

Z++: Temporal, Non-Uniform, and 3D Z Sampling

ABDALLA G. M. AHMED, Shenzhen University, China

MATT PHARR, NVIDIA, USA

HUI HUANG*, Shenzhen University, China

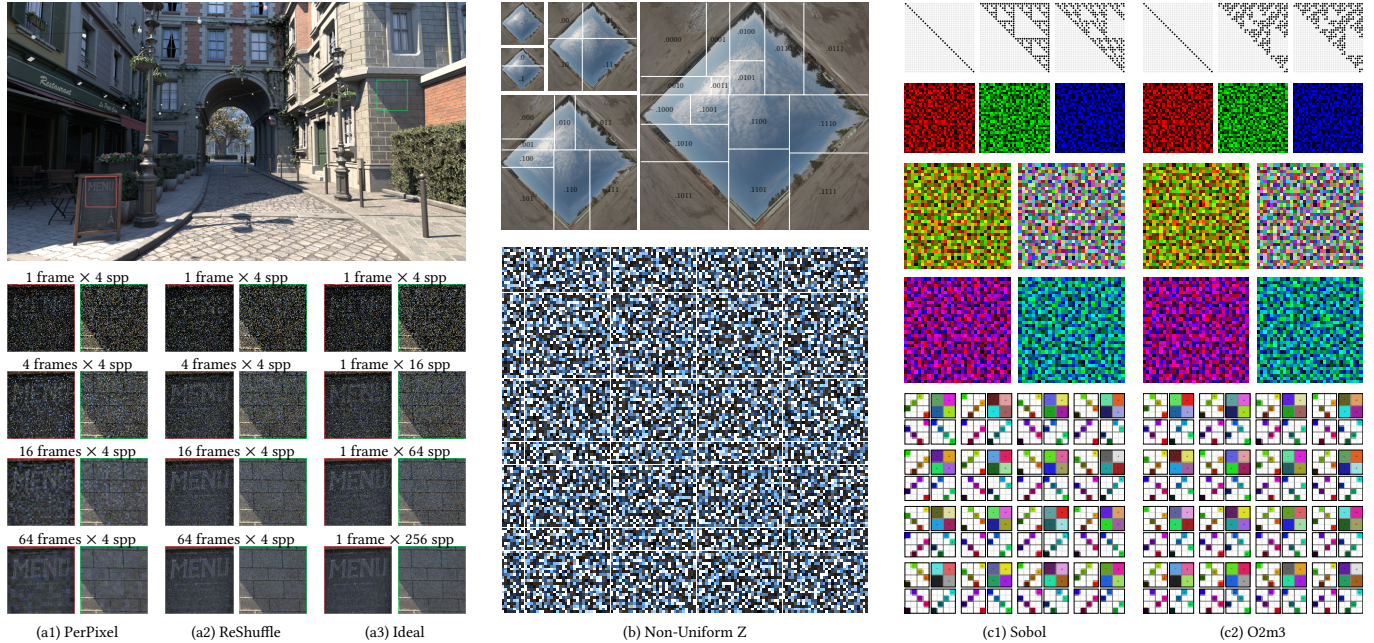


Fig. 1. Visual demonstrations of our proposed extensions to Z sampling. (a) Two extremes of our temporal Z sampling optimized for (a1) fastest PerPixel convergence or (a2) best dithering, compared to (a3) an ideal target using aggregate sampling rate. (b) A recursive slicing of an importance map for non-uniform sampling, and a corresponding Z dithering mask. (c) Generator matrices, 1/2/3D masks, and distinct 4×4 constituent mask blocks, comparing (c2) our novel Z-compatible O2m3 sequence to (c1) 3D Sobol’.

Z sampling redistributes Monte Carlo error in screen space by assigning consecutive low-discrepancy sub-sequences to Z-ordered pixels, using differently shuffled 1D or 2D sequences for the individual constituent domains of a light-transport path. We first analyze the reasons that Z sampling is effective and distill a set of practical design guidelines for Z sampling algorithms. We then propose extensions to Z sampling in two complementary directions: temporally across frames, and spatially to non-uniform and higher-dimensional constituent domains. Our approach thus improves temporal frame accumulation by assigning negatively correlated samples to consecutive frames, without additional memory or runtime cost, while preserving individual-frame quality. It further allows applying Z sampling

to non-uniform, importance-sampled distributions, via a tabulated hierarchical equal-mass subdivision of the importance map. For 3D constituent integration domains, we also introduce a novel low-discrepancy sequence that simultaneously maintains high uniformity in the full 3D domain, along each coordinate axis, and across each pairwise 2D projection. The proposed sequence is quaternary (base-4) structured, hence optimal stratification is attained only for power-of-four sample counts. We empirically demonstrate the advantage of our approach over the common practice of splitting a 3D constituent domain into a 1D–2D pair, and evaluate it against existing 3D sampling alternatives.

CCS Concepts: • **Mathematics of computing** → **Quadrature**; • **Computing methodologies** → **Rendering**.

Additional Key Words and Phrases: sampling, low discrepancy sequences, dithering

ACM Reference Format:

Abdalla G. M. Ahmed, Matt Pharr, and Hui Huang. 2026. Z++: Temporal, Non-Uniform, and 3D Z Sampling. *ACM Trans. Graph.* 45, 6, Article 210 (December 2026), 14 pages. <https://doi.org/10.1145/3842533>

1 INTRODUCTION

Sampling is an essential process in computer graphics, underlying applications that range from traditional halftoning to rendering and

*Corresponding author: Hui Huang (hhzhiyan@gmail.com)

Authors’ addresses: Abdalla G. M. Ahmed, Guangdong Provincial Key Laboratory of Visual Media and Multidimensional Intelligence, College of Computer Science and Software Engineering, Shenzhen University, China, abdalla_gafar@hotmail.com; Matt Pharr, NVIDIA, USA, matt@pharr.org; Hui Huang, Guangdong Provincial Key Laboratory of Visual Media and Multidimensional Intelligence, College of Computer Science and Software Engineering, Shenzhen University, China, hhzhiyan@gmail.com.

Please use nonacm option or ACM Engage class to enable CC licenses. This work is licensed under a Creative Commons Attribution 4.0 International License. © 2026 Copyright held by the owner/author(s). ACM 0730-0301/2026/12-ART210 <https://doi.org/10.1145/3842533>

modern machine learning. In rendering, the pixel-based nature of images amplifies the challenge: the sampling process is repeated over millions of pixels and across tens of constituent domains of the light-transport path, making sampling quality inseparable from cost. Different rendering applications have therefore led to different sampling strategies. In cinematic rendering, where visible error is highly undesirable, low-discrepancy (LD) sequences at high sampling rates are the standard tools. In contrast, undersampling is often unavoidable in real-time applications such as games, so perceptual heuristics such as blue-noise sampling are commonly used to make the remaining error less objectionable. Since the work of Georgiev and Fajardo [2016], a common tool in this setting has been a pre-optimized, tiled dithering mask that coordinates Monte Carlo error across pixels, analogous to traditional halftoning masks in the lower-dimensional setting of image dithering [Ulichney 1993].

In contrast to common real-time sampling practices, Ahmed and Wonka [2020] introduced a mask-free alternative, Z sampling, that assigns consecutive LD subsequences to recursively shuffled Morton-ordered pixels. In practice, this simple ordering mechanism achieves screen-space dithering quality comparable to explicitly optimized masks, while retaining the efficiency, scalability, and versatility of LD and independent sampling. This makes Z sampling a promising candidate for bridging real-time and high-quality rendering. Its adoption as the default sampler in *pbrt-4* [Pharr et al. 2023], and its recent integration with the high-dimensional SZ sequences [Ahmed et al. 2025], further support this view.

Toward the goal of a more versatile and effective sampler, we introduce three extensions to Z sampling. First, we extend Z sampling temporally, improving temporal accumulation by assigning negatively correlated samples to consecutive frames without additional memory or runtime cost. Second, we extend it to non-uniform constituent domains, allowing Z sampling to operate with importance-sampled distributions commonly used in light transport. Finally, for three-dimensional constituent integration domains, we introduce a novel LD sequence construction that balances uniformity in the full 3D domain, in each 1D coordinate projection, and across all three pairwise 2D projections. We empirically demonstrate its advantage over splitting a 3D constituent into a 1D–2D pair, and evaluate it against existing 3D sampling alternatives.

2 RELATED WORK

In this section, we review related work, and in the following section we introduce the essential technical background needed to understand the current paper in more detail.

2.1 Nets and (t, s) -Sequences

Low-discrepancy (LD) sampling is the foundation of quasi-Monte Carlo integration and has been widely used in computer graphics since the work of Shirley [1991], Mitchell [1992], and Keller and collaborators [1996; 2004; 2006; 2013]. By minimizing *discrepancy*, a measure of the irregularity of a point distribution, LD sampling provides error guarantees that depend on the variation of the integrated function. Widely used examples include the 1D van der Corput sequence [1935] and higher-dimensional Halton [1960], Sobol' [1967], and Faure [1982] sequences. Unlike fixed point sets, sequences allow

an arbitrary number of samples to be generated, although they often perform best at specific sample counts, such as powers of two for Sobol'.

Niederreiter [1987; 1992] provides a standard framework for describing LD quality. A base- b (t, m, s) -net is a set of b^m points in the s -dimensional unit hypercube such that every elementary subdivision into b^{m-t} congruent strata contains exactly b^t points per stratum. A base- b (t, s) -sequence is an infinite sequence in which, for every m , each consecutive block of b^m points forms a (t, m, s) -net. For example, the van der Corput sequence is a dyadic $(0, 1)$ -sequence, and the first two dimensions of Sobol' form a dyadic $(0, 2)$ -sequence. Smaller bases and smaller quality parameters t generally indicate stronger uniformity.

Research on LD sampling in computer graphics has largely focused on either 2D [Ahmed et al. 2023b; Ahmed and Wonka 2021; Kollig and Keller 2002] or high-dimensional constructions [Ahmed et al. 2025; Bonneel et al. 2025; Paulin et al. 2022]. Only limited work has specifically addressed 3D nets and sequences. Grünschloß and Keller [2009] proposed an algorithm for generating $(0, m, 3)$ -nets whose first two dimensions correspond to Larcher–Pillichshammer points [2001; 2003]. Munkberg et al. [2011] extended this construction to improve screen-space projections. Both approaches generate point sets rather than infinite sequences.

Randomization, or scrambling, turns deterministic LD sequences into uniformly distributed random samples while preserving their net and sequence properties. Owen scrambling [1995; 1998] is the standard randomization construction and has several efficient implementations [Ahmed et al. 2023a; Burley 2020; Helmer et al. 2021; Owen 2003].

2.2 Stochastic Sampling

Foundational early work on sampling for computer graphics by Cook et al. [1986; 1984] and Dippé and Wold [1985] was based on insights about visual perception that drew from Yellott's work [1983] measuring the distribution of photoreceptors in monkeys' eyes. The key insights were that by using irregular distributions of samples with *blue noise* [Ulichney 1988] spectral properties, approximating photoreceptor distributions, aliasing is converted into high-frequency noise, which is more perceptually pleasing to humans. More generally, their insight was that, given human viewers, the perceptual manifestation of error is as important as the absolute amount of error in an image. This led, for example, to Mitchell's work [1987] applying error-diffusion [Floyd and Steinberg 1976] to sample distribution. Closely related are ideas underlying both error diffusion [Ulichney 1987] and dithering masks [Bayer 1973; Ulichney 1993] in halftoning.

2.3 Screen-Space-Oriented Sampling

Building on the perceptual view of sampling error discussed above, Georgiev and Fajardo [2016] introduced the screen-space distribution of Monte Carlo error as an explicit optimization target for rendering samples. This inspired subsequent work by Heitz and collaborators [2019; 2019], Wolfe and collaborators [2025; 2022], and Donnelly et al. [2024]. Öztireli developed related algorithms for

optimizing sampling patterns according to prescribed spectral properties [Öztireli 2020]. These methods generally focus on optimizing precomputed sets or masks of low-dimensional samples.

The intuition behind screen-space error redistribution can be understood as an economic use of the limited sample budget. The goal is to assign complementary samples, or complementary combinations of samples, to neighboring pixels so that they compensate for each other's potential sampling error. This allows local image regions to incorporate more light-path possibilities than any individual pixel can afford independently, making feasible strategies possible even in the extreme case of one sample per pixel. For example, when one integration dimension dominates, as in motion blur or an extended linear light source, neighboring pixels can collectively sample that dimension by taking complementary decisions.

2.4 Mapping LD Samples to Pixels

Screen-space error redistribution contrasts with traditional LD sampling, which optimizes the distribution of samples but does not explicitly consider how the resulting error is distributed over pixels or perceived by viewers. A classic rendering approach is to use the first two dimensions of an LD sequence to choose the pixel and subpixel position [Grünschloß et al. 2012]. A fundamental shortcoming of this mapping is that nearby sequence samples are assigned to far-apart pixels. Consequently, the remaining dimensions used for light-transport sampling within a pixel come from non-consecutive samples of the sequence. Although the resulting per-pixel point set remains LD, it can be spectrally distorted, exhibiting strong interdimensional correlations that manifest as checkerboard artifacts before convergence [Ahmed et al. 2026; Pharr et al. 2023, Fig. 8.37]. A common alternative is to use independently scrambled high-dimensional LD sample sets for individual pixels [Bonneel et al. 2025], which replaces checkerboard artifacts with a white-noise distribution of Monte Carlo error.

Inspired by Georgiev and Fajardo [2016], Heitz et al. [2019] addressed this issue by replacing the implicit pixel-sample mapping of the LD sequence with an explicitly optimized redistribution that assigns negatively correlated samples to nearby pixels. This ensures that each pixel receives a contiguous block of samples, so that the samples used for light-transport integration at that pixel remain well distributed for suitable sample counts. The downside is that explicit offline optimization sacrifices part of the scalability and versatility that make LD sampling attractive.

Ahmed and Wonka's Z sampler [2020] introduced a different mapping based on a scrambled screen-space Morton, or Z, curve [1966]. This construction exploits the modular structure of dyadic LD sequences, which systematically attain high-quality distributions at power-of-2 point counts. The approach can be used either with *padding*, where independent low-dimensional LD sequences are used for the higher-dimensional integrand, or with high-dimensional LD sequences such as Sobol' [1967] or SZ [Ahmed et al. 2025].

3 BACKGROUND

We begin by reviewing key details of generator-matrix-based implementations of binary sequences as well as the original Z sampler. These allow us to revisit the Z sampler and interpret it from the

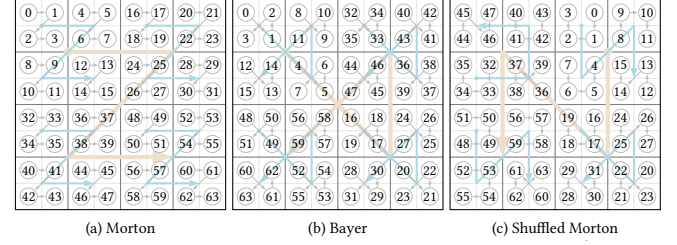


Fig. 2. A miniature 4×4-pixel image, with 2×2 subpixels hierarchically indexed using (a) Morton [1966], (b) Bayer [1973], and (c) recursively shuffled Morton ordering [Ahmed and Wonka 2020], which is essentially indistinguishable from shuffled Bayer's. An 8×8 dithering mask is readily obtained by assigning to each subpixel the bit-reversed index.

perspective of dithering. This, in turn, provides a better understanding of its structure and the foundation for our temporal extension, introduced in Sec. 4.

3.1 Generator Matrices

A common approach to building low-discrepancy sequences, known as the *digital construction* [Niederreiter 1992], is to linearly transform a digit-reversed sequence using a carefully constructed set of *generator matrices*, one for each sampled dimension. The Sobol' sequence, for example, is defined by a set of binary matrices:

$$\left(\begin{array}{c|c|c|c|c} \text{Matrix 1} & \text{Matrix 2} & \text{Matrix 3} & \text{Matrix 4} & \dots \end{array} \right) V, \quad (1)$$

where V is a binary column vector encoding the van der Corput sequence; i.e., for the i th sample, V is the binary representation of i , bit-reversed. The recently introduced SZ sequences [Ahmed et al. 2025] likewise use binary generator matrices, but follow a different matrix design. For both families, matrix-vector multiplication uses arithmetic over the Galois field $\text{GF}(2)$, enabling efficient implementations on digital computers using bitwise AND and XOR for multiplication and addition, respectively [Ahmed et al. 2025; Bratley and Fox 1988; Sobol' et al. 2011].

3.2 Z Sampling

Ahmed and Wonka's Z sampler [2020] uses a randomly scrambled Morton curve to assign sample indices in an LD sequence to pixel samples in an image. They originally presented Z sampling mainly through its local operation: mapping pixel samples to indices in a low-discrepancy sequence. Here we focus on the global effect of this mapping across an entire image.

To simplify the discussion, in the remainder of this subsection we will ignore the effects of randomization in order to directly focus on the implications of Morton-curve assignment of sample indices. Indeed, while randomization is crucial for removing visible structure in the sampling pattern, it is analogous to LD randomization schemes like Owen scrambling in that it preserves the key underlying characteristics of the sample assignment. Thus, insights from investigating non-randomized Z sampling apply to its randomized use in practice.

Fig. 2(a) illustrates unrandomized Z sampling in the simplified setting of a 4×4 -pixel image with 4 samples per pixel. First, note that the four samples in each pixel are visited consecutively: the first pixel has indices 0, 1, 2, 3, the one to its right is assigned 4, 5, 6, 7 and, more generally, in the j th pixel visited, the pixel samples are assigned consecutive sample indices $4j, 4j + 1, 4j + 2, 4j + 3$. Due to the Morton curve’s construction, progressively wider sample index intervals are assigned at coarser granularities. For example, each 2×2 block of pixels with even coordinates at its upper-left pixel—(0,0), (2,0), etc.—is assigned a contiguous block of 16 sample indices: 0–15 for the upper-left block, 16–31 for the one next to it, and so on. This pattern continues for higher-resolution images.

When used with base-2 LD sequences, this approach to sample assignment ensures well-distributed samples across a hierarchy of scales. As an example, consider assigning the first two dimensions of the Sobol’ sequence to pixel samples for use with sampling an area light source. Recall from Sec. 2.1 that for a $(0, 2)$ -sequence like Sobol’s first two dimensions, each consecutive block of 2^m points forms a $(0, m, 2)$ -net that satisfies a variety of stratification constraints. Here, this property implies that not only will the samples in an individual pixel be well-stratified themselves, but that the samples used within the 2×2 blocks of pixels will be both stratified within each pixel and also stratified collectively, ensuring that nearby pixel samples are sampling the light in different places than their neighbors. This, too, continues progressively at larger power-of-two-sized blocks of pixels. This sample distribution can be seen as a form of antithetic sampling [Hammersley and Morton 1956], though multi-resolution and low-discrepancy.

3.3 Connection to Dithering

Having seen that the Z-indexed samples assigned to a pixel tend to be decorrelated from those assigned to its quadtree neighbors, we can gain some insight about how this reduces perceptual error in rendered images by considering sampling in the context of dithering and error diffusion in halftoning, where neighboring pixels collectively represent shades that no individual pixel can represent alone. First, we define a *dithering mask* as the mapping

$$m : \mathbb{Z}^2 \rightarrow [0, 1), \quad (x, y) \mapsto m(x, y), \quad (2)$$

from integer pixel coordinates to a unit-interval offset. A mask may be defined procedurally, as in ordered dithering [Bayer 1973], or generated ahead of time through an optimization process, as in void-and-cluster [Ulichney 1993]. Given a grayscale image and a dithering mask m , we represent each pixel value v as white if $v > m(x, y)$ and black otherwise. The essence of this process is that, as long as the mask values are uniformly distributed over $[0, 1)$, the fraction of white pixels approximates v , preserving the average brightness.

Consider now defining a dithering mask using Z sampling with the 1D van der Corput points. Each aligned 2×2 pixel block will have one mask value in each of the intervals $[0, 1/4)$, $[1/4, 1/2)$, $[1/2, 3/4)$, $[3/4, 1)$. Similarly, each aligned 4×4 block contains one mask value in each sixteenth of the unit interval, and so on. We make the observation that Morton ordering produces the same recursive structure as Bayer’s ordered-dithering mask, differing only in the ordering of quadrants, which disappears under the recursive

shuffling used by Z sampling, as illustrated in Fig. 2. Thus, the 1D mask produced by Z sampling is best understood as a recursively shuffled ordered-dither mask. The resulting noise is therefore not strictly blue noise, but a multi-resolution stratified pattern, which we hereafter call *Z noise*.

We next note that upper-triangular matrix shuffling preserves power-of-2 ranks [Faure and Tezuka 2002] and is therefore absorbed by the recursive quadrant shuffling. Consequently, each dimension of a Z-indexed dyadic LD sequence, such as Sobol’ or SZ, induces a Z mask. For a 2D source domain, a $(0, 2)$ -sequence induces not just one mask, but a family of masks over the binary stratifications of the domain, including its 1D and mixed projections. For example, a Z mask over a red×green domain also induces masks over a hierarchy of mixed yellow shades.

Fig. 3 illustrates the practical utility of Z masks. On the one hand, they are generated procedurally like Bayer’s ordered dither, requiring no precomputed optimization or storage. On the other hand, they achieve reasonable antialiasing quality, not matching optimized void-and-cluster masks, but at orders of magnitude lower cost. Beyond this low cost, Z masks offer several advantages over explicitly constructed masks by leveraging the underlying LD sequences. For example, they automatically scale with dimension over larger pixel blocks and adjust naturally to different sampling rates. These properties make Z masks a promising foundation for a more versatile sampler.

We conclude this section with the observation that severely under-sampled rendering can thus be viewed as applying a set of dithering masks, one per sampled axis, induced by the sampler’s mapping from sample dimensions to pixels. Just as Z masks give dithering with visually pleasing high-frequency error across images, Z sampling similarly distributes error from Monte Carlo integration in rendering across images. This connection between scrambled ordered dithering, blue noise, and stratification is interesting because stratified sampling was already proposed by Dippé and Wold [1985] as an approximation to Poisson-disk sampling, yet its role as an ordered-dithering mask appears to have been largely overlooked. A possible historical explanation is that the works of Bayer [1973] and Dippé and Wold [1985] predate Owen [1995] scrambling, and hence predate the digital-net language that makes this connection natural. Z sampling can therefore be viewed as sitting at the intersection of halftoning, stochastic sampling, and low-discrepancy construction. The following section builds on this foundation by extending Z sampling to temporal accumulation, non-uniform source domains, and three-dimensional constituents.

4 Z++

This section develops the core ideas of our proposed Z++ extensions, building on the ideas from the previous section and applying them to temporal accumulation, non-uniform source domains, and three-dimensional constituents.

4.1 Temporal Z Sampling

Many real-time rendering pipelines rely on temporal accumulation to amortize sampling cost across frames [Donnelly et al. 2024; Wolfe et al. 2022]. This makes temporal behavior an important requirement

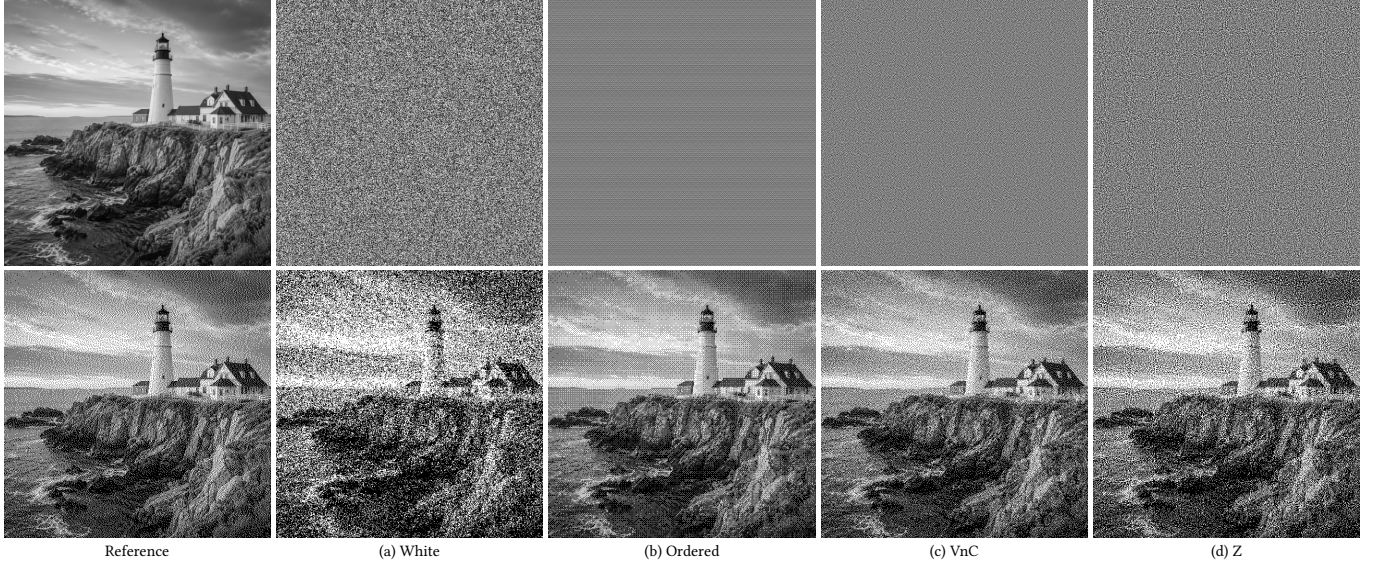


Fig. 3. A grayscale image thresholded against different dithering masks, comparing (a) white noise, (b) Bayer's ordered dither [1973], (c) Ulichney's void-and-cluster (VnC) mask [1993], and (d) Z. The masks are shown in the top row and the corresponding dithered images below. Z masks are computed on the fly like ordered dither, yet provide antialiasing behavior closer to optimized VnC masks. Floyd–Steinberg error-diffusion dithering is shown below the input image for reference.

for screen-space error redistribution. In this section, we develop this functionality for Z sampling. Temporal averaging is typically considered over a finite perceptual window, which we take as 64 frames in the following discussion.

A useful property of the LD sequences underlying Z sampling is that complementing samples are easy to obtain. For upper-triangular-matrix-based sequences such as Sobol' and SZ, flipping the least significant bit of the sample index retrieves a sample in the opposite half of each axis. More generally, a complementary block of 2^m consecutive samples can be generated by flipping the m th bit of each sample index, namely the least significant bit of the block index. Thus, a new temporal mask may be derived using

$$\text{pixelSeqNo} \hat{=} \text{frameIndex}, \quad (3)$$

where `pixelSeqNo` is the Z, or Morton, index of the pixel. This may sound like a simple approach to temporal Z sampling, and our complete solution is indeed simple, though not trivial: it relies on the interaction between Z sampling and bit-level LD sequence manipulation.

The immediate difficulty is that complementary samples are already exploited by Z sampling to distribute samples across quadtree-neighboring pixels. Complementing over four consecutive frames, for example, gives each individual pixel $4\times$ best-quality samples, but a 2×2 block of pixels recycles the same shared sample set, merely permuting it among the pixels. At very low sampling rates, or in high-dimensional shadow and glossy regions, such under-sampled pixel blocks may average substantially differently from adjacent blocks, producing the visible pixelation shown in Fig. 1(a1). As more frames accumulate, the pixelation becomes coarser, e.g., 4×4 blocks over 16 frames, while its contrast fades between blocks. The exact

behavior depends on the local scene complexity and the severity of undersampling. Each individual pixel still converges ideally over time, and each complemented frame is a valid Z mask, but consecutive masks do not complement each other properly at the block level. We therefore call this variant *PerPixel* temporal Z sampling, because it provides the best temporal sample sequence for each individual pixel. It works well when the aggregate sampling rate reaches convergence within the designated temporal window.

Our first remedy for the pixelation issue is to permute the frame sequence, analogously to spatial interlacing in early display systems. For example, the reordering

$$(0, 1, 2, 3, 4, 5, 6, 7) \mapsto (0, 1, 6, 7, 4, 5, 2, 3) \quad (4)$$

effectively exchanges samples between each sibling pair of pixels in a 2×2 block and a cousin pair from the enclosing 4×4 block. This reordering is obtained by using the second bit of the frame index to permute the third. Hierarchical temporal interlacing is obtained using

$$\text{frameIndex} \hat{=} (\text{frameIndex} \& 0xAAAAAAAA) \ll 1; \quad (5)$$

The essence of this exchange is that the complementary paired block is itself randomly shuffled within each enclosing block. This makes the temporal extension fit naturally within the Z-sampling framework, using the same hierarchical shuffling mechanism to exchange samples between nearby pixels over time. Each pixel now obtains its $4\times$ best-quality samples over a doubled cycle of eight frames, while the pixelation scale is reduced by half. We refer to this variant as temporal interlacing, or *TZ*.

TZ extends the *PerPixel* model to converge over a longer temporal window. For more severe undersampling, we propose *spatiotemporal interlacing*, or *STZ*. The idea is to vary the temporal averaging

behavior of pixels according to their spatial index. After shuffling the Z index of each pixel, we further permute it using

$$\text{pixelSeqNo} \leftarrow \text{pixelSeqNo} \ll 2; \quad (6)$$

which exchanges samples locally or remotely depending on the shuffled index, before applying temporal interlacing. That is, after applying Eq. (5) once per frame, Eq. (6) is applied before Eq. (3), both once per pixel. This variant compromises the per-frame mask quality, but substantially reduces pixelation under severe undersampling.

At the opposite extreme from the PerPixel model, where full convergence is not attainable within the temporal window, we *ReShuffle* the mask independently for each frame, trading convergence for inter-frame dithering quality. Instead of shuffling pixel indices independently for each constituent subspace, we use a single shuffled Z curve together with a high-dimensional sequence such as SZ [Ahmed et al. 2025], and apply an independent shuffle per frame. Hence, interdimensional correlations are dispersed temporally rather than spatially. The first three models thus offer convergence guarantees, strongest for PerPixel through its contiguous per-pixel LD blocks, whereas *ReShuffle* is a heuristic for suppressing pixelation by making per-pixel blocks independent between frames. Empirical evaluation is provided in Sec. 5.1 and the supplementary video.

4.2 Non-Uniform Z Sampling

The basic Z-sampling construction assumes uniform constituent domains. In practice, however, it is useful to be able to sample domains according to a prescribed density map reflecting the relative importance of regions of the domain—examples include lens apertures, tabulated reflection functions, and images used to specify emission distributions from surfaces or as environment maps that illuminate an environment. As shown by Wolfe et al. [2025], it can be beneficial to be able to generate such samples while maintaining the distribution properties of the underlying point distribution. To this end, we have developed an extension of Z sampling to such problems.

In a preprocess, we generate a summed-area table [Ahmed 2014; Crow 1984] of the density map and then recursively split the map into rectangles such that there is equal mass on both sides of each split. The table guides the partitioning by (1) locating candidate slicing pixels via binary search, and (2) evaluating mass at continuous coordinates using interpolated summed-area-table queries. This continues to a specified depth, after which the set of split rectangles as well as an array that records the splits is returned. The recursive construction is carefully written so that rectangles are created in a Morton-like order. In this way, subsequent sampling of a rectangle retains the hierarchical multi-resolution stratification of Z sampling. This process is summarized in Alg. 1. Note that the approach can easily be applied to non-uniform sampling in higher-dimensional domains. Further, the parameter D allows trading off sampling accuracy for memory use and computation time; Fig. 6 compares results with varying D values.

Sampling is straightforward, requiring just a single bit-reversed index V , as provided by the Z sampler. A rectangle is selected with uniform probability and then a point in it is selected uniformly as well; see Alg. 2. The sampling algorithm thus lacks any control-

Algorithm 1: Recursive construction of the rectangle table and slicing tree using the summed-area table S of an importance distribution. **HSPLOT/VSPLOT** bisect on S to locate the continuous coordinate at which the left/lower sub-box holds fraction ρ of the parent’s mass.

Input: Summed-area table S of a $w \times h$ importance map; subdivision depth D

Output: leaf rectangles $\mathcal{R}$ and internal split table $\mathcal{T}$

```

1  $N \leftarrow 2^D$ 
2 push ( $B = [0, w] \times [0, h]$ ,  $n = N$ ,  $k = 1$ )
3 while stack is nonempty do
4   pop ( $B$ ,  $n$ ,  $k$ )
5   if  $n = 1$  then
6      $\mathcal{R}[k - N] \leftarrow B$ 
7   else
8      $n_1 \leftarrow \lfloor n/2 \rfloor$ 
9      $\rho \leftarrow n_1/n$ 
10    if width( $B$ ) > height( $B$ ) then
11       $s \leftarrow \text{HSPLOT}(S, B, \rho)$ 
12       $\mathcal{T}[k] \leftarrow (s, X)$ 
13       $(B_L, B_R) \leftarrow \text{split } B \text{ at } x = s$ 
14    else
15       $s \leftarrow \text{VSPLOT}(S, B, \rho)$ 
16       $\mathcal{T}[k] \leftarrow (s, Y)$ 
17       $(B_L, B_R) \leftarrow \text{split } B \text{ at } y = s$ 
18    push ( $B_L$ ,  $n_1$ ,  $2k$ )
19    push ( $B_R$ ,  $n - n_1$ ,  $2k + 1$ )
```

Algorithm 2: Z-sample a point from a sliced environment map. The D high bits of V select a rectangle and the remaining $32 - D$ bits are used to generate a 2D Sobol’ point that is used to uniformly sample within the rectangle’s area.

Input: Bit-reversed 32-bit sample index V ; rectangle table $\mathcal{R}$ of size $N = 2^D$

Output: Sampled uv coordinates and associated PDF

```

1  $R \leftarrow \mathcal{R}[V \gg (32 - D)]$  // Uniformly sample rectangle
2  $V' \leftarrow V \ll D$ 
3  $(x, y) \leftarrow (V'/2^{32}, P(V')/2^{32})$  // 2D Sobol’
4  $(u, v) \leftarrow (R.x_0 + x(R.x_1 - R.x_0), R.y_0 + y(R.y_1 - R.y_0))$ 
5  $A \leftarrow (R.x_1 - R.x_0)(R.y_1 - R.y_0)$  // Rectangle area
6 PDF  $\leftarrow 1/(NA)$ 
7 return  $((u, v), \text{PDF})$ 
```

flow, unlike traditional importance sampling algorithms that require binary searches or hierarchical warping during sampling [Clarberg et al. 2005]. Further, only a single dimension must be sampled by the sampler, benefiting performance and avoiding the consumption of an additional dimension from the sampling domain.

The split table $\mathcal{T}$ is used for computing the PDF for a given sample location, as is required for multiple importance sampling [Veach 1997]; it hierarchically encodes the splitting decisions made when

constructing the rectangles, allowing an $O(\log N)$ search to find the offset for the rectangle a sample is in.

4.3 A Dedicated 3D Sampling Sequence

Three-dimensional domains are ubiquitous, appearing not only as spatial domains but also in color spaces. Yet, the requirements of a balanced 3D sampling sequence are not well served by the common choices. The original Z-sampling formulation splits a 3D constituent into independently shuffled 1D and 2D sequences, which preserves the simplicity of the constituent-wise construction but weakens coverage of the full 3D domain and privileges some projections over others. Other alternatives also fall short. A Sobol' sequence is binary and efficient, but it is not neutral across the three pairwise projections, favoring the first dimension pair over pairs involving the third dimension. The Faure sequence is more dimension-neutral, but is not binary and therefore does not fit as naturally with digital, bitwise implementations. The Halton sequence is neither binary nor dimension-neutral. These limitations motivate a dedicated 3D LD sequence construction.

We start our development by noting that, although a dyadic $(0, 3)$ -sequence provably does not exist [Niederreiter 1992, Corollary 4.24], dyadic $(0, m, 3)$ -nets are realizable, offering excellent and balanced 3D coverage along with excellent 2D and 1D projections [Grün-schloß and Keller 2009; Munkberg et al. 2011]. Inspired by the recent heuristic development of SZ sequences [Ahmed et al. 2025], our idea to bypass the binary base limitation is to search instead over base-4. That is, we seek to find a sequence that delivers optimal nets over powers of 4 instead of 2: a sequence of points such that, for all m , every consecutive block of 2^{2m} points is a $(0, 2m, 3)$ -net, which we hence call an "O2m3" sequence.

For a matrix-based construction, the $(0, m, 3)$ -net constraints translate into an established condition that any hybrid $m \times m$ matrix of leading rows from the three generator matrices is invertible [Niederreiter 2005, Theorem 1], since each such hybrid matrix maps points to a specific stratification. Hence our goal is to find a triplet of binary Galois field $\text{GF}(2)$ matrices that satisfies this condition for all m across its $2m \times 2m$ upper-left corner matrices. Without loss of generality, we may assume the identity for the first matrix, and search over complementing pairs of matrices. Our search is limited to upper-triangular matrices built from 2×2 blocks, factoring out lower-triangular components. We start with an arbitrary triplet of 2×2 matrices that satisfies the condition, for example,

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad (7)$$

and try to extend the matrices by 2 in each step, retaining sets that satisfy the condition.

Through an extensive pilot search, we noted that each initial set of any matrix size has a single binary degree of freedom for extension, applying to the choice of the top 2×2 block of the extension pair of columns in each matrix. Given that, we then performed a search with backtracking through a systematic choice of the top pair of rows, and obtained the triplet

$$\begin{pmatrix} I & A & B \\ 0 & I & C \\ 0 & 0 & I \end{pmatrix}, \quad (8)$$

as a canonical representative O2m3 sequence.

As far as Z sampling is concerned, the essence of O2m3 sequences is that (i) their $(0, 2m, 3)$ -net building blocks imply $(0, 2m, 1)$ -net and $(0, 2m, 2)$ -net projections, and (ii) their base-4 form aligns with that of the Z-indexing tree. Thus, they are well aligned with both the source 3D and target image-plane domains. To demonstrate this, in Fig. 1(c) we compare 1 sample per pixel (SPP) dithering 3D Z masks using 3D Sobol' vs O2m3. The top plots show the joint-3D and pairwise 2D masks, and in the bottom we see 16 sets of distinct 2-bit quantized 4-point blocks, which comprise the complete sets, up to shuffling. While the masks look similar at first glance, the shortcoming of the 3D Sobol' sequence appears upon careful examination of the second (green) and third (blue) pair of axes, where two quadrants are left empty in each 4-point block. Instead of having a representative of each quadrant in every 2×2 block in the green-blue mask, we therefore have either dark and light cyan or distinct blue and green. In contrast, the O2m3 sequence retains the best $(0, 2m, 2)$ -net quality across all three 2D projections for each $2^m \times 2^m$ block of pixels, while also attaining a better $(0, 2m, 3)$ -net 3D coverage, in contrast to $(1, 2m, 3)$ -nets with Sobol'.

Furthermore, upon careful inspection of the matrices in Eq. (8), we find that the two non-identity matrices invert each other, which implies that, at power-of-four-octaves, the sequence actually has identical projections on the (x, y) , (y, z) , and (z, x) planes: maximal symmetry, as may be observed in the enlarged plots of Fig. 1(c2). Different projections may be obtained where needed via Owen scrambling. As for 1D projections, even though the O2m3 generator matrices in Eq. (8) are not upper-triangular, the sequence still maintains all ranks over power-of-4-point blocks, hence their 1D masks are also indistinguishable from the original Z construction. To summarize, at least for power-of-4 sample counts, the O2m3 sequence provides a neat 3D extension to the original Z samplers. Compared to the original model with three 1D masks and a single 2D mask, we retain the same quality, and provide two more optimal pairwise masks, plus an optimal 3D mask.

5 RESULTS AND APPLICATIONS

We now present empirical evaluations of the proposed extensions. Since different readers may be interested in different aspects of the method, we organize the results by extension.

5.1 Temporal Sampling

Our interlacing model is based on providing consecutive LD samples to each pixel; hence, its effectiveness can be measured by directly comparing to the ideal target of sampling with the same Z-indexed sampler at the aggregate sampling rate accumulated over the frames. We subsequently assess different aspects.

5.1.1 Visual Comparison. In Fig. 1(a), we visually compare time-averaged frames using PerPixel interleaving vs independent Z shuffling. A closeup pixel-level view shows better quality of individual pixels, with less noise, while a distant macro view clearly favors independent shuffling. We observed the same behavior consistently across severely undersampled scenes. Fig. 4 compares the models on one scene rendered at 8 SPP and averaged over 64 consecutive

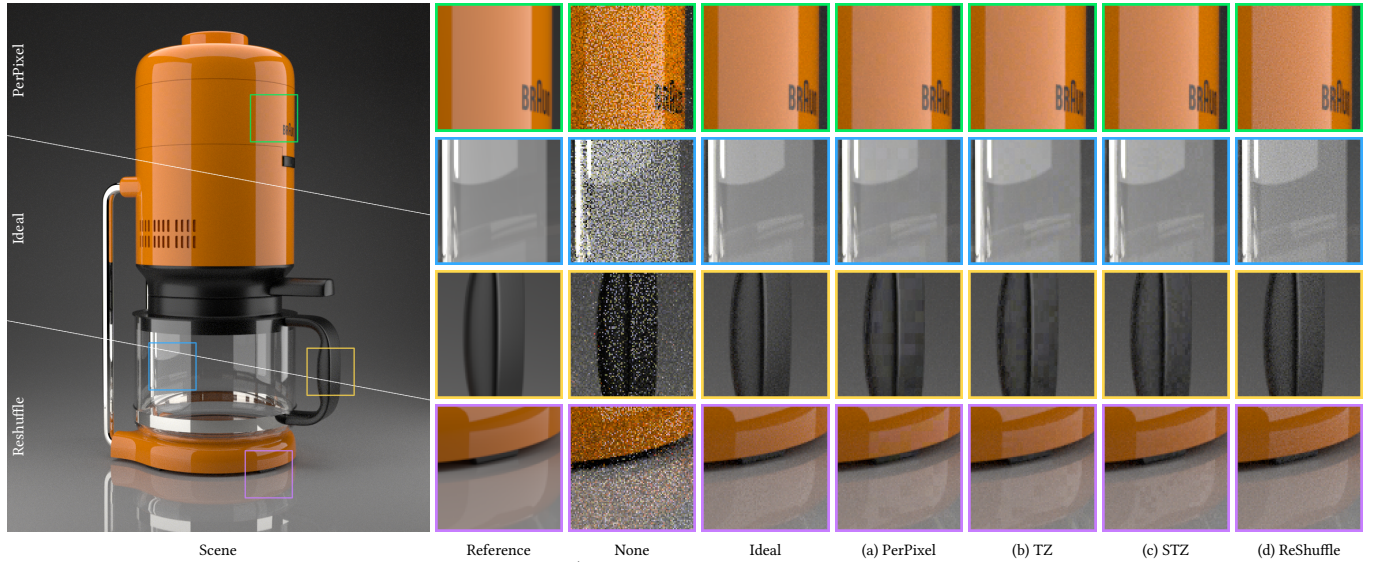


Fig. 4. A single scene demonstrating different temporal Z models sampled at 8 SPP and averaged over 64 consecutive frames. “Reference” is a sufficiently converged 1M-SPP single frame; “None” is a single 8-SPP frame, the no-temporal-accumulation baseline; and “Ideal” is a single 512-SPP (64×8) frame, the best hoped-for temporal-accumulation benchmark. Coffee Maker scene from Bitterli’s *Rendering Resources* [2016].

frames. *Ideal* is a single-frame 512-SPP (64×8) render, representing the best hoped-for benchmark for temporal accumulation. The green insets show a region where 512 SPP appears sufficient to make noise unnoticeable: PerPixel performs best, while ReShuffle is noticeably noisy. The cyan insets show a semi-converged region where TZ and STZ balance PerPixel’s pixelation against ReShuffle’s noise. The orange insets show an unconverged region favoring ReShuffle. The final purple insets combine a converged highlight with its non-converged reflection. Full-resolution images and an animated evaluation covering additional configurations and empirical measurements are provided in the supplementary materials.

5.1.2 Convergence. Besides visual inspection of the Bistro Café scene shown in Fig. 1(a), we also conducted a quantitative evaluation, as demonstrated in Fig. 5. As analytically predicted, the PerPixel model comes close to the ideal. Interlacing does not significantly degrade the quality, but reshuffling does. We also include a ZSobol reference to support our argument toward the end of Section 4.1.

5.1.3 Speed and Memory Footprint. Our first three temporal models are especially attractive in terms of computational cost. Indeed, the PerPixel model amounts to the single bitwise XOR per pixel in Eq. (3), giving a rare case where one machine instruction performs a meaningful sampling operation. ReShuffling may sound costly, but has become quite feasible with the recently introduced efficient Q-ART algorithm [Ahmed 2025] for base-4 Owen scrambling. Compared to multiple megabyte tables in the original Z sampling paper [Ahmed and Wonka 2020], Q-ART achieves a better shuffling quality using a 1 KB table. Reconstructing such a table in real time is quite feasible.

5.1.4 Discussion. Our four temporal solutions provide different trade-offs between dithering quality and convergence over time.

Each variant may suit a different application. For example, the Per-Pixel model may be used in interactive rendering applications. An important conclusion of our evaluation is that a high-dimensional sampler with ReShuffling performs similarly to a ZSobol sampler at the aggregate sampling rate, which is quite satisfactory. This combination therefore provides a good candidate for a universal sampler that performs well in both real-time and high-quality applications.

5.2 Non-Uniform Sampling

We implemented a sampler, *ZEnv*, for image-based lighting with environment maps in *pbrt-4*, based on the non-uniform sampling algorithm described in Section 4.2. At startup time, a summed area table based on image luminance is used for the hierarchical refinement algorithm, which terminates at a depth of 20, giving 1M rectangles. Sampling is then performed following Alg. 2, avoiding the binary search over a precomputed cumulative distribution function used in the default *Infinite* algorithm.

5.2.1 Convergence and Screen-Space Performance. Fig. 6 compares convergence and screen-space performance of our proposed non-uniform solution for Z sampling to the current default piecewise model of *pbrt-4*. Our model achieves a similar performance, which we find quite satisfactory for an early test. Indeed, our model is ideally indexed by a single 32-bit integer dimension of the LD sequence, saving the other dimension. Unlocking this potential, however, requires a major redesign of the renderer’s sampling infrastructure, which we leave for future research.

5.2.2 Importance Map. The proposed importance sampling model assigns sampling probabilities to regions rather than pixels, which suggests it favors slowly varying maps over those concentrating energy in small, scattered regions. This is empirically demonstrated

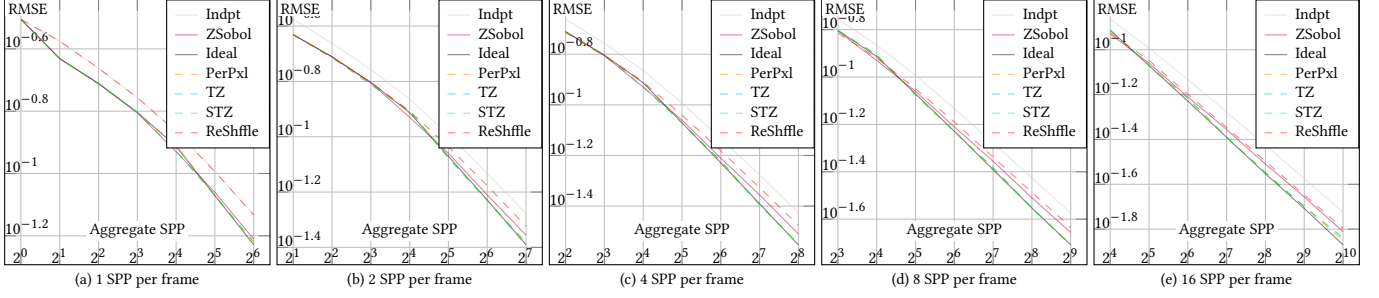


Fig. 5. Convergence of temporally averaged frames comparing different sequencing strategies, benchmarked against a single frame sampled at aggregate SPP using (Ideal) the same StART [Ahmed et al. 2026] sampler used to render the frames, and standard *pbrt-4* (Indpt) Independent and ZSobol samplers.

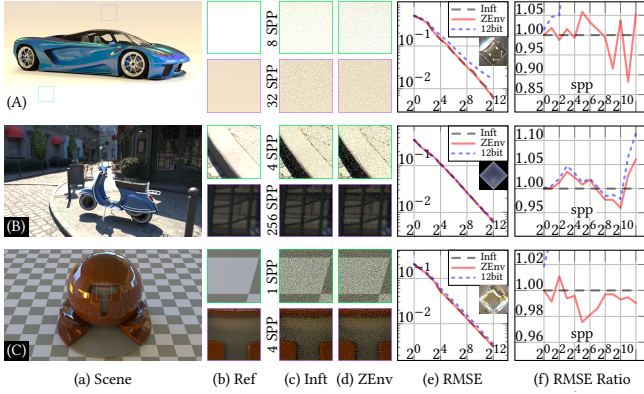


Fig. 6. Convergence and screen-space comparison of ZEnv to *pbrt-4*'s Infinite non-uniform sampler across three scenes using three Z-indexed samplers: (A) SZ [Ahmed et al. 2025], (B) O2 [Ahmed et al. 2026], and (C) *pbrt-4*'s native ZSobol on Material Test Ball scene [Bitterli 2016]. ZEnv performs nearly equivalently overall; the green/purple insets show occasional inferior/superior screen-space performance. Red/blue curves denote slicing depths 20/12, respectively; icons in RMSE plots show the lighting environment maps. The simple map in (B) permits significantly shallower slicing.

in Fig. 7, which shows the best performance for evenly distributed lighting and slight degradation for concentrated sources. We note, though, that these variations are observed with significant under-sampling, and quickly drop below 2% for a reasonable sampling rate matching the scene complexity.

5.2.3 Speed and Memory Footprint. Our model scales flexibly with memory usage. We achieved good results with as few as 4K rectangles, corresponding to a 12-bit slicing depth. This is especially useful for smoother importance maps. We also observed a 1.5% reduction in overall GPU rendering time at a 20-bit slicing depth, with slightly larger reductions at smaller depths.

5.2.4 Discussion. While we have not yet achieved significant improvements over existing models, we still advocate our proposed model for non-uniform Z sampling. It is not all about quantitative performance. The most important aspect is that recursive slicing is semantically compatible with Z sampling, mapping a binary tree to

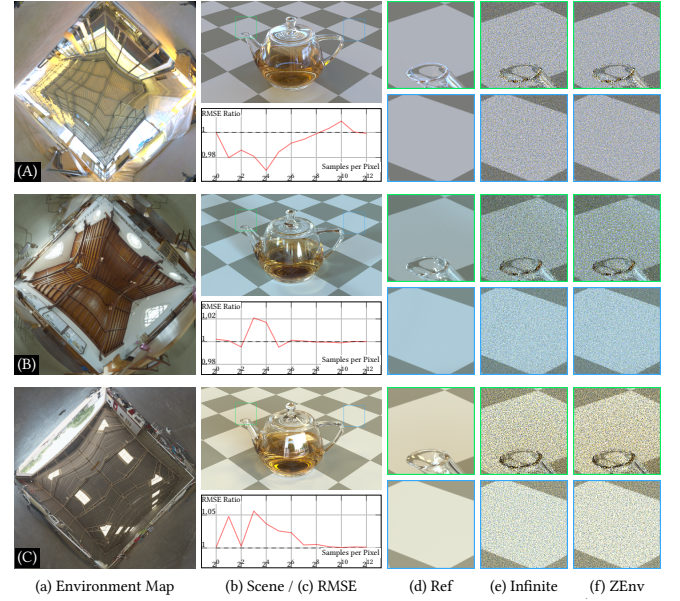


Fig. 7. Same Teapot Full scene under (A) the original [Bitterli 2016] and (B, C) alternative [Poly Haven 2026] environment maps: our non-uniform ZEnv model (A) outperforms, (B) matches, and (C) underperforms *pbrt*'s CDF-based Infinite model in the relative-convergence plots and undersampled 4-SPP screen-space insets.

a binary tree, thus providing a more reliable basis for future development. The key point is that it is more *2D-aware*: same-suffix-indexed samples belong to the same spatial region, a desirable Z-sampling feature not optimally satisfied by the conditional-marginal CDF model used for piecewise-constant 2D distributions [Pharr et al. 2023, Appendix A.5.5]. The model admits further improvements via adapting related components of the rendering pipeline, as demonstrated in Fig. 8. For example, (1) the original 1×1 map may be rotated by 45° and reassembled to 2×1 format in order to align the intrinsic map directions with the slicing axes, and (2) the sampler may be adapted to apply base-4 Owen scrambling [Ahmed 2025] to the sample index. Other possible implementation-specific improvements include special handling of too-bright objects, e.g., a single-pixel sun, that significantly distort the sliced map.

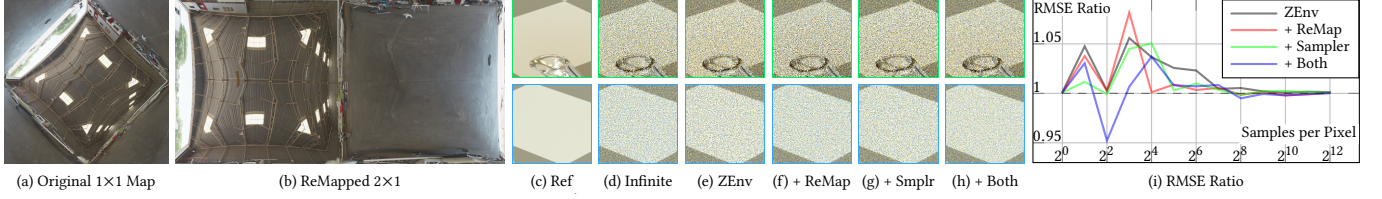


Fig. 8. The effect of different tweaks to improve the performance of ZEnv with the map in Fig. 7(C), including adapting (f) the map, (g) the sampler, and (h) both.

5.3 3D Sampling

While best suited for dithering mask construction, O2m3 is a generic LD sampling sequence that can serve other applications even beyond computer graphics. We therefore subsequently present a more extensive sample-space evaluation to benchmark its performance against established alternatives.

5.3.1 Discrepancy. Because they are comprised of dyadic $(0, 2m, 3)$ -net blocks, O2m3 sequences are LD by construction; specifically, dyadic $(1, 3)$ -sequences. We empirically benchmark their measured discrepancy against established sequences in Fig. 9. The average discrepancy of the base-3 Faure $(0, 3)$ -sequence nearly coincides with that of the base-2 Sobol' $(1, 3)$ -sequence. O2m3 oscillates below and above these two references at even and odd power-of-2 point counts, respectively, while consistently remaining below the 3D subset of the base-4 SZ $(0, 4)$ -sequence. Interestingly, its curve resonates with SZ, reflecting their shared block-matrix structure. These observations make Sobol' the most informative reference for evaluating O2m3. The oscillatory behavior of O2m3 is more pronounced in pairwise projections, which we therefore separate into two plots. At even powers of 2, the discrepancy of all three O2m3 dimension pairs coincides with that of the first Sobol' pair, a dyadic $(0, 2)$ -sequence. Thus, all three O2m3 pairwise projections consistently outperform the other two Sobol' $(1, 2)$ -sequence pairs and match the reference $(0, 2)$ behavior at these point counts. This empirically validates the main design target. At odd powers of two, however, O2m3 exhibits a more complex and suboptimal projection behavior: across successive point-count ranges, the three dimension pairs cycle between one favorable and two weaker projections. Sobol', in contrast, maintains the same projection bias over all point counts, consistently favoring the first pair of dimensions.

5.3.2 Spectra. We cross-check the findings above against the frequency power spectra of pairwise 2D projections, as shown in Fig. 10. The spectra confirm the main discrepancy findings: O2m3 is highly symmetric at even power-of-2 point counts, attaining the typical dyadic $(0, m, 2)$ -net spectra across all three pairwise projections, but becomes distorted at odd powers of two, while Sobol' remains asymmetric across dimension pairs at all power-of-2 point counts. Beyond that, the spectra reveal additional structure. For example, Sobol's bias across the 0–2 and 1–2 pairs varies slightly with the point count. More notably, at the even power-of-2 point count shown in Fig. 10(a), Sobol's distortion appears in the 0–2 pair, rather than in the 1–2 pair visually emphasized in our mask discussion in Section 4.3. A

possible explanation is that, although Sobol' behaves asymmetrically in the 1–2 pair within 4-point blocks, this asymmetry balances out over consecutive blocks. In contrast, the spectra reveal a more persistent, but less visually obvious, correlation in the 0–2 pair. This emphasizes the complementary roles of different assessment tools, each exposing blind spots of the others.

5.3.3 Numerical Integration. We have also evaluated the O2m3 sampler using a set of synthetic functions, based on the approach introduced by Jarosz et al. [2019]. If projection of an n -dimensional point p to dimensions d_i is denoted by

$$p_{d_1, \dots, d_n} = (p_{d_1}, \dots, p_{d_n}), \quad (9)$$

then n -dimensional functions can be defined using underlying 1D functions g^D using the vector norm

$$f_{d_1, \dots, d_n}^D(p) = g^D(\|p_{d_1}, \dots, p_{d_n}\|). \quad (10)$$

Three 1D functions g^D with different smoothness properties are then used:

$$g^0(r) = 1 - \text{binaryStep}(r, r_{\text{end}}), \quad (11)$$

$$g^1(r) = 1 - \text{linearStep}(r, r_{\text{start}}, r_{\text{end}}), \quad (12)$$

$$g^\infty(r) = e^{-r^2/(2\sigma^2)}, \quad (13)$$

with $r_{\text{end}} = 3/\pi$, $r_{\text{start}} = r_{\text{end}} - 0.2$, and $\sigma = 1/3$ and where plots of each one's $f_{1,2}^D$ 2D version are shown alongside.

We then measured MSE when integrating four representative functions; results are shown in Fig. 11. For the 3D function $f_{1,2,3}^D$, we find that O2m3 outperforms Sobol' at even power-of-two sample counts and that both have the same performance at odd powers of two. For $f_{1,2}^D + f_3^D$, the sum of a 2D function in the first two dimensions and a 1D function in the third, O2m3 is worse than Sobol' at odd power-of-two sample counts and matches it at even powers of two; this is to be expected, as both have good 2D projections in their first two dimensions and both are equivalent in their 1D projections, though O2m3 requires even power-of-two sample counts. However, when integrating $f_1^D + f_{2,3}^D$ where the 2D constituent uses the second and third dimensions, O2m3 is superior at even power-of-two sample counts, thanks to good 2D projections over all dimension pairs, but is occasionally inferior at its suboptimal odd power-of-two counts. Finally, as expected, both have equivalent performance when integrating the sum of three 1D functions $f_1^D + f_2^D + f_3^D$ at even power-of-two sample counts, with O2m3 worse at odd powers of two.

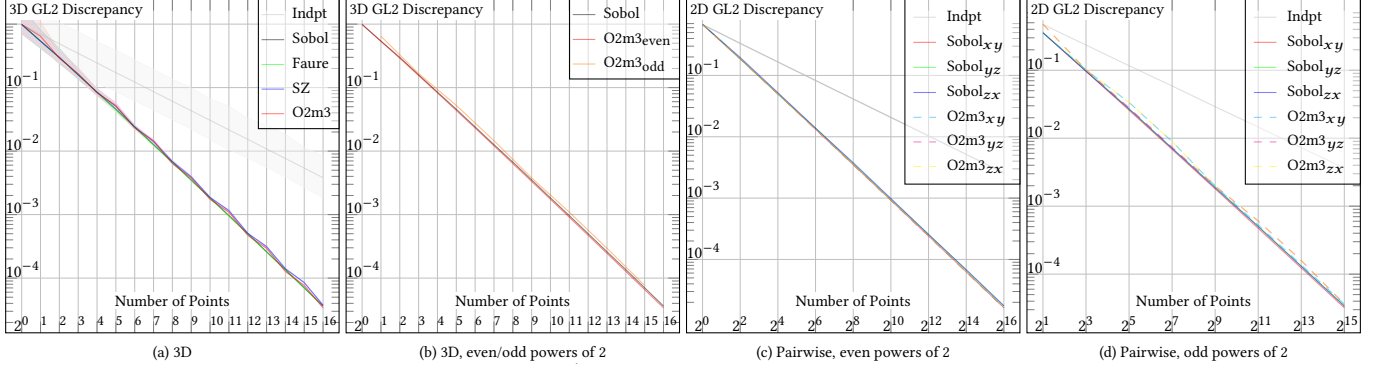


Fig. 9. Generalized L_2 discrepancy plots comparing O2m3 to common LD sequences. (a) 3D discrepancy benchmarked against the dyadic Sobol' (1, 3)-sequence, the base-3 Faure (0, 3)-sequence, and a 3D subspace of the base-4 SZ (0, 4)-sequence, with independent sampling included as a Monte Carlo baseline. (b) 3D discrepancy plot of O2m3 split between even/odd power-of-2 point counts. (c) 2D discrepancy of pairwise projections at power-of-4 point counts. (d) Pairwise discrepancy at odd power-of-2 point counts. All plots are averaged across 10,000 Owen-scrambled realizations.

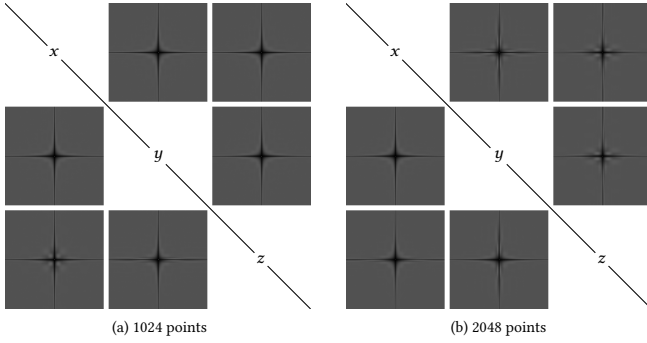


Fig. 10. Pairwise frequency power spectra comparing O2m3 (top-right) to Sobol' (bottom-left), using (a) an even and (b) an odd power-of-2 point count. Each plot is averaged across 100,000 Owen-scrambled realizations.

5.3.4 Rendering. We have also evaluated the performance of O2m3 sequences for rendering, focusing on two cases where 3D integration performance is important. We first considered the combination of motion and defocus blur, where one dimension is used for time and two to sample the lens [Cook et al. 1984], and we tested three Z-indexed sampling schemes, as demonstrated in Fig. 12: (a) the original Z-sampling approach [Ahmed and Wonka 2020] of mixing independently shuffled (0, 2)- and (0, 1)-sequences, (b) using the first three dimensions of Sobol', which form a (1, 3)-sequence, and (c) using O2m3. We found that 3D Sobol' provides a marginal benefit, roughly 2%, to the (0, 1)–(0, 2) combination, while O2m3 has $\sim 1.25\times$ lower MSE than 3D Sobol' at both odd and even power-of-two sample counts. We then experimented with subsurface scattering, where three dimensions are required to sample the BSS-RDF [King et al. 2013]. We tested with both the classical independent per-subspace shuffling [Ahmed and Wonka 2020] and the most recent high-dimensional orchestration [Ahmed et al. 2026], obtaining substantial improvements in both cases, as demonstrated in Figs. 13 and 14, respectively.

5.3.5 Inversion. Retrieving the index of a given sample is useful for adaptive and importance sampling, mesh construction, and applications that require locating nearby samples. Easy inversion is therefore a valuable feature of a sequence. Our first attempt toward an invertible O2m3 variant was to seek a self-similar construction, analogous to 2D ξ -sequences [Ahmed et al. 2023b]. In such a construction, the point p_{8i} would lie at half the coordinates of p_i , so that the first octant is a scaled copy of the full sequence. After a few failed searches, however, we realized that this power-of-8 self-similarity is impossible: the first octant of the first 64-point set is a (0, 3, 3)-net by construction, whereas the first eight points are not. We still believe that self-similar O2m3 sequences may exist over power-of-64 octaves, but the search space grows significantly, so we leave this direction for future work. Instead, we successfully factored the matrices in Eq. (8) into a tuple of finite-row matrices:

$$\begin{pmatrix} \text{Matrix 1} & \text{Matrix 2} & \text{Matrix 3} \end{pmatrix}. \quad (14)$$

The factored-out upper-triangular matrix on the right applies only a rank-preserving Faure–Tezuka [2002] shuffle, and can therefore be dropped. The key property of the remaining finite-row tuple is that the leading rows of the three matrices interleave into an invertible 3×3 -block matrix that maps between a Morton index Z and a bit-reversed sequence number V ,

$$(C_{V \rightarrow Z}, C_{Z \rightarrow V}) = \begin{pmatrix} \text{Matrix 1} & \text{Matrix 2} \\ \text{Matrix 3} & \text{Matrix 4} \end{pmatrix}. \quad (15)$$

Consequently, all samples in a 2^{-3m} -volume voxel share the same $3m$ trailing bits of their indices.

5.3.6 Speed and Memory Footprint. Although this is not obvious from the canonical tuple in Eq. (8), the apparent self-similar structure of the finite-row tuple in Eq. (14) suggests a diagonal factorization [Ahmed 2024], which we manually derived as

$$\begin{pmatrix} \text{Matrix 1} & \text{Matrix 2} & \text{Matrix 3} \end{pmatrix} \begin{pmatrix} \text{Matrix 4} & \text{Matrix 5} & \text{Matrix 6} \end{pmatrix}. \quad (16)$$

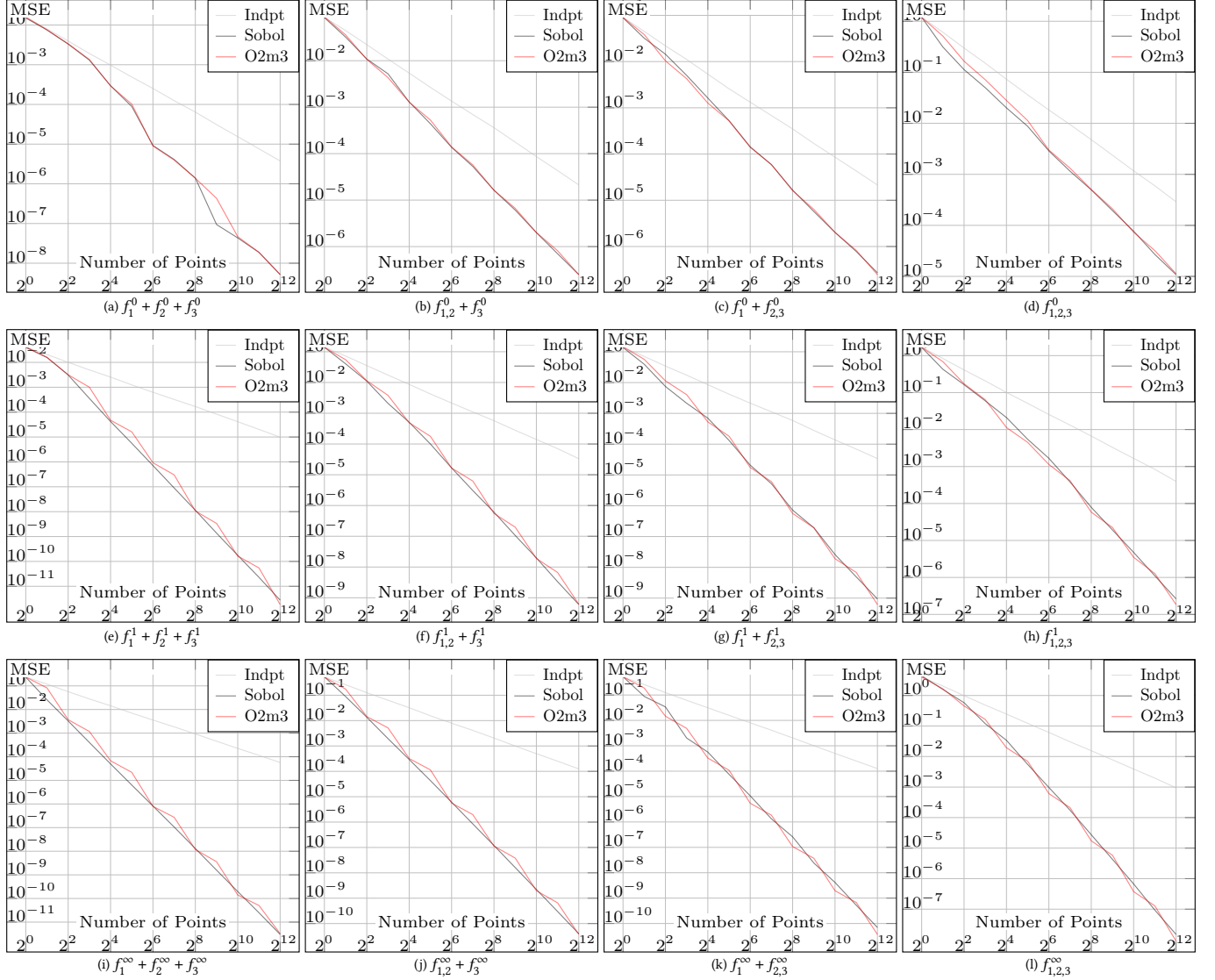


Fig. 11. Mean squared error when integrating synthetic functions of various dimensionalities using independent random, Sobol', and O2m3 samplers. With smoother functions (f^1 and f^∞), the LD samplers are especially effective compared to independent random sampling. O2m3 has superior performance to Sobol' at even power-of-two sample counts when integrating fully 3D functions and when integrating 2D functions over dimensions other than the first two. Sobol' gives equivalent or better performance at odd power-of-two sample counts and for dimension configurations where both samplers are equivalent (2D in the first two dimensions or purely 1D).

This factorization enables very fast table-free matrix-vector multiplication, reducing the evaluation time complexity from $O(m)$ to $O(\log m)$ for m -bit numerical precision [Ahmed 2024]; see Fig. 15. While equally fast algorithms are available for separately evaluating the split 2D and 1D constituents in classic Z sampling, O2m3 reduces the overall evaluation cost by saving a second Z-index shuffle for the split constituents.

5.3.7 Discussion. The results above may at first appear to indicate inconsistent behavior of O2m3 sequences. On closer inspection, however, they convey a consistent message: despite their binary

implementation, O2m3 sequences have an intrinsic base-4 structure. At odd powers of two, we are effectively using a suboptimal subset, much like taking a non-power-of-two, non-contiguous, or non-aligned subset of a 2D Sobol' sequence. An odd-power-of-2 O2m3 set can be viewed as the union of two optimal power-of-4 sets, but these two sets are not independent and need not combine optimally. Despite this larger effective base, O2m3 remains a practical sampling sequence, combining balanced joint 3D coverage and balanced 2D and 1D projections with efficient bitwise implementation.

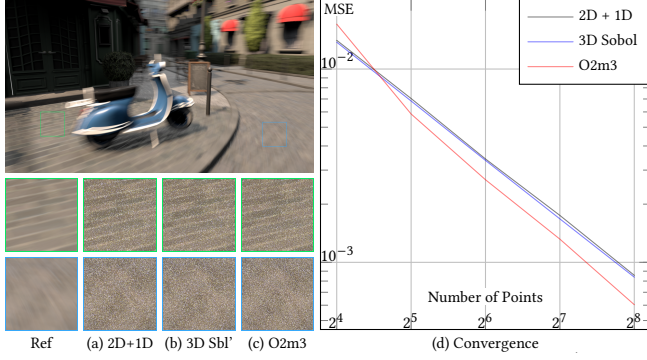


Fig. 12. Screen-space (at 16 SPP) and convergence performance of O2m3 for sampling a depth-of-field + motion-blur combination, compared to the native *pbrt-4* ZSobol' sampler using a 2D+1D combination, and to 3D Sobol'.

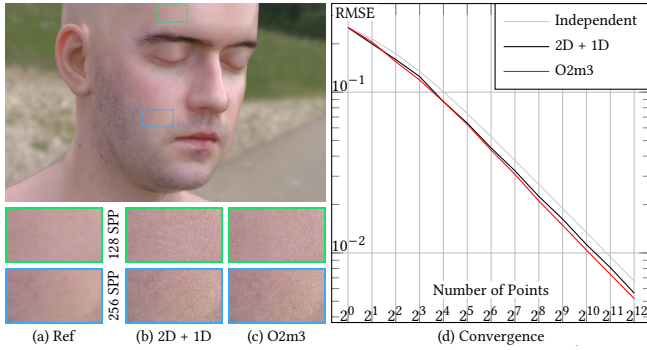


Fig. 13. A scene with subsurface materials comparing O2m3 3D sampling to the native ZSobol' sampler using a (2+1)D combination instead.

6 CONCLUSION AND FUTURE WORK

In this work, we have expanded the application of Z sampling techniques to three-dimensional problems and non-uniform sampling. We have shown that a straightforward extension to the algorithm decorrelates temporal samples, benefiting from the same multi-level stratification that Z sampling provides in the image plane due to the scrambled Morton ordering of sample indices. By applying a base-4 construction, as was done successfully to derive SZ sequences, we have further introduced a novel low-discrepancy sequence in 3D that provides $(0, 2m, 3)$ -net distributions for all consecutive blocks of 2^{2m} points, for all m . This sequence gives superior performance to 3D Sobol' sequences for rendering tasks involving integration over three dimensions, given even power-of-two numbers of samples.

In future work, we are interested in applying these insights from Z sampling to dithering masks for diffusion models where bias may not be visually noticeable in final results, but potentially propagates into machine-learned content.

ACKNOWLEDGMENTS

This work was supported in part by Guangdong S&T Program (2024B0101050004), ICFRT (W2441020), Shenzhen S&T Program (KQTD20210811090044003, KJZD20240903100022028), and Scientific

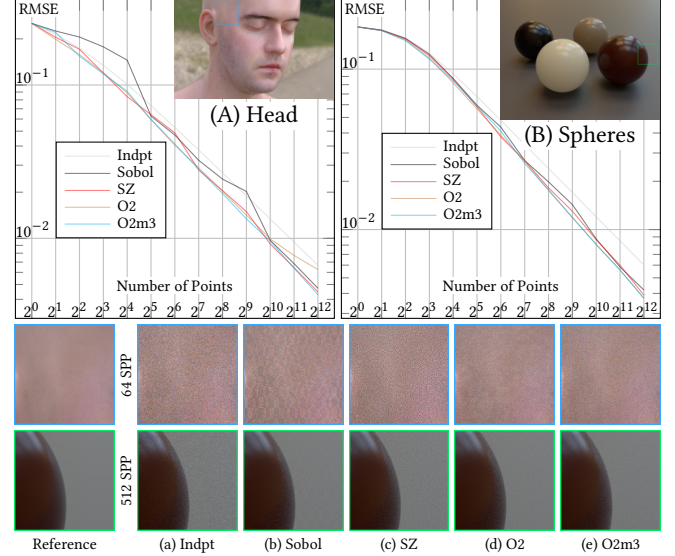


Fig. 14. Two scenes with subsurface materials comparing O2m3 to global Sobol' and Z-indexed SZ samplers, and to the recent NILE-O2, which effectively embeds 3D Sobol' under SZ [Ahmed et al. 2026]. We similarly embed O2m3 for better high-dimensional coverage. The plots and insets respectively show excellent O2m3 convergence and screen-space performance.

```
inline uint32_t x02m3(uint32_t V) {
    V ^= ((V << 2) & 0xF3CF3CF3) ^ ((V << 3) & 0x41041041);
    V ^= ((V << 4) & 0x3F03F03F);
    V ^= ((V << 8) & 0x0FFF000F);
    V ^= ((V << 16) & 0x00FFFF);
    return V;
}

inline uint32_t y02m3(uint32_t V) {
    return x02m3(
        ((V >> 1) & 0x45145145) ^
        (V & 0xB5EB5EB5) ^
        ((V << 1) & 0xA28A28A2)
    );
}

inline uint32_t z02m3(uint32_t V) {
    return x02m3(
        ((V >> 1) & 0x45145145) ^
        (V & 0x7AD7AD7A) ^
        ((V << 1) & 0xA28A28A2)
    );
}
```

Fig. 15. All of the code needed to evaluate an O2m3 sample from a bit-reversed index V . The y and z coordinate functions differ only in a single constant.

Development Fund from Guangdong Provincial Key Laboratory of Visual Media and Multidimensional Intelligence.

REFERENCES

- Abdalla G. M. Ahmed. 2014. Modular Line-Based Halftoning via Recursive Division. In *Proceedings of the Workshop on Non-Photorealistic Animation and Rendering (NPAR '14)*. ACM, 41–48. <https://doi.org/10.1145/2630397.2630403>
- Abdalla G. M. Ahmed. 2024. An Implementation Algorithm of 2D Sobol' Sequence: Fast, Elegant, and Compact. In *Eurographics Symposium on Rendering (EGSR)*. The Eurographics Association. <https://doi.org/10.2312/sr.20241147>
- Abdalla G. M. Ahmed. 2025. Q-ART Owen Scrambling. In *Computer Graphics and Visual Computing (CGVC)*. The Eurographics Association. <https://doi.org/10.2312/cgvc.20251215>
- Abdalla G. M. Ahmed, Matt Pharr, Victor Ostromoukhov, and Hui Huang. 2025. SZ Sequences: Binary-Constructed $(0, 2^q)$ -Sequences. *ACM Trans. Graph.* 44, 6, Article 206 (Dec. 2025). <https://doi.org/10.1145/3763272>
- Abdalla G. M. Ahmed, Matt Pharr, Victor Ostromoukhov, and Hui Huang. 2026. NILE: Nested Interleaving of Low-Dimensional Elements. *ACM Trans. Graph.* 45, 4, Article 105 (July 2026). <https://doi.org/10.1145/3811283>

- Abdalla G. M. Ahmed, Matt Pharr, and Peter Wonka. 2023a. ART-Owen Scrambling. *ACM Trans. Graph.* 42, 6, Article 258 (Dec. 2023). <https://doi.org/10.1145/3618307>
- Abdalla G. M. Ahmed, Mikhail Skopenkov, Markus Hadwiger, and Peter Wonka. 2023b. Analysis and Synthesis of Digital Dyadic Sequences. *ACM Trans. Graph.* 42, 6, Article 218 (Dec. 2023). <https://doi.org/10.1145/3618308>
- Abdalla G. M. Ahmed and Peter Wonka. 2020. Screen-Space Blue-Noise Diffusion of Monte Carlo Sampling Error via Hierarchical Ordering of Pixels. *ACM Trans. Graph.* 39, 6, Article 244 (Nov. 2020). <https://doi.org/10.1145/3414685.3417881>
- Abdalla G. M. Ahmed and Peter Wonka. 2021. Optimizing Dyadic Nets. *ACM Trans. Graph.* 40, 4, Article 141 (July 2021). <https://doi.org/10.1145/3450626.3459880>
- Bryce E. Bayer. 1973. An Optimum Method for Two-Level Rendition of Continuous Tone Pictures. In *IEEE International Conference on Communications, June, 1973*, Vol. 26.
- Benedikt Bitterli. 2016. Rendering Resources. <https://benedikt-bitterli.me/resources/> Accessed: 2025-11-08.
- Nicolas Bonneel, David Coeurjolly, Jean-Claude Iehl, and Victor Ostromoukhov. 2025. Sobol' Sequences with Guaranteed-Quality 2D Projections. *ACM Trans. Graph.* 44, 4, Article 97 (July 2025). <https://doi.org/10.1145/3730821>
- Paul Bratley and Bennett L. Fox. 1988. Algorithm 659: Implementing Sobol's Quasirandom Sequence Generator. *ACM Transactions on Mathematical Software (TOMS)* 14, 1 (1988), 88–100.
- Brent Burley. 2020. Practical Hash-based Owen Scrambling. *Journal of Computer Graphics Techniques (JCGT)* 9, 4 (Dec. 2020), 1–20. <http://jcgt.org/published/0009/04/01/>
- Petrik Clarberg, Wojciech Jarosz, Tomas Akenine-Möller, and Henrik Wann Jensen. 2005. Wavelet Importance Sampling: Efficiently Evaluating Products of Complex Functions. *ACM Trans. Graph.* 24, 3 (2005), 1166–1175.
- Robert L. Cook. 1986. Stochastic Sampling in Computer Graphics. *ACM Trans. Graph.* 5, 1 (Jan. 1986), 51–72. <https://doi.org/10.1145/7529.8927>
- Robert L. Cook, Thomas Porter, and Loren Carpenter. 1984. Distributed Ray Tracing. In *Proceedings of SIGGRAPH*. ACM, 137–145. <https://doi.org/10.1145/800031.808590>
- Franklin C. Crow. 1984. Summed-Area Tables for Texture Mapping. In *Proceedings of SIGGRAPH*. ACM, New York, NY, USA, 207–212. <https://doi.org/10.1145/800031.808600>
- Mark A. Z. Dippé and Erling Henry Wold. 1985. Antialiasing through Stochastic Sampling. In *Proceedings of SIGGRAPH*. ACM, 69–78. <https://doi.org/10.1145/325334.325182>
- William Donnelly, Alan Wolfe, Judith Bütetage, and Jon Valdés. 2024. FAST: Filter-Adapted Spatio-Temporal Sampling for Real-Time Rendering. *Proc. ACM Comput. Graph. Interact. Tech.* 7, 1, Article 13 (May 2024). <https://doi.org/10.1145/3651283>
- Henri Faure. 1982. Discrepance de Suites Associées à un Système de Numération (en Dimension s). *Acta Arithmetica* 41, 4 (1982), 337–351. <http://eudml.org/doc/205851>
- Henri Faure and Shu Tezuka. 2002. Another Random Scrambling of Digital (t,s)-Sequences. In *Monte Carlo and Quasi-Monte Carlo Methods 2000*. Springer, 242–256.
- R. Floyd and L. Steinberg. 1976. An Adaptive Algorithm for Spatial Grayscale. *Proceedings of the Society for Information Display* 17, 2 (1976), 75–77.
- Iliyan Georgiev and Marcos Fajardo. 2016. Blue-Noise Dithered Sampling. In *ACM SIGGRAPH 2016 Talks*. 1–1.
- Leonhard Grünschloß and Alexander Keller. 2009. (t, m, s)-Nets and Maximized Minimum Distance, Part II. In *Monte Carlo and Quasi-Monte Carlo Methods 2008*. Springer, 395–409.
- Leonhard Grünschloß, Matthias Raab, and Alexander Keller. 2012. Enumerating Quasi-Monte Carlo Point Sequences in Elementary Intervals. In *Monte Carlo and Quasi-Monte Carlo Methods 2010*. Springer, 399–408.
- John H. Halton. 1960. On The Efficiency Of Certain Quasi-Random Sequences Of Points In Evaluating Multi-Dimensional Integrals. *Numer. Math.* 2 (1960), 84–90.
- J. M. Hammersley and K. W. Morton. 1956. A New Monte Carlo Technique: Antithetic Variates. *Mathematical Proceedings of the Cambridge Philosophical Society* 52, 3 (1956), 449–475.
- Eric Heitz and Laurent Belcour. 2019. Distributing Monte Carlo Errors as a Blue Noise in Screen Space by Permuting Pixel Seeds Between Frames. In *Computer Graphics Forum*, Vol. 38. Wiley Online Library, 149–158.
- Eric Heitz, Laurent Belcour, Victor Ostromoukhov, David Coeurjolly, and Jean-Claude Iehl. 2019. A Low-Discrepancy Sampler That Distributes Monte Carlo Errors as a Blue Noise in Screen Space. In *ACM SIGGRAPH 2019 Talks (SIGGRAPH '19)*. ACM, Article 68. <https://doi.org/10.1145/3306307.3328191>
- Andrew Helmer, Per Christensen, and Andrew Kensler. 2021. Stochastic Generation of (t, s) Sample Sequences. In *Eurographics Symposium on Rendering - DL-only Track*. The Eurographics Association. <https://doi.org/10.2312/sr.20211287>
- Wojciech Jarosz, Afnan Enayet, Andrew Kensler, Charlie Kilpatrick, and Per Christensen. 2019. Orthogonal Array Sampling for Monte Carlo Rendering. In *Computer Graphics Forum*, Vol. 38. Wiley Online Library, 135–147.
- Alexander Keller. 1996. Quasi-Monte Carlo Radiosity. In *Proceedings of the Eurographics Workshop on Rendering Techniques '96* (Porto, Portugal). Springer-Verlag, 101–110.
- Alexander Keller. 2004. Stratification by Rank-1 Lattices. In *Monte Carlo and Quasi-Monte Carlo Methods 2002*. Springer, 299–313.
- Alexander Keller. 2006. Myths of Computer Graphics. In *Monte Carlo and Quasi-Monte Carlo Methods 2004*. Springer, 217–243.
- Alexander Keller. 2013. Quasi-Monte Carlo Image Synthesis in a Nutshell. In *Monte Carlo and Quasi-Monte Carlo Methods 2012*. Springer, 213–249.
- Alan King, Christopher Kulla, Alejandro Conty, and Marcos Fajardo. 2013. BSSRDF Importance Sampling. In *ACM SIGGRAPH Talks*.
- Thomas Kolzig and Alexander Keller. 2002. Efficient Multidimensional Sampling. In *Computer Graphics Forum*, Vol. 21. 557–563.
- Gerhard Larcher and Friedrich Pillichshammer. 2001. Walsh Series Analysis of the L_2 -Discrepancy of Symmetrized Point Sets. *Monatshefte für Mathematik* 132, 1 (2001), 1–18.
- G Larcher and F Pillichshammer. 2003. Sums of Distances to the Nearest Integer and the Discrepancy of Digital Nets. *Acta Arithmetica* 106 (2003), 379–408.
- Don Mitchell. 1992. Ray Tracing and Irregularities of Distribution. In *Third Eurographics Workshop on Rendering Proceedings*. 61–69.
- Don P. Mitchell. 1987. Generating Antialiased Images at Low Sampling Densities. In *Proceedings of SIGGRAPH*. ACM, 65–72. <https://doi.org/10.1145/37401.37410>
- G. M. Morton. 1966. A Computer Oriented Geodetic Data Base; and a New Technique in File Sequencing. *International Business Machines Technical Report* (1966).
- Jacob Munkberg, Petrik Clarberg, Jon Hasselgren, Robert Toth, Masamichi Sugihara, and Tomas Akenine-Möller. 2011. Hierarchical Stochastic Motion Blur Rasterization. In *Proceedings of the ACM SIGGRAPH Symposium on High Performance Graphics*. ACM, 107–118. <https://doi.org/10.1145/2018323.2018341>
- Harald Niederreiter. 1987. Point Sets and Sequences with Small Discrepancy. *Monatshefte für Mathematik* 104, 4 (1987), 273–337.
- Harald Niederreiter. 1992. *Random Number Generation and Quasi-Monte Carlo Methods*. SIAM.
- Harald Niederreiter. 2005. Constructions of (t,m,s)-nets and (t,s)-sequences. *Finite Fields and Their Applications* 11, 3 (2005), 578–600. <https://doi.org/10.1016/j.fffa.2005.01.001>
- Art B. Owen. 1995. Randomly Permuted (t, m, s)-Nets and (t, s)-Sequences. In *Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing*. Springer, 299–317.
- Art B. Owen. 1998. 'Scrambling Sobol' and Niederreiter-Xing Points. *Journal of Complexity* 14, 4 (1998), 466–489.
- Art B. Owen. 2003. Quasi-Monte Carlo Sampling. *Monte Carlo Ray Tracing: SIGGRAPH Course Notes* (2003), 69–88.
- A. Cengiz Öztireli. 2020. A Comprehensive Theory and Variational Framework for Anti-aliasing Sampling Patterns. *Computer Graphics Forum* 39, 4 (2020), 133–148. <https://doi.org/10.1111/cgf.14059>
- Lois Paulin, Nicolas Bonneel, David Coeurjolly, Jean-Claude Iehl, Alexander Keller, and Victor Ostromoukhov. 2022. MatBuilder: Mastering Sampling Uniformity over Projections. *ACM Trans. Graph.* 41, 4, Article 84 (July 2022). <https://doi.org/10.1145/3528223.3530063>
- Matt Pharr, Wenzel Jakob, and Greg Humphreys. 2023. *Physically Based Rendering: From Theory to Implementation* (4th ed.). MIT Press.
- Poly Haven. 2026. Church Museum and Hangar Interior HDRIs. <https://polyhaven.com/hdris> Accessed 8 September 2026.
- Peter Shirley. 1991. Discrepancy as a Quality Measure for Sample Distributions. In *Proc. Eurographics '91*, Vol. 91. 183–194.
- Il'ya Meerovich Sobol'. 1967. 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.
- Il'ya Meerovich Sobol', Danil Asotsky, Alexander Kreinin, and Sergei Kucherenko. 2011. Construction and Comparison of High-Dimensional Sobol' Generators. *Wilmott* 2011, 56 (2011), 64–79. <https://doi.org/10.1002/wilm.10056>
- Robert Ulichney. 1987. *Digital Halftoning*. MIT Press.
- R.A. Ulichney. 1988. Dithering with Blue Noise. *Proc. IEEE* 76, 1 (Jan 1988), 56–79. <https://doi.org/10.1109/5.3288>
- Robert A. Ulichney. 1993. Void-and-Cluster Method for Dither Array Generation. In *IS&T/SPIE's Symposium on Electronic Imaging: Science and Technology*. International Society for Optics and Photonics, 332–343.
- J.G. van der Corput. 1935. Verteilungsfunktionen. *Proceedings of the Nederlandse Akademie van Wetenschappen* 38 (1935), 813–821.
- Eric Veach. 1997. *Robust Monte Carlo Methods for Light Transport Simulation*. Ph.D. Dissertation. Stanford University.
- Alan Wolfe, William Donnelly, and Henrik Halén. 2025. Importance-Sampled Filter-Adapted Spatio-Temporal Sampling. *Journal of Computer Graphics Techniques (JCGT)* 14, 1 (9 May 2025), 174–184. <http://jcgt.org/published/0014/01/08/>
- Alan Wolfe, Nathan Morrical, Tomas Akenine-Möller, and Ravi Ramamoorthi. 2022. Spatiotemporal Blue Noise Masks. In *Eurographics Symposium on Rendering*. The Eurographics Association. <https://doi.org/10.2312/sr.20221161>
- J. I. Yellott. 1983. Spectral Consequences of Photoreceptor Sampling in the Rhesus Retina. *Science* 221 (1983), 382–385.