figures show exactly the same point set). Each partition of size 1×16, 2×8, 4×4, 8×2 and 16×1 contain exactly 1 sample.

Projective Blue-Noise Sampling The idea of having s-D sampling with good blue-noise projections, has been presented in [RRSG16], where the metric of distance between points used in any optimization process has been modified in order to keep track of the projective distance. Although the goal of getting projective BN properties has been achieved that paper does not guarantee uniformity in higher-dimensional space, which could be potentially harmful for integration tasks, especially when the number of samples grows. In the present paper, we achieve simultaneously the goals of getting both projective BN and low-discrepancy (high uniformity in s-D).

3. Our method: Roadmap

Before digging into technical details of our method, let us first provide high-level, simplified and intuitive explanation; rigorous definitions and technical details are provided in Sections 4 and 5. Since our sampler relies on independent scrambling of 2-D projections, in this section we will focus on a single 2-D pair of dimensions. We will detail our higher dimensional construction in Section 7. This section is subdivided into three subsections. In subsection 3.1, we enumerate some properties of Sobol's construction, which are crucial for preserving low-discrepancy property. In our construction, we preserve these important properties, namely that of dyadic partitioning, defined in Section 3.1. In subsection 3.2, we provide a simple description of Owen's scrambling. Finally, in subsection 3.3, we describe the kernel of our contribution: staged optimized permutations, hierarchically applied on subsets of 2-D projections of Sobol point sets.

3.1. Sobol's construction: main properties, useful for our construction

Different properties of the point sets generated using Sobol's construction, have been extensively studied in the literature [Sob67, Nie92, Lem09]. The main reason for large popularity of this construction is based on one key particular property, guaranteed by this construction. A simple and intuitive explanation is often presented in terms of "dyadic partitioning" [Lem09], for dimensions 1 and 2. The entire domain, which contains point sets of size of any integer power p of 2, can be subdivided into rectangular partitions of size 1×2^p, $2 \times 2^{p-1}$, etc. in such a way that each partition contains one sampling point exactly, as illustrated in Figure 2. As this property holds for arbitrarily large point sets, it largely contributes to the high spatial uniformity of Sobol LDS, expressed in terms of discrepancy.

In our construction, we guarantee the same property, even after local reorganization of the point sets, which improves Fourier spectrum of the initial Sobol's construction. Since we bootstrap from the Sobol sequence, we generate samples from primitive polynomials, using a different polynomial for each dimension [Sob67, Nie92, Lem09]. For the first dimension, a pure van der Corput construction is used, which can be considered as Sobol's construction with the trivial primitive polynomial x^0. For other dimensions, different primitive polynomials are used for different dimensions (their enumeration is known as A058947 sequence in Sloan's *On-Line Encyclopedia of Integer Sequences* [Slo17]. In the following, we refer to *Sobol indices* i to define the i-th primitive polynomials.

Strictly speaking, dyadic partitioning holds only for dimensions 1 and 2, which use primitive polynomials x^0 and $x^0 + x^1$. Later, in Section 4 we shall extend the property of dyadic partitioning to more general one of (t,k,s)-net [Nie92, Lem09]. This generalization is applicable to rectangular partitions which contain exactly 2^t sampling points. The lower the "quality factor" t of a (t,k,s)-net, the higher the uniformity of the sampling pattern is. In this generalization, dimensions 1 and 2 have $t = 0$, which corresponds to dyadic partitioning (i.e., each partition contains exactly $2^0 = 1$ sample). Our construction preserves the "quality factor" t of the initial Sobol LDS in two-dimensional projections as demonstrated in the Supplementary Material.

3.2. Owen's Scrambling

Owen's scrambling [Owe95, Mat99] is a key approach to transform a pattern while preserving its low discrepancy properties. As discussed in Section 2, such scrambling performs permutations of point coordinates dimension by dimension using a random Boolean tree.

In 1-D, let us consider a sample with coordinate $\mathbf{x} \in [0,1)$ defined in base 2 with digits x_i (*i.e.* $\mathbf{x} := \sum_i x_i 2^{-i}$). Owen's scrambling goes through all digits and considers digits position by position and swaps the digit x_i if its associated flag is set to 1 in the tree at depth i. Geometrically, swapping a digit x_i swaps all samples with the same $x_0 \dots x_i$ binary prefix, as illustrated in Figure 3. Owen's scrambling in 1-D therefore implies successive permutations between intervals of points of size 2^{-i+1}.

In arbitrary dimensions, such 1-D tree based scrambling is repeated for each dimension with different trees containing random Boolean flags. Since this swapping of samples is done only when a node's flag is set to 1, it is possible to fix the position of some samples by setting some flags to 0. We will rely on this property to ensure the sequentiality of our sampler. Please note that Owen's permutation trees are global. Among all possible trees, some may lead to point sets having interesting spectral properties but efficiently exploring the space of trees is an open problem. Indeed, for n samples in s-D, $s^{2^{n-1}}$ different trees exist. Furthermore, blue-noise sampler usually implies globally and locally correlated samples. A global Owen's tree in a higher dimension does not allow the needed local control. For all these reasons, we find instead near-optimal local permutation trees for the blocks of 16 or 64 samples, as we shall explain in the next Section.

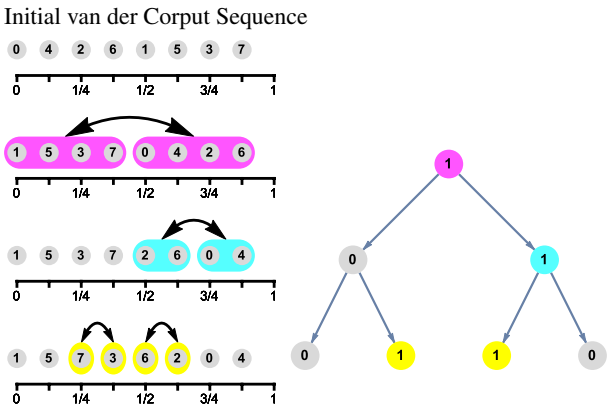

Figure 3: Illustration of 1-D Owen's scrambling, applied on first 8 members of van der Corput sequence. Blocks of points of initial sequences (top-left) are hierarchically swapped according to the flags of the permutation tree (right), where 1 means that we swap blocks of appropriate size, whereas 0 means that we do not swap blocks.

3.3. Our Basic Construction

Our basic construction of multidimensional sequences with low-discrepancy projections and improved spectral properties is illustrated in Figure 4. In this illustration, blocks of 16 samples are used and only one pair of dimensions is shown. Note that similar constructions with blocks of 64 and 256 samples, as well as other pairs of dimensions can be equally used as will be discussed in Sections 4 and 5.

Our construction can be conceptually considered as a set of staged local permutations of blocks of 16 samples of the initial Sobol LDS (the leftmost column in Figure 4). For the first 16 points of the initial set and for dimensions x and y (the top row in Figure 4), a good pair of Owen's permutation trees according to a given blue-noise criterion (see Section 5.2), can be found. Such a pair is chosen to produce a more even distribution of sampling points (the rightmost sub-image of the top row in Figure 4). Note that this permutation preserves the t factor of the initial set of 16 samples [Owe95] (see supplementary material). We call this permutation π^0.

For the set of 16^2 samples (the middle row in Figure 4), we first apply the same permutation π^0 on the entire point set, then find a set of local permutations π_i^1, applied on 16 subsets of 16 samples each, in order to get even distribution of all 256 samples, as shown in the rightmost sub-images of the middle row in Figure 4. Note that the second stage of permutations π_i^1 does not move first 16 samples, performed by the first stage π^0 – compare large colored dots which mark the 'pivots' of each block of 16 samples (we call them pivots, because they are immovable samples w.r.t. previous stage).

Similarly, the set of 16^3 samples (the lower row in Figure 4) undergoes 3 stages of permutations, in order to keep first 16 and 256 well distributed, and achieve good distribution of all 16^3 samples. This process can be continued to higher orders k of 16^k samples. In Sections 4 and 5 we provide all required technical details, including pseudo-code of the algorithm. The formal proof for the preservation of k of a (t,k,s)-net in two-dimensional projections is given in the

supplementary material. There, we also discuss implementation issues, such as complexity, computational efficiency and parallelism (leading to a GPU implementation).

4. Our Method: Technical Presentation

As sketched in the roadmap, our sampler relies on a hierarchical construction using Sobol samples that are locally optimized using specific Owen's permutation trees We first define the regular subdivision of a s-dimensional domain $[0,1)^s$ that carries our samples. Such system is parametrized by a subdivision factor $K := 2^n$ for a given $n > 0$ (*e.g.* K can be 16, 64 or 256 as discussed in Section 3.3). At a level of subdivision λ, each dimension is subdivided K^λ times, and thus the domain is partitioned into $K^{s\lambda}$ squared tiles. This factor K have noticeable impact on the spectral quality, the discrepancy, and the memory footprint of our scrambling, as we'll show below. The i^{th} square tile at a level λ is denoted T_i^λ (see Fig. 4). We also denote $\mathcal{P}^\lambda$ our permuted set from a subdivision at a level λ. $\mathcal{P}^\lambda$ contains $K^{s(\lambda+1)}$ samples, distributed into $K^{s\lambda}$ tiles, so that we have K^s points in each tile T_i^λ. The hierarchical construction ensures that $\mathcal{P}^{\lambda-1} \subset \mathcal{P}^\lambda$ which implies a sequential definition of the sampler, which implies that the samples in $\mathcal{P}^\lambda$ are indexedto be a $(t, \lambda+1, s)$-net in base K^s for each level λ.

In Section 4.1, we first describe the local permutation steps in the hierarchical construction (columns (b), (d) and (f), from $\mathcal{Q}^\lambda$ to $\mathcal{P}^\lambda$, in Figure 4). In Section 4.2, we describe a technique to efficiently implement the hierarchical construction (finding $\mathcal{Q}^\lambda$) in a way that $\mathcal{P}^\lambda$ will only depend on $\mathcal{P}^{\lambda-1}$ and $\mathcal{S}^\lambda$ (the Sobol samples at level λ).

4.1. Pattern local optimization

Let us first consider that we have a point set $\mathcal{Q}^\lambda$ with the following property: each tile T_i^λ contains K^s samples, with a single pivot sample (see Fig.4). We further assume that $\mathcal{Q}^\lambda$ is a $(t, \lambda+1, s)$-net in base K^s, and that each pattern inside each tile T_i^λ of $\mathcal{Q}^\lambda$ is a (t, ns, s)-net in base 2.

The point set $\mathcal{P}^\lambda$ is obtained from $\mathcal{Q}^\lambda$ by locally permuting, for each tile T_i^λ, the pattern $\{\mathbf{q}\}_i^\lambda := \mathcal{Q}^\lambda \cap T_i^\lambda$ using a permutation π_r^λ with the following properties:

(i) π_i^λ does not move the pivot sample;
(ii) π_i^λ is (t, ns, s)-net preserving in base 2.
(iii) π_i^λ does not change the 1-D projections of the input pattern.

We say that $\Pi := \{\pi_i^\lambda\}$ is an *admissible set of local permutations* if each π_i^λ fulfills the above mentioned constraints. For any given admissible Π, we prove in the Supplementary material (Lemma 2) that Π preserves the discrepancy properties of $\mathcal{Q}^\lambda$ in 2-D . More formally:

Proposition 1 For $s = 2$, given an admissible set of permutations Π, if $\mathcal{Q}^\lambda$ is a $(t, \lambda+1, 2)$-net in base K^2 then $\mathcal{P}^\lambda$ is a $(t, \lambda+1, 2)$-net in base K^2.

Note that such permutations can lead to point set with various spectral properties. We can control the spectral properties of $\mathcal{P}^\lambda$ by choosing, among the set Π, an *ad-hoc* permutation $\tilde{\pi}_i^\lambda$ for each pattern $\{\mathbf{q}\}_i^\lambda$. In Section 5, we describe in details how such ad-hoc permutation are computed. We create the point set $\mathcal{P}^\lambda$ from such permutations and from a point set $\mathcal{Q}^\lambda$ by independently permuting the samples in each tile T_i^λ.

4.2. From $\mathcal{P}^{\lambda-1}$ to $\mathcal{P}^\lambda$

In Figure 4, the third level of subdivision is obtained applying successive permutations on samples at different subdivision levels, starting from the first one. In this section, we present a more compact construction allowing us to obtain $\mathcal{Q}^\lambda$ from the three sets: $\mathcal{P}^{\lambda-1}$, $\mathcal{S}^{\lambda-1}$ and $\mathcal{S}^\lambda$. We remind the reader that in each of those sets, the samples are indexed and that $\mathcal{S}^\lambda$ and $\mathcal{S}^{\lambda-1}$ are defined implicitly and not stored in memory. We first define a set of vectors $\mathcal{V}^{\lambda-1}$ as

$$\mathcal{V}^{\lambda-1} := \{\mathbf{so}^{(i)} \oplus \mathbf{p}^{(i)}\}, \tag{1}$$

where $\mathbf{so}^{(i)}$ is the i^{th} sample of $\mathcal{S}^{\lambda-1}$, $\mathbf{p}^{(i)}$ is the i^{th} sample of $\mathcal{P}^{\lambda-1}$, and $\oplus$ the bit-wise `xor` operator on binary representations of point coordinates (Figure 5). By definition, $|\mathcal{S}^{\lambda-1}| = |\mathcal{V}^{\lambda-1}|$, we extend the `xor` operator to indexed point sets of same size as

$$\mathcal{S}^{\lambda-1} \oplus \mathcal{V}^{\lambda-1} := \{\mathbf{so}^{(i)} \oplus \mathbf{v}^{(i)}, \forall i \text{ with } \mathbf{so}^{(i)} \in \mathcal{S}^{\lambda-1}; \mathbf{v}^{(i)} \in \mathcal{V}^{\lambda-1}\}.$$

It is worth noting that

$$\begin{aligned} \mathcal{S}^{\lambda-1} \oplus \mathcal{V}^{\lambda-1} &= \{\mathbf{so}^{(i)} \oplus \mathbf{so}^{(i)} \oplus \mathbf{p}^{(i)}\} \\ &= \{\mathbf{p}^{(i)}\} \\ &= \mathcal{P}^{\lambda-1}. \end{aligned} \tag{2}$$

Therefore, the set $\mathcal{V}^{\lambda-1}$ is another way to describe all the permutations that were applied on $\mathcal{S}^{\lambda-1}$ to create $\mathcal{P}^{\lambda-1}$.

Thanks to Sobol sequence properties, each tile T_i^λ contains a single sample from $\mathcal{S}^{\lambda-1}$, denoted $\mathbf{so}_\mathbf{r}^{(i)}$, where i is the index of this sample in the Sobol sequence. If we assign each vector $\mathbf{v} := \mathbf{so}_i^{(i)} \oplus \mathbf{p}^{(i)}$ to the tile T_i^λ that contains $\mathbf{so}_i^{(i)}$, we are guaranteed to have a single $\mathbf{v} \in \mathcal{V}^{\lambda-1}$ in each tile, denoted $\mathbf{v}_\mathbf{r}^\lambda$ (see Figure 6). We can now formally define $\mathcal{Q}^\lambda$ as

$$\mathcal{Q}^\lambda := \left\{\mathbf{so}_i^{(i)} \oplus \mathbf{v}_i^\lambda, \forall T_i^\lambda\right\}. \tag{3}$$

We demonstrate in the Supplementary material (Lemma 3) that this permutation preserves the discrepancy properties of $\mathcal{S}^\lambda$. More formally:

Proposition 2 For $s = 2$, if $\mathcal{S}^\lambda$ is a $(t, 2n(\lambda+1), 2)$-net in base 2, $\mathcal{Q}^\lambda$ is a $(t, \lambda+1, 2)$-net in base K^2, $K := 2^n$.

Our whole permutation therefore consists in first building the point set $\mathcal{Q}^\lambda$, and then in applying the local permutations onto $\mathcal{Q}^\lambda$ to create $\mathcal{P}^\lambda$. The whole pipeline is illustrated in Figure 7.

We further demonstrate in the Supplementary material (Lemma 5) that $\mathcal{P}^{\lambda-1} \subset \mathcal{P}^\lambda$. This property follows from the facts that $\mathcal{S}^{\lambda-1} \subset \mathcal{S}^\lambda$, $\mathcal{S}^{\lambda-1} \oplus \mathcal{V}^{\lambda-1} = \mathcal{P}^{\lambda-1}$ using (2), and from LDS properties of Sobol.

In conclusion, using the input Sobol sequence and an admissible

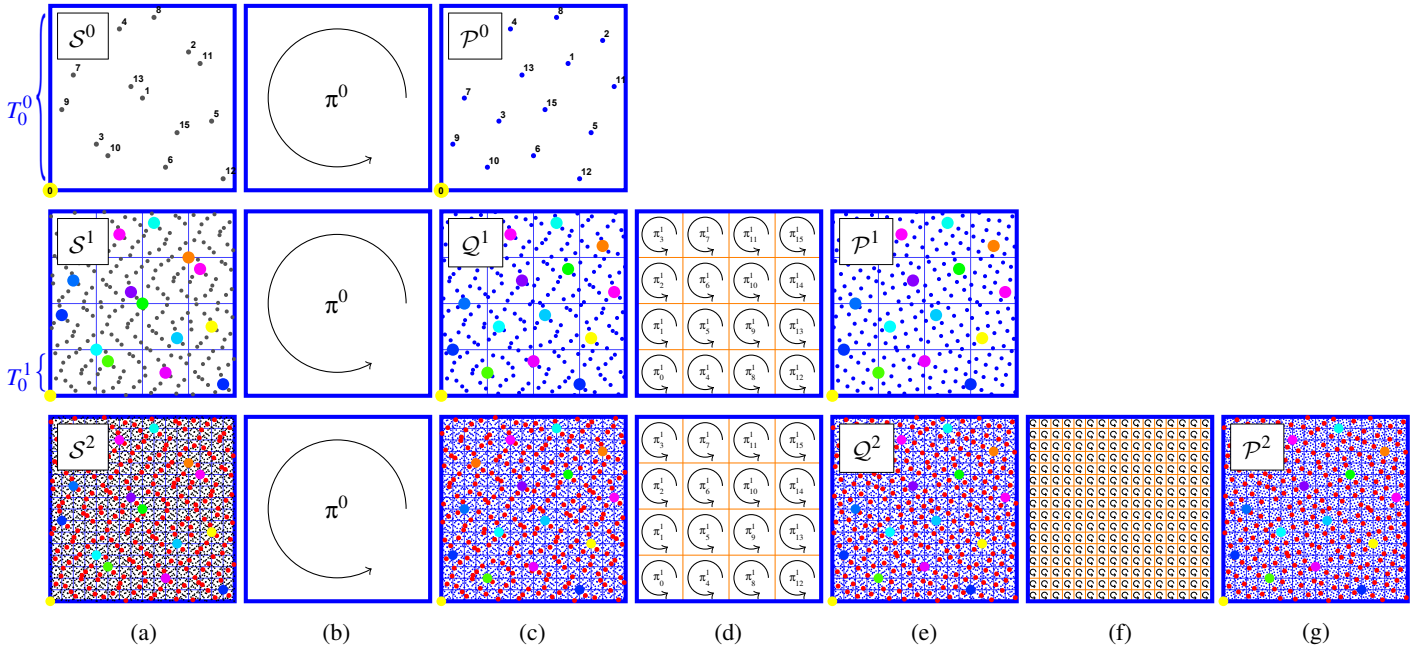

Figure 4: Left column: original Sobol point sets (upper row: 16^1 points; middle row: 16^2 points; lower row: 16^3 points), generated using Sobol indices 3 and 7 for dimensions x and y. Note that all square partitions delimited by thin blue lines contain exactly 16 points each; this remains true for any power p of 16^p sampling points. Upper row: the permutation π^0 makes first 16 points of the Sobol sequence more evenly distributed over the domain $[0,1)^2$. Middle row: the permutation π^0 applied to 16^2 points of the Sobol sequence keeps the first 16 points well distributed (large colored dots in the middle sub-figure, middle row), whereas the remaining 240 points (smaller blue dots) are unevenly distributed. Staged π_i^1 permutations make all first 256 points of the sequence more evenly distributed (rightmost sub-image, middle row). Pivot points (the first point of each group of 16 points, according to Sobol's numbering) are shown as colored larger dots for first 16^2 points and larger red dots for first 16^3 points of the sequence. Note that after application π^0 on 16^2 points, pivots of permuted blocks of 16 points are identical to distribution of first 16 points after application of π^0. Lower row: similarly, 3 levels of permutations are needed in order to obtain even distribution of first 16^3 points of the sequence. Note that after application π^0 on 16^3 points, first 16 points are evenly distributed, as in upper row, whereas staged π_i^1 applied on 16^3 points keep first 16^2 points evenly distributed, as in the middle row. Only three levels of staged permutations are required to achieve even distribution of all 16^3 points (rightmost sub-image, lower row). Please zoom in the sub-figures of the lower row to see fine details.

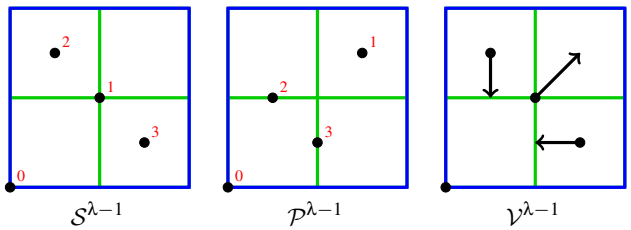

Figure 5: This figure illustrates the first step in computing $\mathcal{Q}^\lambda$ directly from $\mathcal{S}^{\lambda-1}$ and $\mathcal{P}^{\lambda-1}$. This step consists in xoring the samples from $\mathcal{S}^{\lambda-1}$ with the samples from $\mathcal{P}^{\lambda-1}$. This gives us a set of xoring vectors $\mathcal{V}^{\lambda-1}$, that we will apply later on $\mathcal{P}^\lambda$ to create the new set $\mathcal{Q}^\lambda$ (see Figure 6). It is illustrated for $s=2, K=2$ and $\lambda=2$.

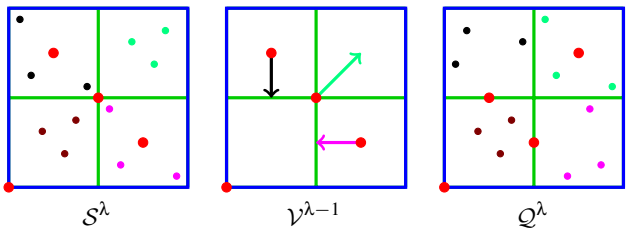

Figure 6: This figure illustrates the second step in computing $\mathcal{Q}^\lambda$ directly from $\mathcal{S}^{\lambda-1}$ and $\mathcal{P}^{\lambda-1}$. This second (and last) step consists in xoring the samples from $\mathcal{S}^\lambda$ with the set of vectors $\mathcal{V}^{\lambda-1}$ computed earlier (see Figure 5). This scrambles the samples inside $\mathcal{S}^\lambda$ to create a new set $\mathcal{Q}^\lambda$ such that $\mathcal{P}^{\lambda-1} \subset \mathcal{Q}^\lambda$. It is illustrated for $s=2, K=2$ and $\lambda=2$.

set of permutations Π, we finally get a point set $\mathcal{P}^\lambda$ that is incremental, and has LD properties in dimension 2 (combining Propositions 1 and 2). Note that our scrambling has a good discrepancy but is not LD in dimensions higher than 2. This is discussed in Section 8. In the next section, we will describe how to construct an admissible set Π and how to precompute from this set the admissible permutations allowing us to control the spatial distribution of points in $\mathcal{P}^\lambda$.

5. Local permutations with spectral control

Our construction relies on admissible permutations $\{\pi_i^\lambda\}$. Among those permutations, we choose the *ad-hoc* ones for a given pattern, meaning the permutations leading to a Fourier spectrum as close as possible to a Blue Noise spectrum. In this section, we will define a construction giving admissible permutations, and describe how we identify the *ad-hoc* permutations for a pattern within this set.

5.1. A set of admissible permutations Π

If we consider any static Boolean tree of depth m per dimension s, Owen's scrambling is known to be (t,m,s)-net preserving in base 2 (property (ii)). However, it does not preserve the 1-D projections of samples (property (iii)) when coordinates are expressed on more than m digits, or when there are fewer than 2^m samples. To alleviate this, we apply the scrambling on the first m bits and exchange the trailing ones in order to preserve the 1-D projections (see Supplementary Section 2). As a consequence, for each tile T_i^λ, any such tree with $m = \log(K)s$ induces a permutation π_r^λ satisfying (ii) and (iii). The number of flags in such trees is thus $2^m - 1 = K^s - 1$. To satisfy (i), (ii) and (iii), we thus have m flags set to 0, while the remaining $K^s - 1 - m$ flags are free.

5.2. Identifying *ad-hoc* permutations

Among all possible admissible permutations of the pattern $\mathcal{Q}^\lambda$ to obtain $\mathcal{P}^\lambda$, we look for transformations that optimize the spectral content of the point set, *e.g.* increasing the minimum distance between points or obtaining a pattern that is as close as possible to a blue noise one. To characterize a pattern as a realization of a stationary and ergodic stochastic point process, we consider its pair correlation functions (PCF) which can be related to the spectral content of the point set [IPSS08]. To evaluate the quality of a pattern with respect to a blue noise one, we compute the l_2 norm of the two PCFs as discussed in [OG12].

In dimension s for a given K, the point set $\{\mathbf{q}\}_i^\lambda$ (defined as $\mathcal{Q}^\lambda \cap T_i^\lambda$) has K^s samples, we have $K^{s\lambda}$ such tiles. For each tile, the permutation is encoded with s Boolean trees of depth m with $(K^s - 1 - m)$ free flags. As an example, for $s = 2$ and $K = 4$, we have 2048^2 different admissible permutations to explore for a given pattern in T_i^λ. For each permutation, we evaluate its quality using its PCF and store permutations leading to a good pattern in a lookup table. Such a lookup table is indexed by the pattern $\{\mathbf{q}\}_i^\lambda$ and stores a set of good permutations with respect to a blue noise target. Note that, when implemented on the GPU, the exhaustive search of an optimized pattern for K=4 can be done in less than 1 second.

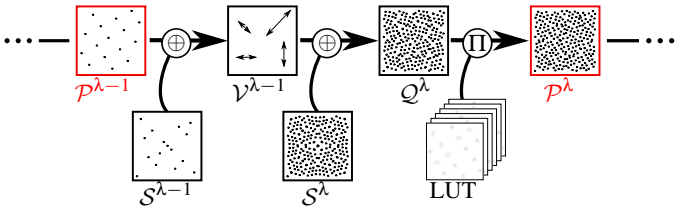

Figure 7: This figure is a simple illustration of the pipeline of our scrambling system. We compute a set $\mathcal{V}^{\lambda-1}$ of vectors, that we will apply of $\mathcal{S}^\lambda$ to create a temporary set $\mathcal{Q}^\lambda$, such that $\mathcal{P}^{\lambda-1} \subset \mathcal{Q}^\lambda$. Then, we scramble this set locally, reading precomputed permutations from a LUT, to create a set $\mathcal{P}^\lambda$ with the targeted Blue Noise property and such that $\mathcal{P}^{\lambda-1} \subset \mathcal{P}^\lambda$.

For $K = 4$, an exhaustive search is possible. For $K > 4$, we optimize exhaustively the first 16 bits of the permutation tree and fill the underlying digits at random. As our GPU implementation performs this search in 1s, we can brute force the last digits to ensure that, even when randomly chosen, they do not harm the quality of the pattern. This whole process takes up to 2s.

Once we have computed the set of *ad-hoc* permutations $\{\tilde{\pi}_i^\lambda\}$ optimized regarding a specific pattern $\{\mathbf{q}\}_i^\lambda$, we store them in the lookup table, indexed by the pattern $\{\mathbf{q}\}_i^\lambda$. Note that multiple *ad-hoc* permutations can be stored for the pattern $\{\mathbf{q}\}_i^\lambda$. This allows us to randomize the scrambler by randomly picking a good permutation (see Algorithm 1). The number of entries in the LUT is the number of distinct $\{\mathbf{q}\}_i^\lambda$ patterns. Due to the regularity of the Sobol sequence, this number is limited and depends on the value K. Discussions on the LUT size as a function of K are included in Appendix A.

6. Our 2-D Sampler Overview

Algorithm 1 sums up the whole pipeline of our permutation (illustrated in Figure 7). We create $\mathcal{P}^\lambda$ by first creating a point set $\mathcal{Q}^\lambda$ by xoring the samples of $\mathcal{S}^{\lambda-1}$ and the samples of $\mathcal{P}^{\lambda-1}$, and reapplying the obtained set of vectors to the samples of $\mathcal{S}^\lambda$. Then, for each tile T_i^λ, we read from the LUT what is the *ad-hoc* permutation for the pattern $\{\mathbf{q}\}_i^\lambda$ and apply it.

7. Using Our Sampler in Higher Dimensions

In many rendering applications, higher dimensional samples may be required. The permutation described above is only defined in 2-D . However, even though it only performs operations that are perfectly defined in s-D, our construction cannot be straightforwardly extended to s-D. We instead extend this approach to s-dimensions by relying on 2-D projections (see Figure 8). In short, we can use a s-D Sobol sequence, optimize some 2-D projections with respect to pre-computed lookup tables and perform an *on-the-fly* random permutation of the remaining dimensions. In Supplementary material (Section 3), we detail our adaptation of Owen's original method to scramble higher dimensional sequential point sets whose selected set of projections have been optimized by our sampler.

Algorithm 1: Refining $\mathcal{P}^{\lambda-1}$ to $\mathcal{P}^{\lambda}$

input : Point Set $\mathcal{P}^{\lambda-1}$ at level $\lambda-1$ and the lookup table $LUT(\cdot)$.
output: Point Set $\mathcal{P}^{\lambda}$.

// Construct $\mathcal{Q}^{\lambda}$ from $\mathcal{S}^{\lambda-1}$, $\mathcal{S}^{\lambda}$, and $\mathcal{P}^{\lambda-1}$
forall *tiles* T_i^{λ} **do**
  $\mathbf{so}^{(i)} \leftarrow \mathcal{S}^{\lambda-1} \cap T_i^{\lambda}$;
  $\mathbf{p}^{(i)} \leftarrow \mathcal{P}^{\lambda-1} \cap T_i^{\lambda}$;
  $\mathbf{v}_{\mathbf{r}}^{\lambda} \leftarrow \mathbf{so}^{(i)} \oplus \mathbf{p}^{(i)}$;
  forall *points* $\mathbf{so} \in \mathcal{S}^{\lambda} \cap T_i^{\lambda}$ **do**
    $\mathcal{Q}^{\lambda} \leftarrow \mathcal{Q}^{\lambda} \cup (\mathbf{so} \oplus \mathbf{v}_i^{\lambda})$;
// Apply local permutations
forall *tiles* T_i^{λ} **do**
  $\{\mathbf{q}\}_i^{\lambda} \leftarrow \mathcal{Q}^{\lambda} \cap T_i^{\lambda}$;
  $\tilde{\pi}_i^{\lambda}$ is randomly selected from LUT$(\{\mathbf{q}\}_i^{\lambda})$;
  $\{\mathbf{p}\}_i^{\lambda} \leftarrow \pi_i^{\lambda}(\mathcal{Q}^{\lambda})$;
  $\mathcal{P}^{\lambda} \leftarrow \mathcal{P}^{\lambda} \cup \{\mathbf{p}\}_i^{\lambda}$;
return $\mathcal{P}^{\lambda}$

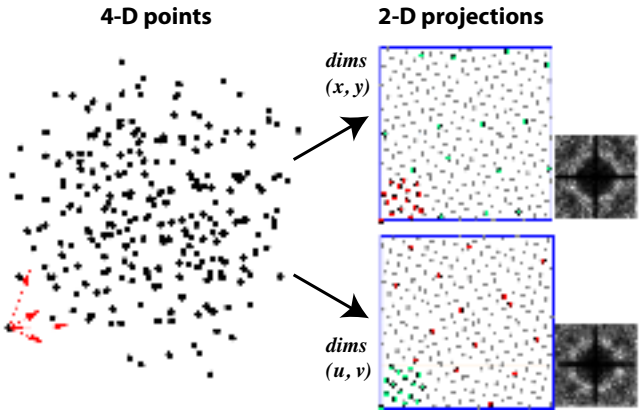

Figure 8: 4-D point set generated with our system, together with two 2-D projections and their Fourier power spectra. Note that subsets of 16 points that occupy the same strata in one projection (red subsets in $\{(x,y)\}$ projections and green subsets in $\{(u,v)\}$ projections), occupy different strata in the other projection.

8. Results

In this section, we perform a series of experiments to demonstrate the discrepancy and spectral properties of our scrambling. Both those properties are known to lead to good results in Monte-Carlo integration thanks to the Koksma-Hlawka's [Hla61] theorem or the variance analysis for stochastic samplers as in [SK13, PSC*15]. In Section 8.1, we first present how we improve the 2-D spectral properties. Then, in Section 8.2, the discrepancy of our 2-D permutation, along with the discrepancy of the final 4-D set we obtain from independently scrambling the 2-D projections of a 4-D Sobol sequence. In Section 8.3, we present integration results, along with renderings in Section 8.4 , and finally demonstrate in Section 8.5 the adaptivity of our scrambling.

In dimension 2, we compare to Blue Noise samplers BNOT [dGBOD12] and LDBN [APC*16]. The later is the only sampler known to us that combines both a high spectral quality and a low discrepancy property (but limited to 2-D and not progressive). We also compare to classical stochastic samplers such as jittered and white noise. We also consider Sobol sequences [Sob67] and Owen's scrambling of such point sets as a baseline for the discrepancy evaluation. In higher dimensions (dimension 4 for numerical comparisons) the Sobol sequence and its scrambling are also considered. For anti-aliasing evaluation, we have compared to Projective Blue Noise (PBN) [RRSG16] that also aims to control projections. Reinert *et al.* have defined two variants of PBN; the first one relies on a Dart Throwing (DT) strategy and the second one on a Lloyd's relaxation approach. Because Lloyd's approach is very algorithmically challenging in higher dimensions, our experiments only consider the DT variant. In 4-D, we have also considered a naive 2-D +2-D variant of LDBN with two 2-D LDBN samples paired using a random permutation to construct a 4-D point set. This variant highlights that beside very good projections with some LD properties, the strength of our sampler relies on its natural extension to higher dimensions.

The properties of all those samplers are summed up in Table 1.

8.1. Spectral properties

In this section, we highlight the 2-D spatial and spectral properties of the various samplers. To do so, we rely on Fourier spectra, as well and their equivalent radial average power spectra. It has been shown by Pilleboue [PSC*15] that for a sampling pattern to be optimal in terms of variance, it should have as little energy in low frequencies as possible. Also, a radial spectrum presenting peaks leads to aliasing in renderings. To illustrate this, we also show results from a zone plate test. This test reconstructs the function $\sin(x^2+y^2)$ that present a wide range of frequencies. Zone plate reconstructed images are thus very prone to aliasing and noise which makes them useful tools to detect it. To present the performances of our exhaustive and genetic optimizers, we present the spectra we obtain with a subdivision factor $K=4$ and $K=8$.

The results of all those tests for our sampler and various others are presented in Figure 9. It can be seen that Sobol's spectrum presents many peaks, that translates into aliasing in the zone plate. Note that those peaks are efficiently removed by our scrambling, and that our spectrum also presents fewer low frequencies than

Sampler	Discrepancy	Sequence	Rendering (Aliasing+Noise)	#Dimensions	Adaptivity	Speed (secs) 10^6 samples
BNOT [dGBOD12]	⋆⋆⋆	n/a	⋆⋆⋆⋆⋆	⋆	⋆⋆⋆⋆⋆	> 1000 (n/a)
Sobol [Sob67]	⋆⋆⋆⋆⋆	⋆⋆⋆⋆⋆	⋆	⋆⋆⋆⋆⋆	⋆⋆⋆	2.3
Halton [Hal60]	⋆⋆⋆⋆	⋆⋆⋆⋆⋆	⋆⋆	⋆⋆⋆⋆⋆	⋆⋆⋆	0.66
Jittered	⋆⋆	n/a	⋆⋆⋆	⋆⋆⋆	⋆⋆⋆⋆	0.02
White noise	⋆	n/a	⋆	⋆⋆⋆⋆⋆	⋆⋆	0.02
Sobol + Owen's scrambling [Owe95]	⋆⋆⋆⋆⋆	⋆⋆⋆⋆	⋆⋆⋆	⋆⋆⋆⋆⋆	n/a	4.8
LDBN [APC*16]	⋆⋆⋆⋆⋆	n/a	⋆⋆⋆⋆⋆	⋆	n/a	0.09
PBN [RRSG16]	⋆⋆	n/a	⋆⋆⋆⋆	⋆⋆⋆	⋆⋆⋆	> 1000 (n/a)
Ours	⋆⋆⋆⋆⋆	⋆⋆⋆⋆	⋆⋆⋆⋆	⋆⋆⋆⋆	⋆⋆⋆⋆	7.5

Adaptive Sobol and Halton are described in [GRK12a]
Jittered is subject of curse of dimensionality for dimensions > 4
The speed was measured on a CPU Intel Core i5-4440 @ 3.10 GHz

Table 1: Summary of properties of various samplers we compare ourselves to. This table is not exhaustive and focuses only on the properties we are looking for in our scrambling. The "Discrepancy" column refers to how close the sampler is to be a low discrepancy sampler. The "Sequence" column refers to whether a point set of size N can be enriched into a point set of size N' with $N' > N$. The "Rendering" evaluates the performances of a sampler in terms of the visual quality of a rendered image using this sampler (taking into account the presence of aliasing and of noise). "#Dimensions" reflects algorithm's capacity to handle dimensions greater than 2. "Adaptivity" refers to algorithm's capacity to add/remove samples locally.

Owen. Theoretically, the higher the K factor for our sampler, the better the spectrum should be. However, as it is impossible to optimize permutations for $K = 8$ as efficiently as for $K = 4$, the spectra are in fact similar.

8.2. Discrepancy

We now evaluate the discrepancy quality of the samplers. A sampler is said to be low discrepancy if its discrepancy decreases as the number N of samples increases at a rate of $O\left(\frac{\log(N)^{s-1}}{N}\right)$ [Lem09] For sake of consistency through dimensions, we evaluate the generalized l_2 discrepancy of samplers in Figures 10 and 11 [Hic98].

As expected, in Figure 10, our scrambling has discrepancy values similar to Sobol sequences [Sob67], Halton sequences [Hal60] or Owen's scrambling [Owe95]. We can note however that our sampler guarantees a low discrepancy for sets in base K^s, when Sobol and Owen guarantee the low discrepancy for sets in base 2. In Figure 11, as we are in 4-D, we no longer guarantee the low discrepancy property. Contrary to Owen, we have performed two 2-D scramblings instead of one 4-D scrambling which partly breaks the discrepancy. Thus, our discrepancy increases slightly. Note that applying two 2-D scrambling over a 4-D Sobol sequence is different than a naive shuffling of low discrepancy 2-D projections. For comparison, we naively recombined point sets from LDBN [APC*16] (LDBN 2-D +2-D) and we can see that this recombination has high discrepancy values. On the other hand, regarding our sampler, even though its discrepancy is increased compared to Sobol, we can note that it still performs better than most s-D samplers.

8.3. Integration

We present the resulting variance in integration of various samplers. In each test, we computed this variance by integrating a given 2-D or 4-D function with several realizations of the point sets, using Cranley-Patterson random shifting to randomize deterministic samplers [CP76]. Our first function is a 2-D disk of radius 0.25 (Figure 12), a test known to be a difficult integration set [PSC*15]. We also integrated a 2-D gray scale HDR Image of size 1600x1600, from the sIBL archive (Figure 13), and a 4-D ball of radius 0.25 (Figure 14).

On each graph, we can see that our sampler has a variance similar to that of low-discrepancy samplers such as Sobol, Halton or Owen's scrambling (but which present less noise or aliasing in rendering). Also, since it's very efficient, we can use it to generate up to 10^6 samples, which is not the case for samplers such as BNOT. In 2-D , we perform similarly to LDBN, but in 4-D, the naive recombination of LDBN has a variance similar to white noise, which stresses that independently scrambling each 2-D projection is not similar to naively combining 2-D projections.

8.4. Renderings

We have implemented our sampler for integration into PBRT [PJH16]. Our PBRT sampler uses our scrambling technique in 4-D or 6-D, complemented with our Owen's scrambling for higher dimensions (Sect. 7). Figures 15, 16, and 17 illustrate rendering comparisons between our sampling technique versus standard Sobol [Sob67], Halton [Hal60] and stratified sampling. Mean square errors (MSE) are computed compared to ground truth renders.

In Figure 15, we start by using our sampler solely for 4 dimensions. This scene uses 8 dimensional samples: 2 for the image plane, 2 for the point on the light, 2 for the light scattering, and 2 for the object's material. We rendered it using various samplers to sample the image plane and the light source, with all other dimensions sampled using the Sobol sequence. Since the Sobol sequence is deterministic, differences between the images are only due to the difference in sampling for the image and light. It can be seen that our sampler has a MSE similar to Sobol, but no longer presents aliasing in the shadow.

In Figure 16, we render a complex scene with various samplers.

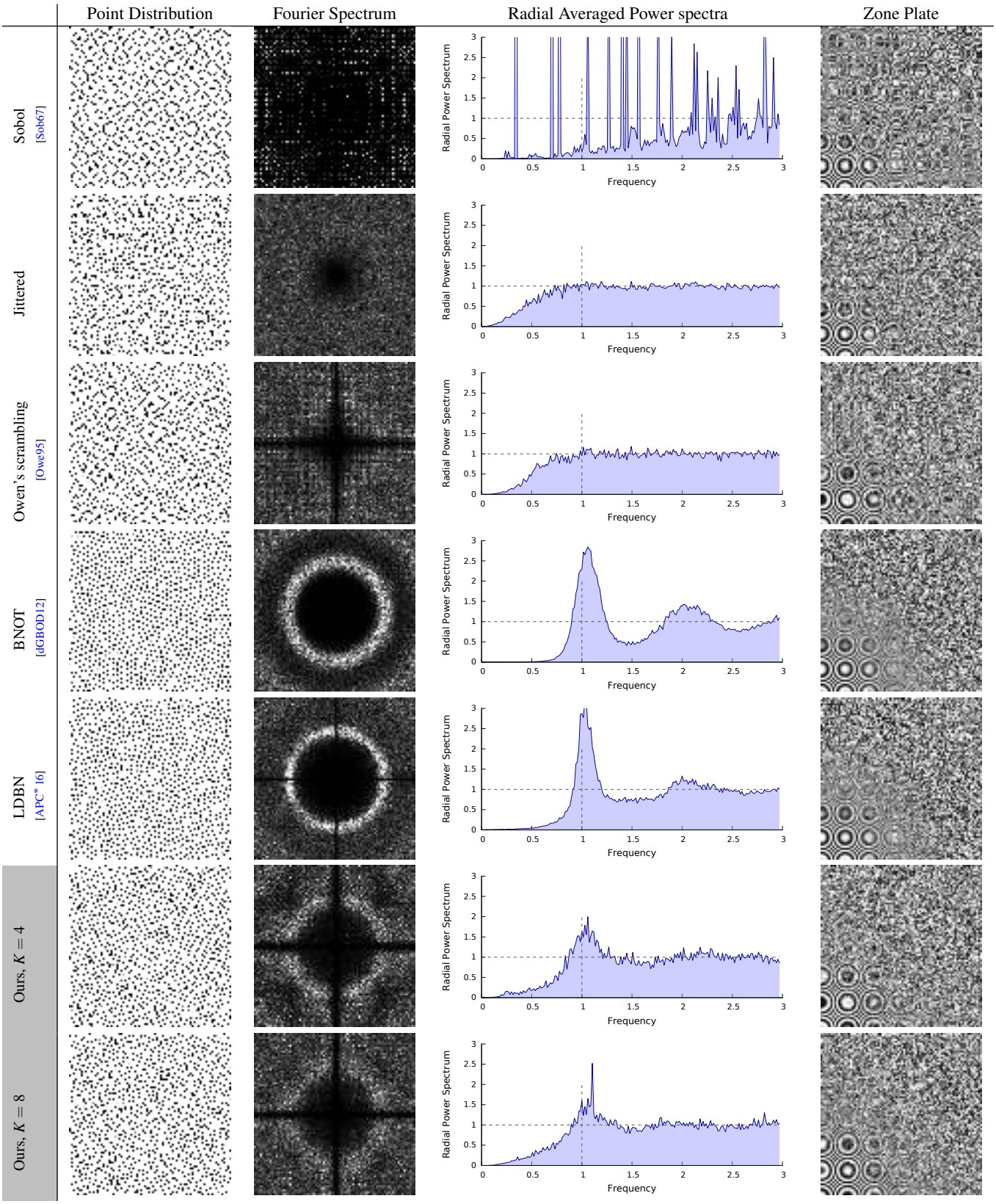

Figure 9: Fourier spectra of a single realization of 4096 points. The zone plate aliasing test uses a single sample per pixel. Please zoom in the Fourier spectra to see narrow peaks.

Lowres version

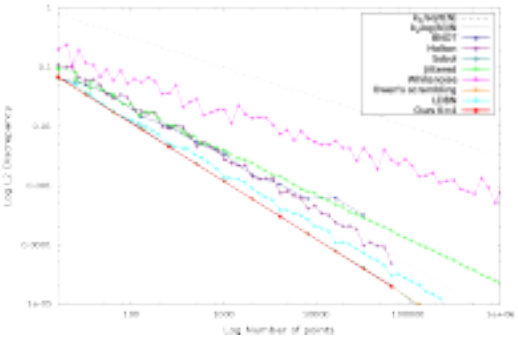

Figure 10: 2-D l_2 discrepancy of our permutation, compared to several other samplers. Note that our sampler generates sets that have a discrepancy similar to Sobol or Owen's scrambling.

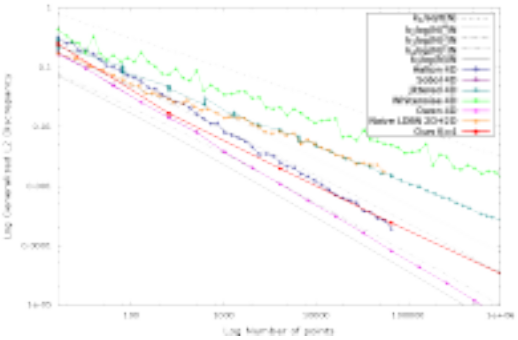

Figure 11: 4-D discrepancy of our permutation, compared to several other samplers. In 4-D, our sampler has a discrepancy worse than Sobol or Owen's scrambling. However, it can be noted that this discrepancy is better than the discrepancy of a stratified set. It is also better than the discrepancy of a naive association of two 2-D sampling patterns (LDBN 2D+2D). This is because scrambling 2-D projections of a Sobol sequence preserves some of the discrepancy of the original 4-D Sobol set, which can't be achieved by a naive recombination of 2D samplers.

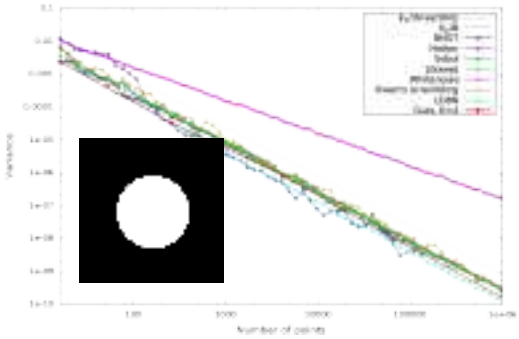

Figure 12: Variance in integration of an analytical disk. Results are given for our permutation and are compared to several other samplers. Note that contrary to other BN samplers such as BNOT, we were able to measure it up to 10^6 samples.

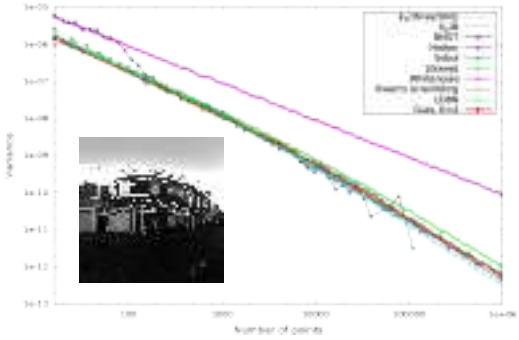

Figure 13: Variance in integration over a 1600x1600 HDR image from the sIBL archive. Results are given for our permutation and are compared to several other samplers. Note that, contrary to other BN samplers such as BNOT, we were able to measure our variance up to 10^6 samples.

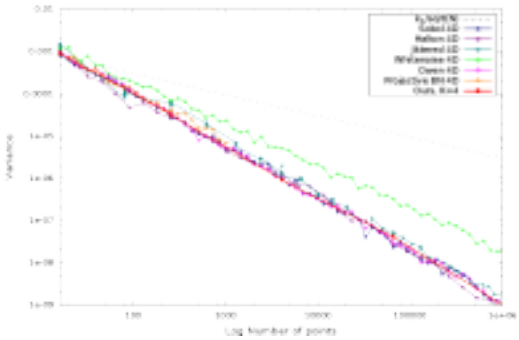

Figure 14: Variance in integration of a 4-D ball. Results are given for our permutation and are compared to several other samplers. Note that, apart for our sampler, for no other 4-D Blue Noise were we able to measure the variance up to 10^6 samples.

In the case of our sampler, we used K=8 to generate 6-D optimized samples, used for the image plane, the light and the lens. We used our hierarchical Owen's scrambling for the remaining dimensions. Similarly to Figure 15, we note that our approach's MSE is better than most methods as is is similar to Sobol's. However, we present much less aliasing. A similar setup was used in Figure 1 (right). There, we used our scrambling technique to sample the image lane, the lens for the depth of field effect, and the light plane of a scene that requires 20-dimensional samples. Compared to the Sobol sequence, we can clearly see that the aliasing has been removed.

Figure 17 presents the difference in renderings when using $K = 4$ or $K = 8$.

8.5. Adaptivity

Since our method relies on a hierarchical grid structure with the $\mathcal{P}^{\lambda-1} \subset \mathcal{P}^{\lambda}$ property, we can perform adaptive refinement on a function using Ostromoukhov's algorithm [Ost07] and using the index of each sample as its ranking (note that this is vary naive way to rank samples). Adaptive sampling obtained with this method is shown in Figure 18. Further experiments are given in supplementary material (Section 4). We stress that this section is to be perceived as a proof of concept, and we do not pretend to outperform true stippling oriented systems.

Using the same scheme as for adaptive sampling, we point out that we can generate the i^{th} sample directly with an $O(\log_{K^s}(i)K^s)$ algorithm. This is detailed in the supplementary material, Section 3.

9. Conclusions and Discussion

In this paper, we have demonstrated a compact and efficient construction which, by applying a set of optimized tile-based permutations on a s-D Sobol LDS, achieves almost BN spectral properties in 2-D projections. In supplementary material, we provide a complete proof that our scrambling preserves LD properties of the initial sequence. As all sampling points after permutations are associated with ordinal numbers, our construction generates a *sequence* of samples. At the same time, thanks to our tile-based approach, our construction naturally supports adaptive sampling.

As a proof of concept, we demonstrated construction of several pairs of optimized dimensions. For other dimensions, any other LDS compatible with the optimized sub-spaces can be used. For example, it could be standard Sobol LDS with appropriate primitive polynomials, or our hierarchical Owen's scrambling, as described in Section 7 and in Section 3 of the supplemental material. In both cases—Sobol or our hierarchical Owen's scrambling—the coordinates of all points of the sequence are uniquely determined by their ordinal numbers and parameters (primitive polynomial's index and the permutation tree of depth d, associated with each dimension).

As a validation of our approach, we have performed a set of standard tests to compare our technique with the state-of-the-art ones in terms of discrepancy, spectral content, and variance in different integration schemes, as well as aliasing tests. Table 1 compares the main features of our construction with the state-of-the-art competitors.

Several improvements may further improve the results shown in this paper. First, the spectral content of our sampler clearly outperforms classical LDS spectral content while being only an approximation of high quality BN samplers such as [dGBOD12]. The lower BN quality results from the sequentiality and LD properties which constrain the samples positions, preventing further spectral optimization. If one discards the sequentiality constraint, global Owen's scrambling could have been considered but it is unclear if there exist permutation trees leading to high quality spectra in higher dimensions.

The number of optimized pairs of dimensions should be increased to address more complex rendering applications. It would be interesting to see whether BN properties in 3-D projections rather than actual 2-D projections would be beneficial in certain integration tasks, where some triplets of dimensions are tightly coupled. We leave these challenging tasks for future work.

Source code and pre-computed LUTs are available in supplementary materials.

Acknowledgments

We thank the anonymous reviewers for their detailed feedback to improve the paper. We also thank Jean-David Génevaux for reviewing preliminary versions of the paper. This project was supported in part by French ANR Excellence Chair (ANR-10-CEXC-002-01) and CoMeDiC (ANR-15-CE40-0006).

Appendix A: Size of the Lookup Tables

As discussed in Section 5, the LUT stores ad-hoc permutations for each pattern $\{\mathbf{q}\}_i^{\lambda}$ in order to generate the set $\mathcal{P}^{\lambda}$. Since the set $\mathcal{Q}^{\lambda}$ is defined from $\mathcal{P}^{\lambda-1}$ (Fig. 7), our offline optimization process follows the hierarchical construction: starting from $\lambda = 1$, we generate successive $\mathcal{P}^{\lambda}$ and $\mathcal{Q}^{\lambda}$ sets. Each time we have a pattern $\{\mathbf{q}\}_i^{\lambda}$ which is not in the LUT, we explore admissible permutations and use our PCF test to select best permutations and add them to the LUT (see Section 5.2). All patterns $\{\mathbf{q}\}_i^{\lambda}$ having the same number of points K^s, the number of distinct patterns, and thus the upper bound on the LUT size is difficult to bound (*e.g.* $O((K^s!)^2)$ if we enumerate the set of Latin hypercube configurations from which we must remove non-dyadic sets). However, starting from the Sobol sequence, the number of distinct $\{\mathbf{q}\}_i^{\lambda}$ sets is limited and as λ increases, the number of entries of the lookup table saturates quickly. As shown in Figure 19, for $K = 2$ only 12 configurations exist up to 10^7 samples ($\lambda = 10$). For $K = 4$, only 512 configurations exist up to 10^6 samples ($\lambda = 6$). For $K = 8$, we found up to 1899 configurations for about 2.10^5 samples ($\lambda = 3$).

In our experiments, most graphs consider a scrambling with $K = 4$ and thus a very compact LUT with 512 entries. For $K = 8$, the graphs show that further configurations may exist for larger point sets. In practice, Figure 9, we have used the LUT with 1899 entries which are fine up to 2×10^5 samples. For larger point sets or if the user wants to reduce the size of the LUT, one can simply return a random permutation when accessing to a non-existing pattern in the LUT. The LD property of the sampler will still be preserved and only the spectral quality will be affected.

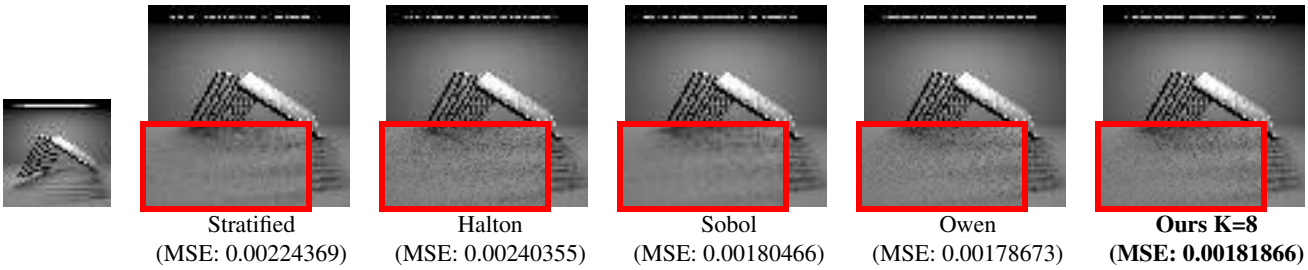

Figure 15: Direct lighting (single light bounce) with 16spp. The deterministic Sobol sampler is used on every dimension apart from the light plane. This dimension was sampled using various samplers, allowing to compare the impact of each of them and compare it with our sampler.

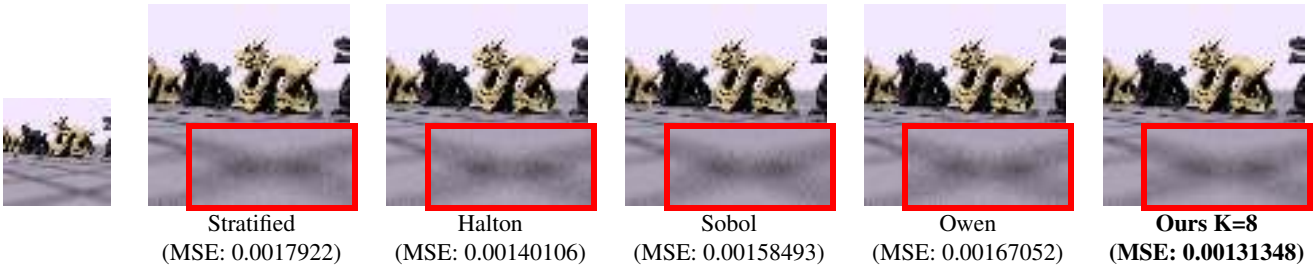

Figure 16: *s*-D Renderings with 16 spp and various samplers. The scene rendered presents depth of field effect, and uses a single light bounce. When using our sampler, the three optimized projections affect the sampling of the image plane, the sampling of the lens for the Depth of Field effect, and the light plane.

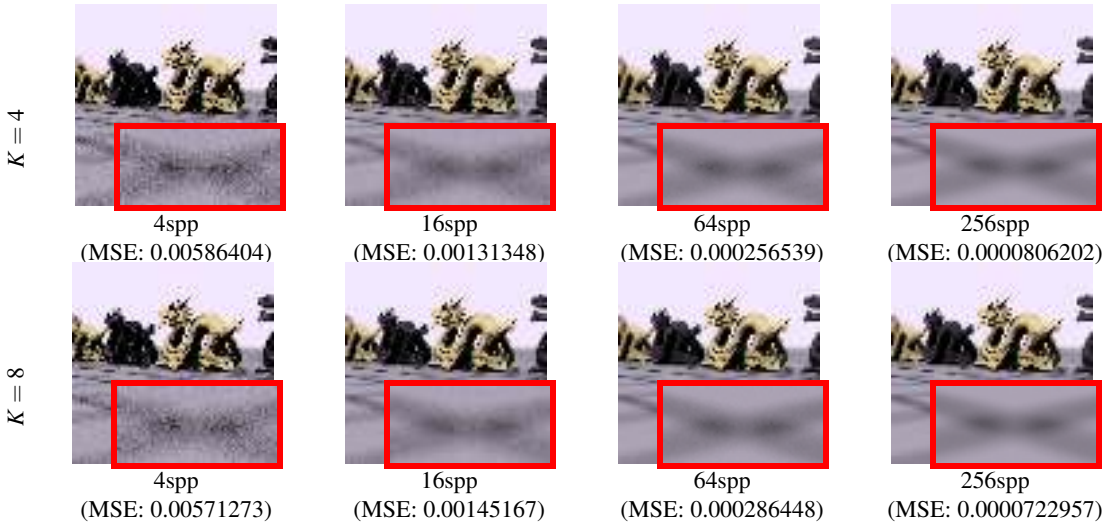

Figure 17: Renderings with our sampler when $K = 4$ and $K = 8$ with a various number of samples per pixel. The scene rendered presents depth of field effect, and uses a single light bounce. The optimized projections affect the sampling of the image plane and the sampling of the lens for the Depth of Field effect.

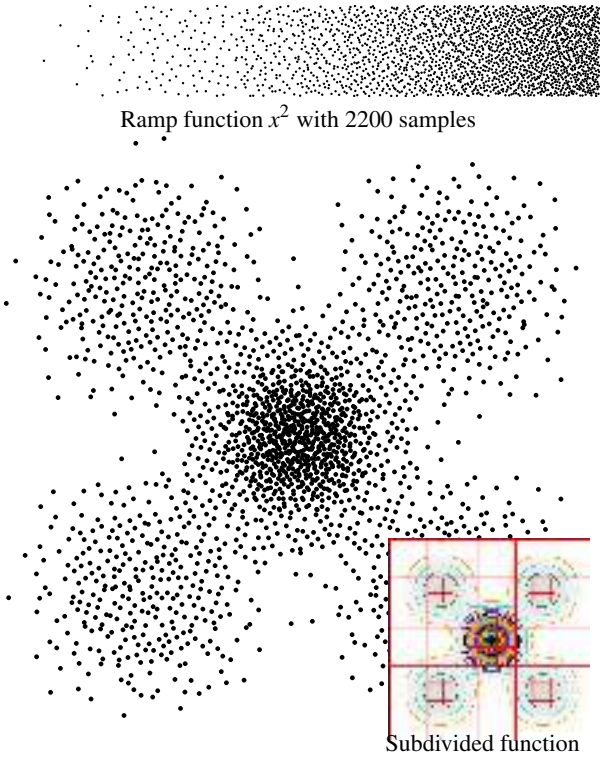

Figure 18: Thanks to our local refinement property and the sequentiality of our sampler, we can use the indices of the samples as ranking and perform adaptive sampling.

As discussed in Section 5, in 2-D , a permutation is defined by 2 Boolean trees with K^2-1 nodes for $K := 2^n$. Hence, for $K = 4$ (resp. $K = 8$), a straightforward encoding of a permutation π_r^λ requires 30 bits (resp. 126 bits). Each entry in the lookup table is indexed by a pattern containing K^2 samples, each sample being encoded onto $2 \cdot 2 \cdot n$ bits. If we allow the LUT to contain m ad-hoc permutations for each pattern, we need

$$K^2 \cdot 4 \cdot n + m \cdot 2(K^2-1)$$

bits.

Following the settings of Section 8, for $K = 4$ and $m = 1$, the LUT size is $512 \cdot (86+30) = 59392$ bits which is about 7 kbytes (about 485 kbytes for $K = 8$).

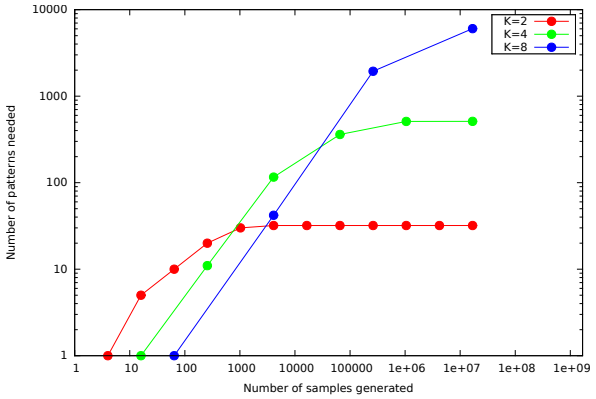

Figure 19: Size of our lookup table in number of entries need to generate a certain amount of samples, depending on the K factor used. Please note that this is in logarithmic scale.

References

[ANHD17] AHMED A., NIESE T., HUANG H., DEUSSEN O.: An adaptive point sampler on a regular lattice. *ACM Trans. Graph. 36*, 4 (jul 2017), 138:1–138:13. 2, 3

[APC*16] AHMED A., PERRIER H., COEURJOLLY D., OSTROMOUKHOV V., GUO J., DONGMING YAN H. H., DEUSSEN O.: Low-discrepancy blue noise sampling. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH Asia 2016) 35*, 6 (2016), 247:1–247:13. 2, 3, 8, 9, 10

[BSD09] BALZER M., SCHLÖMER T., DEUSSEN O.: Capacity-constrained point distributions: A variant of Lloyd's method. *ACM Trans. Graph. 28*, 3 (2009), 86:1–8. 2, 3, 14

[BWWM10] BOWERS J., WANG R., WEI L.-Y., MALETZ D.: Parallel Poisson disk sampling with spectrum analysis on surfaces. *ACM Trans. Graph. 29* (2010), 166:1–166:10. 3

[Coo86] COOK R. L.: Stochastic sampling in computer graphics. *ACM Trans. Graph. 5*, 1 (1986), 51–72. 2, 3

[CP76] CRANLEY R., PATTERSON T. N.: Randomization of number theoretic methods for multiple integration. *SIAM Journal on Numerical Analysis 13*, 6 (1976), 904–914. 9

[CSHD03] COHEN M., SHADE J., HILLER S., DEUSSEN O.: Wang tiles for image and texture generation. *ACM Trans. Graphics 22*, 3 (2003), 287–294. 3

[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 (Oct. 2012), 1784–1796. 2, 3

[dGBOD12] DE GOES F., BREEDEN K., OSTROMOUKHOV V., DESBRUN M.: Blue noise through optimal transport. *ACM Trans. Graph. 31*, 6 (2012), 171:1–171:11. 2, 3, 8, 9, 10, 12

[DH06] DUNBAR D., HUMPHREYS G.: A spatial data structure for fast poisson-disk sample generation. *ACM Trans. Graph. 25*, 3 (July 2006), 503–508. 2, 3

[Dur11] DURAND F.: A frequency analysis of Monte-Carlo and other numerical integration schemes. *MIT CSAIL Technical report TR-2011-052* (2011). 2

[DW85] DIPPÉ M. A. Z., WOLD E. H.: Antialiasing through stochastic sampling. In *ACM SIGGRAPH* (1985), pp. 69–78. 2, 3

[EAG*14] EBEIDA M. S., AWAD M. A., GE X., MAHMOUD A. H., MITCHELL S. A., KNUPP P. M., WEI L.-Y.: Improving spatial coverage while preserving the blue noise of point sets. *Computer-Aided Design 46* (2014), 25 – 36. 2013 {SIAM} Conference on Geometric and Physical Modeling. 2

[EMP*12] EBEIDA M. S., MITCHELL S. A., PATNEY A., DAVIDSON A. A., OWENS J. D.: A simple algorithm for maximal poisson-disk sampling in high dimensions. *Comp. Graph. Forum 31*, 2pt4 (May 2012), 785–794. 2, 3

[Fat11] FATTAL R.: Blue-noise point sampling using kernel density model. *ACM Trans. Graph. 30*, 3 (2011), 48:1–48:12. 2, 3

[FK01] FRIEDEL I., KELLER A.: Fast generation of randomized low-discrepancy point sets. In *Monte Carlo and Quasi-Monte Carlo Methods*

2000 (Berlin, 2001), Fang K.-T., Hickernell F. J., Niederreiter H., (Eds.), Springer-Verlag, pp. 257–273. 3

[GHSK08] GRÜNSCHLOSS L., HANIKA J., SCHWEDE R., KELLER A.: *(t, m, s)-Nets and Maximized Minimum Distance*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008, pp. 397–412. 2

[GRK12a] GRÜNSCHLOSS L., RAAB M., KELLER A.: Enumerating quasi-monte carlo point sequences in elementary intervals. In *Monte Carlo and Quasi-Monte Carlo Methods 2010*. Springer, 2012, pp. 399–408. 9

[GRK12b] GRÜNSCHLOSS L., RAAB M., KELLER A.: *Monte Carlo and Quasi-Monte Carlo Methods 2010*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, ch. Enumerating Quasi-Monte Carlo Point Sequences in Elementary Intervals, pp. 399–408. 2

[Hal60] HALTON J. H.: On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. *Numerische Mathematik 2*, 1 (1960), 84–90. 9

[Hic98] HICKERNELL F.: A generalized discrepancy and quadrature error bound. *Mathematics of Computation of the American* (1998). 9

[Hla61] HLAWKA E.: Funktionen von beschränkter variatiou in der theorie der gleichverteilung. *Annali di Matematica Pura ed Applicata 54*, 1 (1961), 325–333. 8

[HSD13] HECK D., SCHLÖMER T., DEUSSEN O.: Blue noise sampling with controlled aliasing. *ACM Trans. Graph. 32*, 3 (2013), 25:1–25:12. 2, 3

[IPSS08] ILLIAN J., PENTTINEN A., STOYAN H., STOYAN D.: *Statistical analysis and modelling of spatial point patterns*, vol. 70. John Wiley & Sons, 2008. 7

[KCODL06] KOPF J., COHEN-OR D., DEUSSEN O., LISCHINSKI D.: Recursive Wang tiles for real-time blue noise. *ACM Trans. Graph. 25*, 3 (2006), 509–518. 2, 3

[Kel12] KELLER A.: Quasi-Monte Carlo image synthesis in a nutshell. *Monte Carlo and Quasi-Monte Carlo Methods* (2012), 213–252. 2

[LD08] LAGAE A., DUTRÉ P.: A comparison of methods for generating poisson disk distributions. *Computer Graphics Forum 27*, 1 (Mar. 2008), 114–129. 2, 3

[Lem09] LEMIEUX C.: *Monte Carlo and Quasi Monte Carlo Sampling*. Springer-Verlag New York, 2009. 1, 2, 3, 4, 9

[Mat99] MATOUSEK J.: *Geometric Discrepancy: An Illustrated Guide*. Algorithms and Combinatorics. Springer Berlin Heidelberg, 1999. 3, 4

[MF92] MCCOOL M., FIUME E.: Hierarchical Poisson disk sampling distributions. In *Proc. Graphics Interface '92* (1992), pp. 94–105. 2, 3

[Mit91] MITCHELL D.: Spectrally optimal sampling for distributed ray tracing. In *Proc. SIGGRAPH '91* (1991), vol. 25, pp. 157–164. 2, 3

[Nie92] NIEDERREITER H.: *Random Number Generation and quasi-Monte Carlo Methods*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1992. 1, 2, 3, 4

[Ö16] ÖZTIRELI A. C.: Integration with stochastic point processes. *ACM Trans. Graph. 35*, 5 (Aug. 2016), 160:1–160:16. 2

[ODJ04] OSTROMOUKHOV V., DONOHUE C., JODOIN P.-M.: Fast hierarchical importance sampling with blue noise properties. *ACM Trans. Graph. 23*, 3 (2004), 488–495. 2, 3

[OG12] ÖZTIRELI A. C., GROSS M.: Analysis and synthesis of point distributions based on pair correlation. *ACM Trans. Graph. 31*, 6 (Nov. 2012), 170:1–170:10. 2, 3, 7

[Ost07] OSTROMOUKHOV V.: Sampling with polyominoes. *ACM Trans. Graph. 26*, 3 (2007), 78:1–78:6. 3, 12

[Owe95] OWEN A. B.: *Randomly Permuted (t,m,s)-Nets and (t, s)-Sequences*. Springer New York, New York, NY, 1995, pp. 299–317. 3, 4, 9, 10

[PJH16] PHARR M., JAKOB W., HUMPHREYS G.: *Physically based rendering: From theory to implementation*, 3rd ed. Morgan Kaufmann, 2016. 1, 2, 9

[PSC*15] PILLEBOUE A., SINGH G., COEURJOLLY D., KAZHDAN M., OSTROMOUKHOV V.: Variance analysis for Monte Carlo integration. *ACM Trans. Graph. (Proc. SIGGRAPH) 34*, 4 (2015), 124:1–124:14. 2, 8, 9

[QCHC17] QIN H., CHEN Y., HE J., CHEN B.: Wasserstein blue noise sampling. *ACM Trans. Graph. 36*, 4 (July 2017). 2

[RRSG16] REINERT B., RITSCHEL T., SEIDEL H.-P., GEORGIEV I.: Projective blue-noise sampling. *Computer Graphics Forum 35*, 1 (2016), 285–295. 3, 8, 9

[SGBW10] SCHMALTZ C., GWOSDEK P., BRUHN A., WEICKERT J.: Electrostatic halftoning. *Comput. Graph. Forum 29*, 8 (2010), 2313–2327. 2, 3

[SHD11] SCHLÖMER T., HECK D., DEUSSEN O.: Farthest-point optimized point sets with maximized minimum distance. In *Symp. on High Performance Graphics* (2011), pp. 135–142. 2, 3

[Shi91] SHIRLEY P.: Discrepancy as a quality measure for sample distributions. In *Proc. Eurographics '91* (Sept. 1991), pp. 183–194. 2, 3

[SK13] SUBR K., KAUTZ J.: Fourier analysis of stochastic sampling strategies for assessing bias and variance in integration. *ACM Trans. Graph. 32*, 4 (2013), 128:1–128:12. 8

[Slo17] SLOANE N. J. A.: The on-line encyclopedia of integer sequences. *https://oeis.org/* (2017). 4

[Sob67] SOBOL' I. M.: On the distribution of points in a cube and the approximate evaluation of integrals. *Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki 7*, 4 (1967), 784–802. 2, 3, 4, 8, 9, 10

[Uli88] ULICHNEY R.: Dithering with blue noise. *Proceedings of the IEEE 76*, 1 (Jan 1988), 56–79. 2

[WPC*14] WACHTEL F., PILLEBOUE A., COEURJOLLY D., BREEDEN K., SINGH G., CATHELIN G., DE GOES F., DESBRUN M., OSTROMOUKHOV V.: Fast tile-based adaptive sampling with user-specified Fourier spectra. *ACM Trans. Graph. 33*, 4 (2014). 2, 3

[WW11] WEI L.-Y., WANG R.: Differential domain analysis for non-uniform sampling. *ACM Trans. Graph. 30*, 4 (July 2011), 50:1–50:10. 2

[ZHWW12] ZHOU Y., HUANG H., WEI L.-Y., WANG R.: Point sampling with general noise spectrum. *ACM Trans. Graph. 31*, 4 (2012), 76:1–76:11. 2, 3