Sliced Optimal Transport Sampling

Lois Paulin, Nicolas Bonneel, David Coeurjolly, Jean-Claude Iehl, Antoine Webanck, Mathieu Desbrun, Victor Ostromoukhov

Sliced Optimal Transport Sampling

LOIS PAULIN, Univ Lyon, CNRS
NICOLAS BONNEEL, Univ Lyon, CNRS
DAVID COEURJOLLY, Univ Lyon, CNRS
JEAN-CLAUDE IEHL, Univ Lyon, CNRS
ANTOINE WEBANCK, Univ Lyon, CNRS
MATHIEU DESBRUN, ShanghaiTech/Caltech
VICTOR OSTROMOUKHOV, Univ Lyon, CNRS

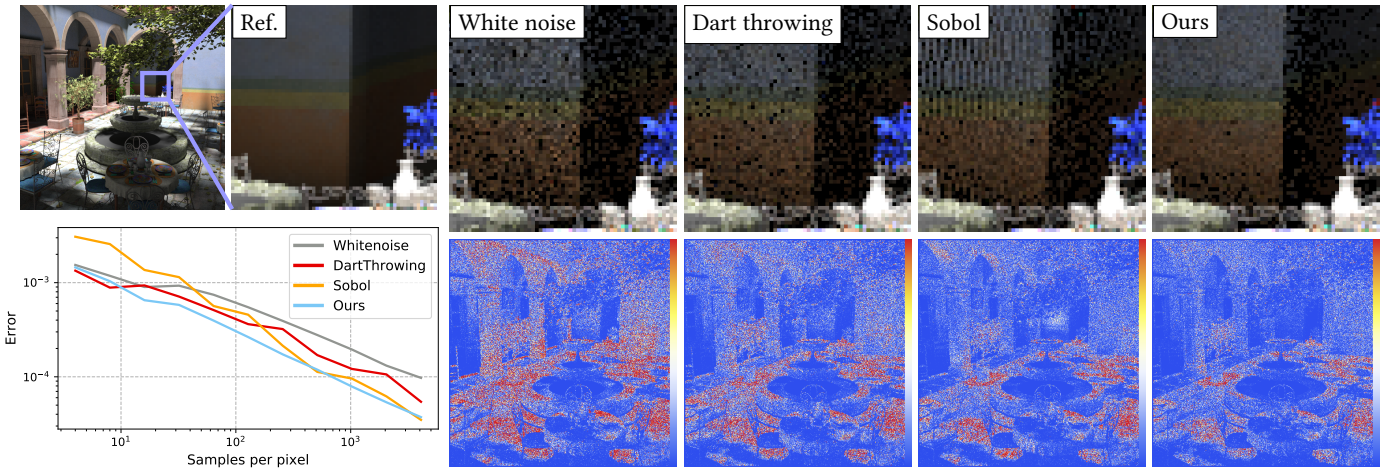


Fig. 1. Sliced Optimal Transport Sampling. Global illumination of a scene (top left, San Miguel) requires integrating radiance over a high-dimensional space of light paths. The projective variant of our sliced optimal transport (SOT) sampling technique, leveraging the particular nature of integral evaluation in rendering and further combined with a micro-Cranley-Patterson rotation per pixel, outperforms standard Monte Carlo and Quasi-Monte Carlo techniques, exhibiting less noise and no structured artifact (top right, 32spp) while offering a better spatial distribution of error (bottom right, errors from blue (small) to red (large)). Moreover, our projective SOT sampling produces better convergence of the mean absolute error for the central 7×7 zone of the highlighted reference window as a function of the number of samples per pixel (from 4spp to 4096spp, bottom-left graph) in the case of indirect lighting with one bounce.

In this paper, we introduce a numerical technique to generate sample distributions in arbitrary dimension for improved accuracy of Monte Carlo integration. We point out that optimal transport offers theoretical bounds on Monte Carlo integration error, and that the recently-introduced numerical framework of sliced optimal transport (SOT) allows us to formulate a novel and efficient approach to generating well-distributed high-dimensional pointsets. The resulting sliced optimal transport sampling, solely involving repeated 1D solves, is particularly simple and efficient for the common case of a uniform density over a d-dimensional ball. We also construct a volume-preserving map from a d-ball to a d-cube (generalizing the Shirley-Chiu mapping to arbitrary dimensions) to offer fast SOT sampling over d-cubes. We provide ample numerical evidence of the improvement in Monte Carlo integration accuracy that SOT sampling brings compared to existing QMC techniques, and derive a projective variant for rendering which rivals, and at times outperforms, current sampling strategies using low-discrepancy sequences or optimized samples.

CCS Concepts: • **Computing methodologies → Computer graphics**.

Additional Key Words and Phrases: multidimensional sampling, optimal transport, Radon transform, Monte Carlo integration, point distributions

ACM Reference Format:
Lois Paulin, Nicolas Bonneel, David Coeurjolly, Jean-Claude Iehl, Antoine Webanck, Mathieu Desbrun, and Victor Ostromoukhov. 2020. Sliced Optimal Transport Sampling. *ACM Trans. Graph.* 39, 4, Article 1 (July 2020), 17 pages. https://doi.org/10.1145/3386569.3392395

Authors' addresses: Lois Paulin, Nicolas Boneel, David Coeurjolly, Jean-Claude Iehl, Antoine Webanck, Victor Ostromoukhov: CNRS/LIRIS, Lyon, France; Mathieu Desbrun: California Institute of Technology, Pasadena, CA, USA, on sabbatical at SIST in ShanghaiTech University, Shanghai, China.

0730-0301/2020/7-ART1 $15.00
https://doi.org/10.1145/3386569.3392395

1 INTRODUCTION

The need to evaluate integrals of high-dimensional signals arises in a number of applications such as finance or machine learning. It is particularly crucial in global illumination where the radiance through a pixel must be integrated across the multidimensional space of possible light transport paths. Monte Carlo integration, which approximates an integral through averaging the values of

its integrand evaluated at n discrete locations within the integration domain, is often the only practical technique to handle this challenging numerical task.

While the original Monte Carlo (MC) technique relies on *random* sample locations, the resulting approximation error is often greatly affected by sample clumping: intuitively, the amount of information gathered by each sample on the integrand should be maximized, so clumped samples are wasteful. Since the simple choice of regular grid distributions (possibly jittered to avoid aliasing) is ill-suited to the case of high dimensions, several efforts to define the notion of "well-distributed" sampling followed. Guided by a known bound on integration error depending on a measure of spatial uniformity of the samples called *discrepancy*, quasi-Monte Carlo (QMC) methods proposed the use of specifically-crafted deterministic sample locations, typically generated via low-discrepancy sequences. This family of approaches is currently the cornerstone of most rendering algorithms as it greatly improves the integration accuracy for a fixed number of samples. More recently, spectral statistics of the spatial distribution of samples have also been proven key to reducing the approximation error, suggesting that carefully optimized sampling locations can further improve upon low-discrepancy sequences. Since optimizing sample distributions over high dimensional domains is computationally challenging, variational approaches to sampling (where samples are optimized to minimize a select measure of uniformity) have only been offered in low dimensions so far. Yet, none specifically targets the reduction of Monte Carlo integration error, focusing instead of stippling or halftoning applications.

In this paper, we argue for the design of high-dimensional sampling via optimal transport, and provide an efficient and practical sample distribution optimization method in arbitrary dimensions through *sliced optimal transport*, solely involving repeated 1D solves. We provide ample numerical evidence of improved accuracy of Monte Carlo integrations, and derive a projective variant which rivals, and at times outperforms, current rendering strategies.

1.1 Background

We begin with a brief review of previous works on the generation of sample distributions (i.e., pointsets) for integration purposes.

Poisson disk sampling. A common approach to generating well-spread point distributions is through Poisson disk sampling, which guarantees that two points are separated by at least a given minimum distance. While fast to generate using dart throwing techniques [Bridson 2007], Poisson disk samples suffer from an inherent white noise component [Torquato et al. 2006] that gravely weakens its relevance to integration purposes in high dimension.

Low-discrepancy distributions. While a multitude of quadrature methods for numerical integration exist in two or three dimensions, Monte Carlo (MC) integration has remained the method of choice for the evaluation of high-dimensional integrals: its stochastic nature offers an expected accuracy independent of the dimensionality of the domain of integration. However, this same randomness in the locations of samples often leads to large approximation errors, unless many samples are employed. A better understanding of this issue can be gained by considering the Koksma-Hlawka inequality, which states that the accuracy of MC integration over a d-dimensional unit cube $[0,1]^d$ is bounded by the product of two independent factors:

$$\left| \int f(x)dx - \frac{1}{n}\sum_{i=1}^{n} f(x_i) \right| \leq D(\{x_1, ..., x_n\}).\, \mathrm{Var}_{\mathrm{HK}}\, f, \qquad (1)$$

where $D(.)$ is the so-called *discrepancy* of a point distribution measuring its spatial uniformity, while $\mathrm{Var}_{\mathrm{HK}}\, f$ measures the variation of the function f in the sense of Hardy and Krause (a multidimensional extension of total variation), see [Hlavka 1961]. Evaluating this discrepancy is particularly costly as it requires finding the worst-case density deviation from a uniform pointset over arbitrary convex regions. As a result, simpler notions of discrepancy (like the star discrepancy D^* involving only boxes with a corner at 0, the extreme discrepancy, or the L_2 discrepancy) have been proposed, still offering valid — yet less tight — error bounds. Finally, note that Eq. (1), and thus the notion of discrepancy, is only useful in bounding the error for functions of finite Hardy-Krause variation; in particular, MC integration of discontinuous multivariate functions may result in large errors, while typical functions may have actual errors far lower than the provided upper bound. When no a-priori knowledge is available on the function f, the integration error being bounded by discrepancy implies that using sample distributions with *low discrepancy* is a reliable way to bound the expected MC error. Fortunately, there exists by now a series of sequences (deterministic enumeration of samples) which offer low discrepancy by construction, the most popular ones being Sobol, Halton, Faure, van der Corput, and Niederreiter sequences (see recent surveys in [Lemieux 2009; Dick and Pillichshammer 2010; Keller 2013]). The term quasi-Monte Carlo (QMC) refers to methods based on these low-discrepancy sequences for MC integration. As evidenced by their wide use in modern rendering engines such as PBRT [Pharr et al. 2016] or Mitsuba [Jakob 2013], QMC approaches are considered state-of-the art for error minimization in radiance estimation.

Variational blue noise distributions. Another notion of equidistribution of point samples was introduced by Robert Ulichney [Ulichney 1987] in terms of spectral content: a pointset is said to exhibit "blue noise" characteristics if its isotropic power spectrum has no low-frequency components, has a peak at a characteristic frequency (representing the inverse of the mean distance between points), and is flat (white noise) at high frequencies. Recent work theoretically confirmed that having no low frequency content in the spectrum of a sample distributions ensures better Monte Carlo integration [Durand 2011; Subr and Kautz 2013; Pilleboue et al. 2015; Öztireli 2016; Singh and Jarosz 2017; Singh et al. 2019], explaining a posteriori why a few popular sampling schemes such as Poisson-disk sampling exhibit an asymptotic behavior no better than using random points. While blue noise distributions in graphics were leveraged early on to remove the harmful aliasing artifacts of regular sampling [Dippé and Wold 1985; Cook 1986; Mitchell 1987, 1991], they regained considerable attention within the computer graphics community recently: their spatial distribution in 2D visually offers a good balance between ordered and disordered distributions [Georgiev and Fajardo 2016; Heitz and Belcour 2019; Heitz et al. 2019], making them particularly well suited for stippling and halftoning. Consequently, many *variational* approaches for generating blue noise pointsets through optimization of sample positions have been proposed over the last decade [Ostromoukhov et al. 2004; Kopf et al. 2006; Lagae and Dutré

2006; Wei 2010; Fattal 2011; Zhou et al. 2012; Heck et al. 2013]. Most germane to our work is the fact that a few variational blue noise sampling techniques define their measure of uniformity through *optimal transport* [Balzer et al. 2009; De Goes et al. 2012; Qin et al. 2017]; regrettably, their scalability in 2D and 3D does not extend to arbitrary dimensions, and no discussions on approximation errors when used for Monte Carlo integration were provided.

Projective sampling. Recent works [Kulla et al. 2018; Christensen et al. 2018a; Jarosz et al. 2019] proposed to improve uniformity of sampling distributions in low and high dimensions by starting from good 2D configurations that they then lift to higher dimensions. In sharp contrast, [Reinert et al. 2016] introduced an approach to optimize spatial positions of multidimensional pointsets while preserving spectral properties of lower-dimensional projections. Our approach will be able to accommodate this projective sampling idea (without suffering from the combinatorial explosion in high dimensions), which we will then leverage for rendering purposes.

1.2 Contributions

Designing pointsets in arbitrary dimensions to ensure good accuracy in MC integration remains a difficult goal. The sparsity of theoretical foundations is a first issue: one can only leverage knowledge on either the role of discrepancy or partial spectral properties, which limits the number of approaches that one can design to construct good sample distributions. A second issue is the need for sampling in arbitrary dimensions. While low-discrepancy sequences are naturally able to handle a high dimensional domain, their discrepancy in this case might not be smaller than for a random set for practical values of the number of samples n [Lemieux 2009]; moreover, their periodic structure is known to generate visual artifacts in rendering applications [Pharr et al. 2016].

In this paper, we point out that optimal transport offers theoretical grounds to bound MC integration that are similar, yet different from the discrepancy-based bound from Eq. (1), and that the recently-introduced numerical framework of sliced optimal transport [Pitié et al. 2005; Rabin et al. 2011; Bonneel et al. 2015; Bonneel and Coeurjolly 2019] allows us to formulate a novel and efficient approach to generating well-distributed high-dimensional pointsets. While the resulting sliced optimal transport (SOT) sampling can handle arbitrary densities over arbitrary domains through the sliced Fourier theorem, the archetypical case of a uniform density over a d-dimensional ball is particularly simple and efficient computationally. We further introduce a bi-Lipschitz volume-preserving map from the d-ball to the d-cube (generalizing the original Shirley-Chiu area-preserving disk-to-square mapping [Shirley and Chiu 1997]) to handle the d-cube case efficiently as well. We then present a wide array of numerical tests for various dimensions d to demonstrate the excellent approximation accuracy of our optimized sample distributions. Finally, we show that our technique can exploit specific properties of the integrands to further decrease approximation error: building upon the idea of projective blue noise [Reinert et al. 2016] for rendering, we can specialize SOT sampling to generate samples with both near equidistribution in the high-dimensional space *and* near equidistribution of their projections onto select linear subspaces. Rendering examples demonstrate clear improvements over standard QMC methods used in rendering engines, see Fig. 1.

2 RELEVANCE OF OPTIMAL TRANSPORT TO SAMPLING

Optimal transport (OT) is a theoretical framework useful in a variety of fields, from economics for resource allocation to mathematical physics for hydrodynamics. In its most general form, it defines a formal notion of *distance* between generalized probability density functions by evaluating an optimal transport plans between them [Villani 2009]: intuitively, if each density is viewed as a given amount of sand piled up in space, this metric is the minimum labor needed to move one pile into the other. In our sampling context, though, the simpler, but more specific notion of semi-discrete optimal transport is most relevant to our goals: it provides a simple approximation bound for Monte Carlo integration, which we will exploit for sampling generation in the remainder of this paper.

2.1 Distance of samples to spatial density

Optimal transport also allows to quantify how well a pointset matches a given density function: as points in arbitrary dimensions can be thought of as Dirac masses, OT provides a measure of how much the points need to spread out (i.e., transport their mass across space) to match the density function. This particular case is often referred to *semi-discrete optimal transport*, and when applied to a uniform density function, it defines an intuitive notion of "equidistribution" for a pointset, very different from discrepancy.

More precisely, the semi-discrete optimal transport distance between a spatial density φ within a domain $\Omega \subset \mathbb{R}^d$ and a set of n sample points $X = \{\mathbf{x}^j \in \Omega\}_{j=1..n}$ can be mathematically formulated as finding a spatial assignment π: $\Omega \to \{1,...,n\}$ of every point $\mathbf{x}$ of Ω to a sample $\mathbf{x}^{\pi(\mathbf{x})}$ that minimizes what is known as the p-Wasserstein distance $W_p(X, \varphi)$, defined as:

$$W_p(X,\varphi) \coloneqq \left(\min_{\pi} \int_{\Omega} \|\mathbf{x} - \mathbf{x}^{\pi(x)}\|^p \, \varphi(\mathbf{x}) \mathrm{d}\mathbf{x} \right)^{1/p}, \tag{2}$$

under the constraint that each sample is considered to have the same mass, i.e., $\int_{\pi^{-1}(j)} \varphi(\mathbf{x})\mathrm{d}\mathbf{x} = \frac{1}{n} \ \forall j \in \{1,...,n\}$. In the case $p = 2$, a constrained optimization of the distance $W_2(X, \varphi)$ to compute a pointset X best matching a spatial density φ is known to be efficiently achieved via power diagrams [Aurenhammer et al. 1998]. This is precisely the approach adopted by a number of recent works in graphics to compute "blue noise" pointsets sampling a given continuous density [Mérigot 2011; De Goes et al. 2012; Lévy 2015; Qin et al. 2017; Nader and Guennebaud 2018] through gradient or Newton descent; yet none of these techniques are practical in higher dimensions. Optimal transport was also used in [Rowland et al. 2018] to couple subsets of multidimensional pointsets.

2.2 Error bounds for Monte Carlo integration

The Rubinstein-Kantorovich theorem [Kantorovich 1948; Kantorovich and Rubinstein 1958] states through a duality argument that the Wasserstein distance $W_1(\mu, \nu)$ for two densities μ and ν can in fact be seen as an upper bound for finite Lipschitz functions of the difference of their integrals over these two measures:

$$W_1(\mu,\nu) = \sup_{f:\mathbb{R}^d \to \mathbb{R}} \frac{1}{\mathrm{Lip}(f)} \int_{\mathbb{R}^d} f(x) \, \mathrm{d}(\mu - \nu), \tag{3}$$

where $\mathrm{Lip}(f)$ represents the Lipschitz constant of function f. While this equivalent definition of the optimal transport distance is rarely used in graphics (with the notable exception of [Mullen et al. 2011]),

it is especially interesting in our semi-discrete setting: it allows us to derive a tight bound of the MC integration error for Lipschitz continuous functions. In particular, using a *uniform density* $dx/|\Omega|$ over the d-dimensional domain Ω, we deduce from Eq. (3) that for a sample set $X = \{x^1, ..., x^n\}$, one has for any Lipschitz function f:

$$\left| \int f(x)\, dx - \frac{1}{n} \sum_{i=1}^{n} f(x^i) \right| \leq \mathrm{Lip}(f)\, .\, W_1(X, 1_\Omega), \tag{4}$$

where 1_Ω denotes the constant density $1/|\Omega|$. This derived bound has a very similar form as the Koksma-Hlawka inequality from Eq. (1): it also offers an error bound written as the product of two separate factors, one only dependent on the function being integrated, and one purely based on a measure of spatial uniformity — which is no longer the discrepancy, but an optimal transport analog. Note that while the Koksma-Hlawka bound was assuming that the function had a finite Hardy-Krause variation, this one now assumes that the function is Lipschitz continuous instead. Thus, the OT-based bound is technically more restrictive in its assumption of the integrand. However, Lipschitz continuity is a reasonable assumption in many graphics applications; moreover, both functional spaces fail to include basic discontinuous functions, and practical functions may lead to actual integration errors below both theoretical bounds.

2.3 Computational optimal transport in high dimension

Equipped with the error bound from Eq. (4), one may be tempted to directly optimize a sample distribution by minimizing its Wasserstein distance to the uniform density. However, and despite a number of numerical optimal transport tools recently introduced in graphics [De Goes et al. 2012; Solomon et al. 2014, 2015; Nader and Guennebaud 2018], dealing with transport optimization in high dimensions is not computationally tractable directly. Fortunately, a recent variant of optimal transport offers a practical solution. A modified notion of optimal transort, now called *sliced optimal transport* (SOT [Pitié et al. 2005; Rabin et al. 2011; Bonneel et al. 2015; Bonneel and Coeurjolly 2019]), proposes to express the transport distance between two densities via an integral of 1D optimal transport distances between all 1D projections of these densities. Given that semi-discrete optimal transport in 1D amounts to a simple sort (for all p-Wasserstein distances W_p, $p \geq 1$), the SOT distance W_{SOT} is amenable to tractable computational evaluations and optimizations even in dimensions well above three through repeated 1D optimizations. However, to our knowledge, neither semi-discrete transport nor the optimal location problem have been previously investigated in the sliced optimal transport context. Moreover, the sliced optimal transport distance being bounded by the d-th power of the optimal transport distance W_1 [Bonnotte 2013], the bound from Eq. (4) with sliced optimal transport becomes:

$$\left| \int f(x)\, dx - \frac{1}{n} \sum_{i=1}^{n} f(x^i) \right| \leq C_d\ \mathrm{Lip}(f)\ W_{\mathrm{SOT}}(X, 1_\Omega)^{\frac{1}{d+1}}, \tag{5}$$

where C_d is a constant that depends on the dimension d. Using SOT instead of OT is thus akin to using a simplifications of the notion of discrepancy in QMC approaches: it only relaxes the error bound on integration errors. We are now ready to introduce the notion of sliced optimal transport for sampling by adapting previous work to our context of semi-discrete transport discussed above, and derive a novel algorithm for the generation of equidistributed sample points.

3 SLICED OPTIMAL TRANSPORT SAMPLING

This section details the main theoretical and practical aspects of our algorithm. We first review the general approach, before showing how to efficiently adapt it for the uniform sampling of a d-ball and a d-cube. We also extend our approach to provide projective SOT sampling for rendering purposes.

3.1 Sliced optimal transport formulation

The core idea of our approach is to cast the problem of sampling a distribution in term of the minimization of a sliced optimal transport distance [Rabin et al. 2011; Bonneel et al. 2015], for which repeated 1D transports can be used to spread samples efficiently in arbitrary dimensions. To this effect, we define a SOT distance (or cost) $W_{\mathrm{SOT}}(X, \varphi)$ between a density φ and a pointset X as:

$$W_{\mathrm{SOT}}(X, \varphi) = \int_{\mathbb{S}^{d-1}} \left(\min_{\pi} \int_{\mathbb{R}} \|x - x_\theta^{\pi(x)}\|\, R_\theta \varphi(x)\, \mathrm{d}x \right) \mathrm{d}\theta, \tag{6}$$

where $\mathrm{d}x$ denotes the 1D measure, θ denotes a direction (unit vector in $\mathbb{S}^{d-1} \subset \mathbb{R}^d$), $x_\theta^j := \mathbf{x}^j \cdot \theta$ is the 1D abscissa representing the orthogonal projection of sample point $\mathbf{x}^j$ from X onto the 1D linear subspace along θ, and $R_\theta \varphi$ is the scalar function defined as

$$R_\theta \varphi(s) := \int_{\theta \cdot \mathbf{x} = s} \varphi(\mathbf{x})\, \mathrm{d}x. \tag{7}$$

One recognizes $R_\theta \varphi(s)$ as the Radon transform [Radon 1986] of the density φ, integrating the density φ in the affine subspace orthogonal to a "slice" direction $\theta \in \mathbb{S}^{d-1}$ at abscissa s. Our sliced optimal transport distance can be understood as an integral over all directions θ of the 1D (classical) optimal transport cost W_1 between the projection of the pointset X along direction θ and the orthogonal projection of the spatial density φ along this same direction: essentially, we express the fit between a pointset and a density by the integral of the fit of their 1D projections, see Fig. 2. Instead, previous work considered W_2 for sliced distances [Rabin et al. 2011; Bonneel et al. 2015]; but the W_1 optimization, which provides a tighter bound, results in the same 1D assignments, although the optimal assignment is no longer unique.

As a consequence, this formulation has a number of advantages that we can leverage for computational purposes. First, reducing the distance one (or a few) direction(s) at a time is very efficient as we shall see next, in sharp contrast to the general optimal transport problem. Second, once a distribution is optimized with respect to a direction θ, it cannot have sample alignments along hyper-planes orthogonal to θ: this naturally prevents aliasing artifacts and offers the opportunity to prevent alignments in specific subspaces as well, as we will explore in Sec. 3.6. Third, our iterative optimization is agnostic to the domain dimensionality.

3.2 General approach to SOT sampling

Our sliced OT approach immediately suggests an algorithmic approach to optimizing a pointset: one can *project* both the target density and the pointset onto a series of one-dimensional lines, and successively optimize the location of all points along each of these directions to improve the one-dimensional fit between projected sample density and projected target density.

Discrete set of directions. In order to discretize Eq. (6), we first randomly sample K directions $\{\theta_i\}_{i=1..K}$ (typically we use $K = 64$). Generating each direction with a uniform probability over the

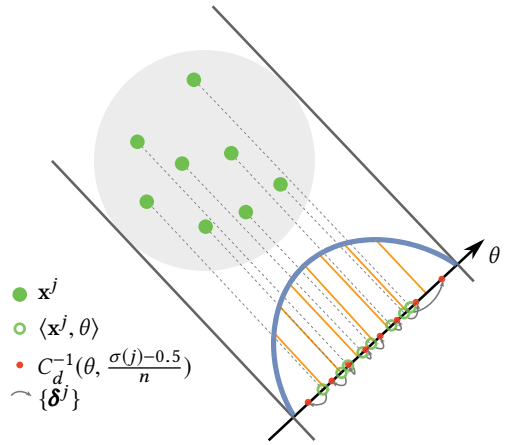


Fig. 2. Optimal transport via one-dimensional slices. Our approach adapts a distribution of n samples (green dots) to a spatial density function (here, uniform, in grey) in arbitrary dimension through iterative 1D optimization of the sliced optimal transport energy between the projected samples (green circles) onto an arbitrary direction θ and the density projected orthogonally to this direction. The blue curve indicates the Radon transform of the disk-shaped density,and the red dots indicate the directional update $\boldsymbol{\delta}^j$ of the samples to optimize their distribution along θ.

hypersphere $\mathbb{S}^{d-1}$ is trivially achieved by first generating d random variables following a normal distribution $\mathcal{N}(0,1)$, and normalizing these variables to obtain the components of a unit vector.

Radon transform. We then compute the Radon transform of φ along the directions $\{\theta_i\}_i$. The Radon transform defines a new probability density function $R_{\theta_i}\varphi$, of which we compute the cumulative density function C_{θ_i} defined as:

$$C_{\theta_i}(x) \coloneqq \int_{-\infty}^{x} R_{\theta_i}\varphi(s)\mathrm{d}s,$$

and its inverse $C_{\theta_i}^{-1}(x) \coloneqq \inf\{t \in \mathbb{R} : C_{\theta_i}(t) > x\}$.

Slice optimization. Finally, we relocate the sample point $\mathbf{x}^j$ to a new location $\mathbf{x}^j + \boldsymbol{\delta}^j$, where the total displacement $\boldsymbol{\delta}^j$ is

$$\boldsymbol{\delta}^j \coloneqq \frac{1}{K}\sum_{i=1}^{K} d_{i,j}\,\theta_i,$$

with $d_{i,j}$ being the displacement of $\mathbf{x}^j$ that minimize the 1D sliced optimal transport along direction θ_i. Denoting σ the permutation of the indices $\{j\}_{j=1..n}$ such that $\left(\mathbf{x}^{\sigma(j)} \cdot \theta_i\right)_j$, is a *sorted* sequence of increasing values, one computes $d_{i,j}$ directly via

$$d_{i,j} = C_{\theta_i}^{-1}\left(\frac{\sigma(j) - \frac{1}{2}}{n}\right).$$

Spreading samples according to the Radon transform of the density along the set of chosen directions then minimizes W_1 along these directions — thus improving the error bound from Eq. (5).

Discussion. The implementation of our approach is thus conceptually simple: after initializing the pointset with a scrambled Sobol sequence [Owen 1998] (which offers, at very low computational cost, a better initial distribution than a random distribution), the iterative process considering K slices at a time to displace the samples is repeated until convergence or when a fixed maximum number of steps is reached. Note that the cost of the algorithm involves K sorts of n projected points per iteration (i.e., $O(Kn\log n)$), as well as the cost for computing (or precomputing) the Radon transform in a number of directions. We show next that this last task can be achieved in constant time for a simple, yet very common case.

3.3 Uniform SOT sampling over d-balls

A key component of our formulation is the (partial) *Radon transform* $R_\theta\varphi$ of the density φ in a direction θ, and its inverse cumulative distribution function. Numerical tools based on the Fast Fourier Transform and the so-called Fourier slice theorem allow for the efficient numerical evaluation the Radon transform [Toft and Sørensen 1996] on discretized domain. However, such numerical approximations can become impractical when dealing with very high dimensions unless a closed-form expression of the Fourier transform is known. Fortunately, the particular, yet common case of a *uniform density on a unit d-ball* is particularly simple to evaluate without advanced numerical methods. Indeed, the radial symmetry of the d-ball implies that the Radon transform does not depend on the chosen direction θ. We will hence denote it as $R\varphi(s) \coloneqq R_\theta\varphi(s)\ \forall\theta$. Additionally, when φ is a uniform density, we can explicitly compute the resulting Radon transform in closed form: its cumulative density function becomes trivial to evaluate for odd-dimensional spaces, and requires only little additional effort for even-dimensional spaces. We prove in App. A that the expression for the function $R\varphi$ and its cumulative density function C_d in dimension d is:

$$R\varphi(s) = \frac{\pi^{d/2}}{\Gamma\left(\frac{d}{2}+1\right)}\sqrt{1-s^2}, \qquad \text{and} \tag{8}$$

$$C_d(x) = \begin{cases} \sum_{k=0}^{(d-1)/2} (-1)^k \binom{\frac{d-1}{2}}{k} \frac{x^{2k+1}}{2k+1} & \text{if } d \text{ is odd,} \\ \frac{\sqrt{\pi}\,\Gamma\left(\frac{1+d}{2}\right)}{2\,\Gamma\left(1+\frac{d}{2}\right)} + {}_2F_1\left(\frac{1}{2}, \frac{1-d}{2}, \frac{3}{2}, x^2\right) x & \text{if } d \text{ is even,} \end{cases} \tag{9}$$

where $\Gamma(.)$ is the Gamma function, and ${}_2F_1(.)$ is the so-called hypergeometric function, involving polynomials and trigonometric functions; see Fig. 3. We further invert the cumulative density function numerically using a gradient descent approach to offer an efficient way to evaluate the inverse cumulative density function $C_d^{-1}(x)$: this inverse function is tabulated once and for all as a precomputation.

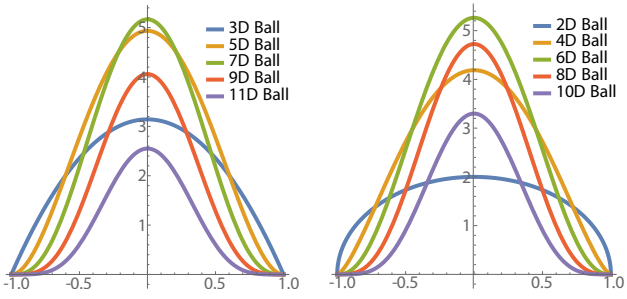


Fig. 3. Radon for d-balls. The Radon transform of a d-ball is direction-independent, so we display the radial component of the Radon transform of a ball as the dimension d increases. While the transform is polynomial for odd dimensions (left), it involves the hypergeometric ${}_2F_1$ function for even dimensions (right).

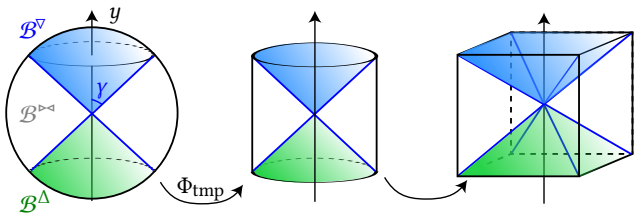


Fig. 4. Ball-to-cube map in arbitrary dimension. We illustrate the ball-to-cube mapping process. First, the ball is divided into two parts: the inside and outside of a double cone. These parts are then transformed to a cylinder via the transformation Φ_{tmp}. They are then mapped to the cube, resulting in a generalized Shirley-Chiu transform [1997] from ball to cube.

3.4 Uniform SOT sampling over d-cubes

While spherical domains are often desirable, cubical domains are also frequently required in a variety of sampling applications. The naive solution in this case, based on rejecting samples generated in a circumscribing ball if they do not lie inside the unit cube, is not algorithmically tractable as the volume ratio of the ball to the cube grows exponentially with the ambient dimension. Computing the (partial) Radon transform for a uniform density within a d-cube suffers from the same exponential dependence with the dimension, rendering this second option impractical as well.

Mapping approach. Instead, we propose a strategy that maps the d-dimensional ball $\mathcal{B}^d$ to the d-dimensional cube $[0,1]^d$ while preserving the uniformity of our distributions. Such a mapping should have a constant Jacobian determinant (thus preserving volume) so as not to affect the local density of points, and should be of minimal distortion. Note that this is precisely what Shirley and Chiu [1997] proposed in 2D (i.e., an area-preserving map from the disk to the square), but now extended to arbitrary dimension. We leverage the work of Griepentrog et al. [2008] to offer such a d-dimensional generalization, which will allow us to directly transform our efficient generation of samples on the uniform d-ball onto the uniform d-cube (Fig. 6 shows pointsets from the disk mapped to the square).

A d-ball to d-cube map. Griepentrog et al. [2008] proposed a volume-preserving invertible mapping Φ from the d-ball to the d-cube using the $(d-1)$-cylinder as an intermediate domain. Since this work does not seem known in our community, we provide a brief overview of the general construction here, and write down the expressions of the various maps involved in App. B for the reader's convenience. First, the ball is decomposed into three parts: $\mathcal{B}^d = \mathcal{B}^\nabla \cup \mathcal{B}^\Delta \cup \mathcal{B}^{\bowtie}$, where $\mathcal{B}^\nabla$ and $\mathcal{B}^\Delta$ form a double cone aligned with the last axis of $\mathbb{R}^d$, while $\mathcal{B}^{\bowtie}$ is the leftover part as depicted in Fig. 4. More specifically, the three subdomains are defined as:

$$\begin{aligned}
\mathcal{B}^\nabla &:= \left\{(\mathbf{x}, y),\, \mathbf{x} \in \mathbb{R}^{d-1}, y \in \mathbb{R}^+ \mid y \geq \gamma \|\mathbf{x}\|\right\} \cap \mathcal{B}^d \\
\mathcal{B}^\Delta &:= \left\{(\mathbf{x}, -y),\, \mathbf{x} \in \mathbb{R}^{d-1}, y \in \mathbb{R}^+ \mid y \geq \gamma \|\mathbf{x}\|\right\} \cap \mathcal{B}^d \\
\mathcal{B}^{\bowtie} &:= \mathcal{B}^d \setminus \{\mathcal{B}^\nabla \cup \mathcal{B}^\Delta\},
\end{aligned}$$

where γ, defining the aperture angle $\frac{\pi}{2} - \arctan(\gamma)$ of the cone, is the unique solution in $\mathbb{R}^+$ of:

$$(d-1)\int_0^{\arctan(1/\gamma)} \sin(\alpha)^{d-2}\,\mathrm{d}\alpha = \int_0^{\arctan(\gamma)} \cos(\alpha)^{d-2}\,\mathrm{d}\alpha\,. \quad (10)$$

Table 1. Ball-to-cube map parameters. Values of γ, ϱ, and τ to compute the invertible map of Griepentrog et al. [2008] for dimensions $d \leq 10$.

dim.	γ	ϱ	τ
3	$2/\sqrt{5}$	$2/3$	$\sqrt{2}/\sqrt{3}$
4	0.821353089207943	0.5890486225480863	0.8382695966098716
5	0.7666031370294717	0.5333333333333332	0.8545740127924683
6	0.723424902134195	0.4908738521234051	0.8673491949880967
7	0.6881297272460576	0.4571428571428572	0.8776916965664375
8	0.6585046305043636	0.4295146206079796	0.8862745508336505
9	0.6331279880529004	0.4063492063492063	0.8935367660649970
10	0.611037644218746	0.3865631585471816	0.89977849007590771

The ball-to-cylinder mapping Φ_{tmp} has a closed form expression on both $\mathcal{B}^\nabla$ (and thus, on $\mathcal{B}^\Delta$ by symmetry) and $\mathcal{B}^{\bowtie}$ (see Appendix). This temporary map allows us to recursively build the ball-to-cube mapping by always separating the last coordinate of a point in dimension d via the relationship:

$$\text{Ball2Cube}_d\begin{pmatrix}\mathbf{x}\\ y\end{pmatrix} = \begin{pmatrix}\text{Ball2Cube}_{d\text{-}1}(\mathbf{x}')\\ y'\end{pmatrix} \text{ where } \begin{pmatrix}\mathbf{x}'\\ y'\end{pmatrix} = \Phi_{\text{tmp}}\begin{pmatrix}\mathbf{x}\\ y\end{pmatrix}, \quad (11)$$

where we use a row vector notation for the arguments and a bold font for multivariable arguments. Only the value of γ for $d = 3$ was analytically computed in the original paper by Griepentrog et al. [2008], but computing the solution of Eq. (10) (and the corresponding ϱ and τ values, see Appendix) in higher dimensions is a simple matter of precomputation. We provide these values up to dimension 10 in Tab. 1 for completeness.

3.5 Multiple class SOT Sampling

The simplicity and generality of our SOT sampling allows a number of useful variants. We can, for instance, achieve multiclass sampling by simply altering our optimization procedure. From an initial distribution of samples already labeled, we can make sure these labels are perfectly interwoven during each 1D sort by reordering the assignment of the samples based on their associated label: this alteration will induce a better distribution of the respective classes with respect to each other in higher dimension. However, it may induce a decrease in global uniformity of the whole sample set due to the enforcement of this interweaving constraint. We thus *alternate*

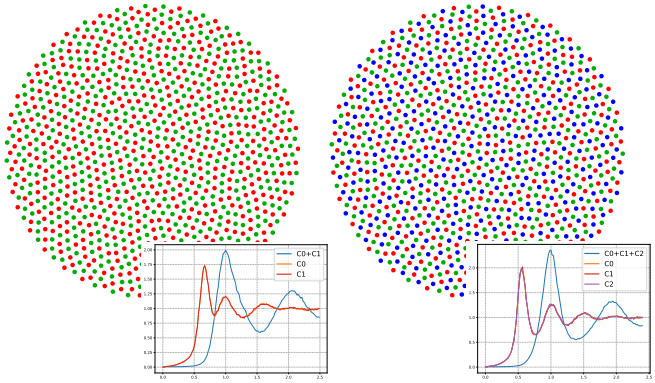


Fig. 5. Multiclass SOT sampling. SOT sampling with two classes (left) and three classes (right) both exhibit excellent spectral behavior for individual classes (C_i), as well as for the entire set; note that the radial mean power spectra were computed using only the samples within the square inscribed in the disk to allow for a proper discrete Fourier transform.

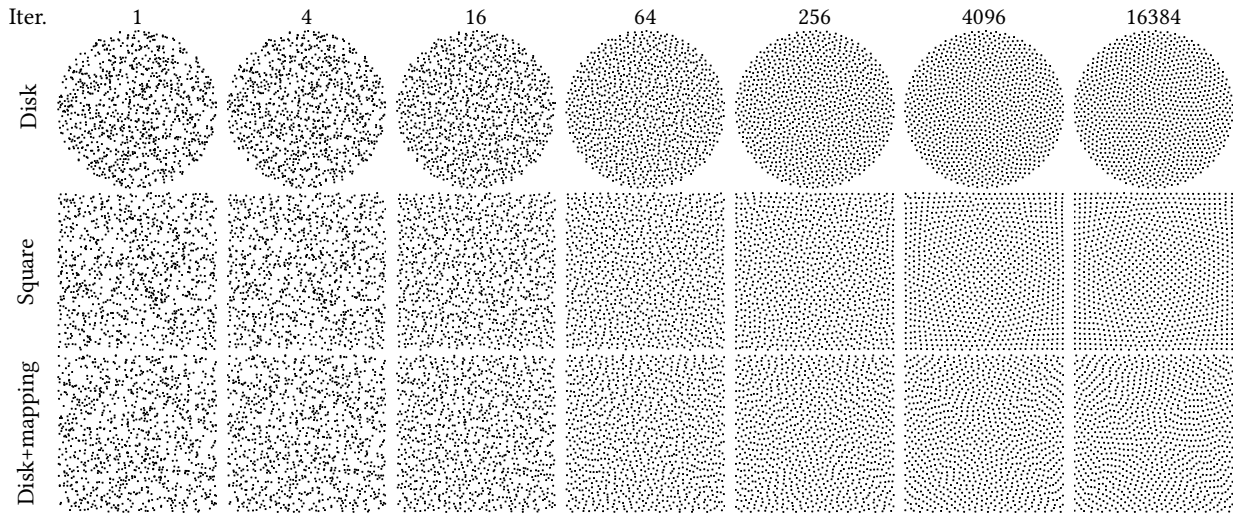


Fig. 6. From random to SOT samples for uniform density. Starting from a random distribution of 1024 samples over a disk (top left) or a square (middle left), iterating batches of 64-slice optimizations (left to right, up 16384 iterations) improves the uniformity of the samples, with no further visual improvement after 300 iterations. If we map our disk distribution (top) onto the square (bottom) via our area-preserving deformation, we observe a very good approximation of the square-based distribution which no longer requires the evaluation of a Radon transform over the square.

this modified 1D treatment with the vanilla one: alternating multi-class 1D steps with regular ones until convergence or a maximum iteration count results in *both* high class distribution quality and global uniformity as seen in Figure 5.

3.6 Projective SOT sampling for rendering

Another variant of our SOT sampling targets rendering specifically. As we discussed early on, Monte Carlo integration is particularly relevant to light transport simulation: rendering requires the estimation of an integral over all light paths within a scene connecting the camera sensor to light sources [Veach 1997]. However, the integrand in this case has a particular structure that our SOT sampling approach can further exploit. To simulate light bouncing k times over surface elements of a 3D scene, radiance must be integrated over a $2k$-dimensional space. Furthermore, the radiance reaching the camera can be integrated over its aperture to simulate depth-of-field effects, over its sensor to avoid anti-aliasing, and over time to handle motion blur. While this integration requires well distributed samples over the entire high-dimensional space, it can be beneficial to distribute samples uniformly within each of these k subspaces as well (i.e., for each possible path length): both Reinert et al. [2016] and Perrier et al. [2018] have proposed techniques to enforce a blue noise or Poisson disk property not only in a high dimensional path space d, but also in each 2-dimensional pair or combinations of dimensions.

Our method easily supports a similar functionality by performing optimization steps using projection directions lying within these subspaces. For a given choice of path-space dimension $2k$ (typically, we pick 6D or 8D), we initialize n samples in this high dimension through random sampling of a uniform density. Then, for each step of our minimization, we pick a set of directions from the full space *as well as* from the target projections in orthogonal two-dimensional subspaces: we randomly pick K directions uniformly distributed in the full high dimensional space, and $K/2$ directions uniformly distributed within *each* of the k projective subspaces. Since a good projective uniformity on the d-ball does not equate a good projective uniformity *after* mapping onto the d-cube, we add a modification of our computation of displacements: for each projective direction, we project the current samples onto its associated 2D subspace, map them to the corresponding 2-ball, and perform the SOT estimate of a displacement vector per sample there; we then map all the displacements back onto the d-cube using the inverse map. When all these displacements for all subspaces and the entire space are computed, we average them to deduce a total displacement per sample as in the vanilla SOT optimization procedure. This averaging can even be made dependent on the subspaces: one can for instance weight the displacements proportionally to the dimensionality (or respectively, the inverse thereof) of their associated subspace to induce a stronger preference for equidistribution in high (respectively, low) dimensional spaces. This modified optimization procedure, iterated until convergence or a maximum iteration count, allows for a balance between full-space equidistributions and equidistributions in the chosen subspaces and their combinations. Other strategies with a different balance between various properties could be derived as well, but we found this simple variant (that we call "projective" sliced optimal sampling) quite efficient as is. We will demonstrate its efficacy in Sec. 4.4 for rendering purposes.

4 RESULTS

We now present a series of numerical tests to ascertain the properties of our new SOT sampling method. While performing an exhaustive analysis of Monte Carlo integration with SOT samples in high dimensions is unsurmountably difficult, we study its behavior from a variety of perspectives, including visual inspection and spectral analysis in low dimensions, numerical accuracy and convergence for both low and high dimensions, as well as computational efficiency.

4.1 Low-dimensional SOT sampling

We first provide visual and numerical evaluations of our results in low dimensions to better understand their basic properties.

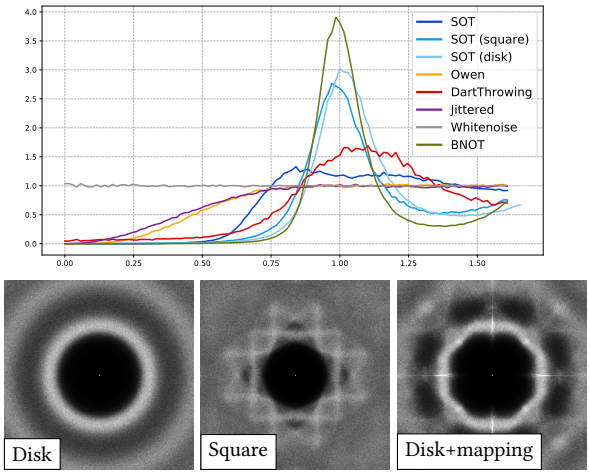


Fig. 7. 2D Fourier spectrum. Whether a SOT sampling with 4K samples is generated on a disk, directly on a square, or via mapping from a SOT sampling of the disk to the square, its power spectrum and radial mean power spectrum clearly exhibit all the hallmarks of blue noise distributions — with additional artifacts due to boundary alignment and/or mapping effects on a square domain. Note that the (radial mean) power spectrum for the disk case was computed using only samples within the square inscribed in the disk to allow for a proper discrete Fourier transform.

SOT iterations in 2D. Fig. 6 shows how a 2D pointset, initialized through random sampling of a uniform distribution, evolves as SOT iterations of 64-slice displacements are performed, for both a disk (top) and a square (middle) domain. The disk domain uses our uniform sampling method over balls as described in Sec. 3.3, while the square domain uses a more general – thus more costly – numerical evaluation of a (partial) Radon transform over the domain. After around 256 iterations, the pointsets have visually converged.

Mapping effects. Still in Fig. 6 (bottom), we also display a SOT sampling in a square domain computed from the disk sampling result via our disk-to-square map as explained in Sec. 3.4. While the resulting pointset based on the map is not as well distributed as the properly evaluated sliced optimal transport sampling, the former does not result in obvious directional alignments, and scales far better to high dimensions in terms of computational complexity.

Spectral properties in 2D. As a means to evaluate SOT sampling compared to previous blue noise sampling methods, we also provide in Fig. 7 the power spectrum, computed via averaging over angles, of a sliced optimal transport pointset in a 2D square generated via mapping. The resulting spectrum is, as expected, close to the characteristic blue noise profile. In particular, we observe the telltale behavior near the DC component where the spectrum remains flat for a large frequency band, and a white noise at high frequencies. Compared to other blue noise methods such as [De Goes et al. 2012], we also exhibit a peak at the characteristic frequency, but observe a minor anisotropy of the spectrum due to the distortion induced by the disk-to-square mapping.

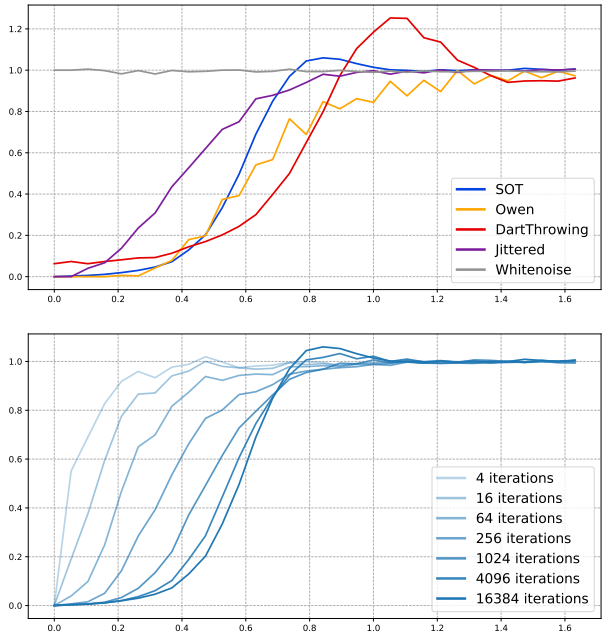


Fig. 8. 4D radial spectra. (top) Radial power spectra comparison in dimension 4 of 4K samples for various samplers; note that while dart throwing exhibits a stronger peak than SOT, its behavior at low frequencies is also significantly worse than all others (bottom) With a random initialization, the radial power spectra of SOT sampling goes from a white-noise shape to a more blue-noise one during optimization.

Spectral properties in 4D. As a first foray into higher dimensions, we show in Fig. 8 how our SOT optimization in 4D improves the spectrum of a sample distribution (computed via a 4D fast Fourier transform of the unit 4-cube) as K-slice batches are performed, still using a random initialization of the samples for illustration purposes. The profile goes from flat at very early stages, to blue noise. Finally, we demonstrate that other usual sampling techniques (dart throwing, Sobol, or jittered grids) do not exhibit a similar profile in 4D (top).

4.2 Integration accuracy for SOT sampling

The main goal of our work is to provide an efficient way to create high-dimensional pointsets for which constant-weight quadrature evaluations offer improved accuracy compared to existing methods. We thus performed a number of numerical evaluations, using simple Gaussian functions and other analytical integrands, to evaluate how our approach fares compared to a regular Monte Carlo method or to quasi-Monte Carlo methods.

Smooth functions in low & high dimensions. In Fig. 10 (left), we test the accuracy of Monte Carlo integration for a multivariate Gaussian distribution of the form $g(x) = \exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right)$, where we constrain the mean vector to satisfy $0 < \mu_i < 1$ and the covariance matrix so that each eigenvalue is at least 0.06 and at most 0.15. For each curve, we average the integration errors over the same 1024 randomly-selected Gaussians to offer a less noisy visualization, but depict the minimum and maximum errors from these 1024 integrations to offer an evaluation of the error variance. A typical multivariate Gaussian distribution used in our integration tests is

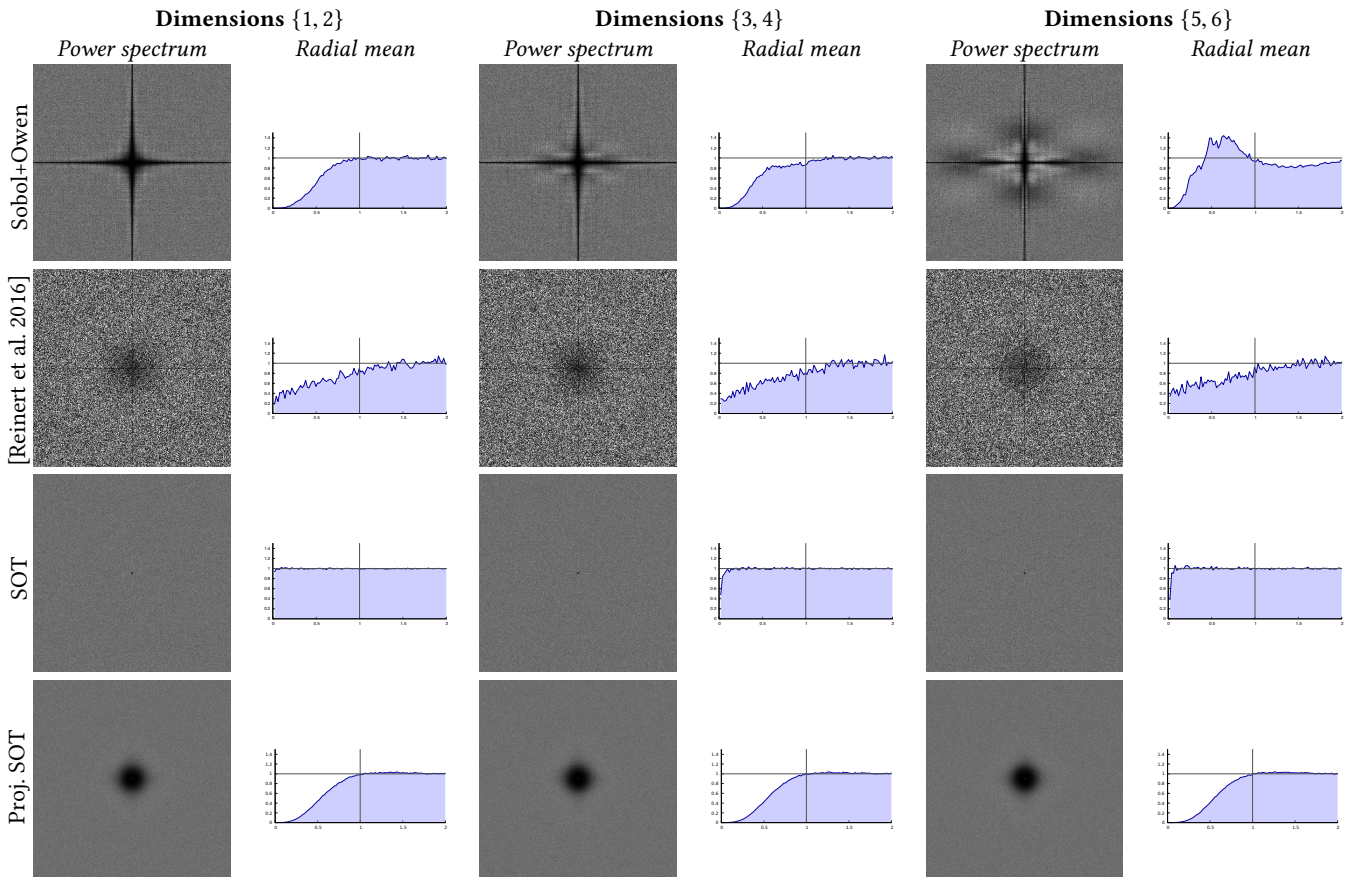
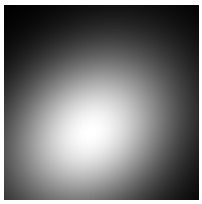


Fig. 9. Fourier spectra on projections. In dimension 6 with 8K samples, we compare our projective SOT sampling (bottom) to other techniques such as Projective Blue Noise [Reinert et al. 2016] to enforce good projective distributions in dimensions $\{1, 2\}$, $\{3, 4\}$ and $\{5, 6\}$. The power spectra and radial mean for these 2D projections demonstrate the quality of our projective SOT variant, and highlights that Owen-scramble Sobol sampling exhibits significant artifacts in these subspaces, while our vanilla SOT shows an expected white noise distribution.



shown in the inset. Monte Carlo integration using our SOT sampling performs far better than integration using Sobol or Sobol with Owen scrambling for 100 samples and above in this 2D test. We even consistently outperform the recent PMJ02 method of Christensen et al. [2018b]. This superior behavior holds true in 4D and 6D with a reduced margin of improvement. The same test in dimension 20 shows that we still outperform Sobol with Owen scrambling. All Sobol initialization vectors were taken from [Joe and Kuo 2008], in the same order. For completeness, we also included in all our graphs an example of optimized low-discrepancy pointsets: we used rank-1 lattices [Keller 2004] generated via the implementation of [L'Ecuyer and Munger 2016]. As a side note, rank-1 samples may happen to have toroidal symmetry over the unit domain due to their lattice nature, and this special configuration is known to lead to exponential convergence for the integration of smooth and periodic functions; our choice of functions purposely prevents this case, which would have otherwise been not representative of the typical use of numerical integration in higher dimensions.

Discontinuous functions in low & high dimensions. In order to explore the numerical behavior of our sampling strategy more broadly, we also tested the accuracy of MC integration for discontinuous functions. As a representative example, we chose a simple Heaviside ($f_n(p) = 1$ if $p \cdot n > 0$, 0 otherwise) function to show our robustness to discontinuities. The same error plot as above (with this time, 1024 different Heaviside with each a randomly-selected normal n) do not exhibit large differences between our SOT sampling and traditional QMC methods: as Fig. 10(right) indicates, this class of functions is not really well handled by either, consistent with the expected absence of theoretical error bounds in this case. Yet, SOT sampling performs as well as, or at times slightly better than, Sobol with Owen scrambling — even in dimension 20.

4.3 Computational complexity of SOT sampling

Our algorithm for SOT sampling generation requires the prior selection of two parameters: the number of slices K and the total number of batches. In order to evaluate a good strategy to automatically set these parameters, we plotted the SOT energy of a pointset as a function of batch and slice numbers. For all dimensions up to 20, it indicates that there is virtually no numerical advantage in using

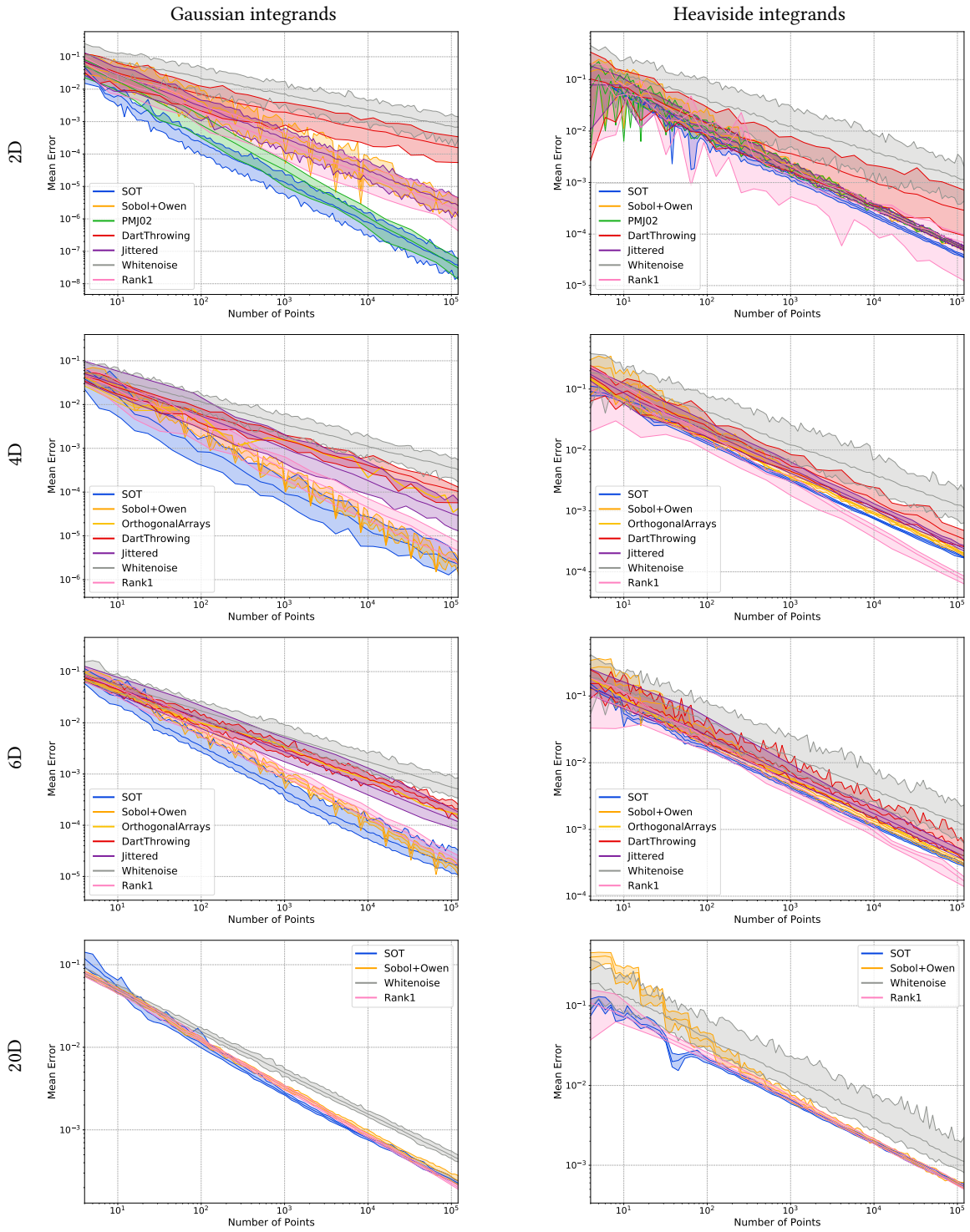
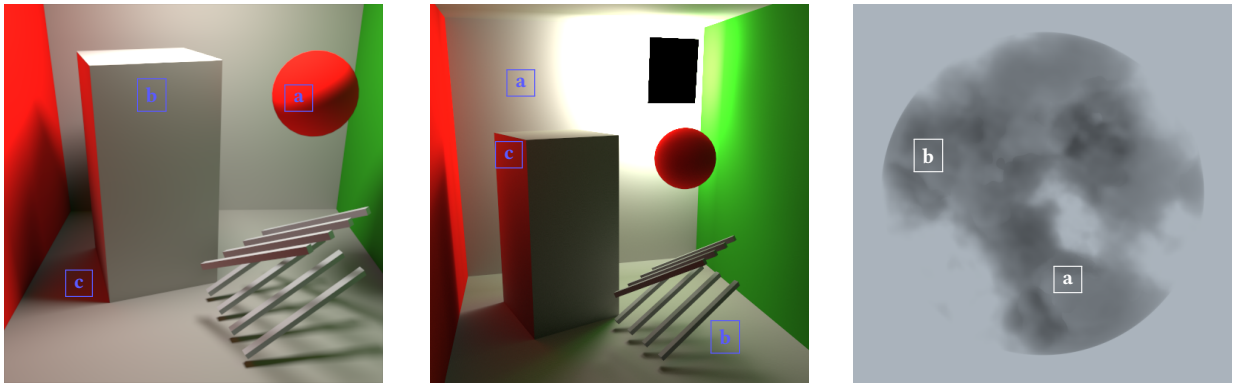
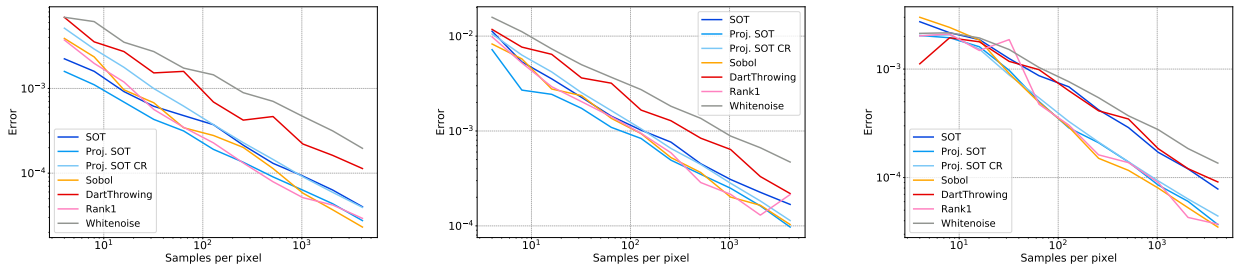


Fig. 10. Monte Carlo integration on canonical functions. We consider various integration tests in dimensions 2, 4, 6, and 20 to evaluate MC integration error as a function of the number of samples n. The Gaussian integrands (left) column depicts the error averaged over 1024 integrations of random multivariate Gaussian distributions in the $[0, 1)^d$ domain; each solid curve indicates this error for a different sampler, and the associated shaded region indicates the max and min error over the 1024 integral evaluations. The Heaviside integrands (right) column depicts the error averaged over 1024 integrations of random Heaviside functions going through the center of the $[0, 1)^d$ domain; the same visualization of min, mean, and max errors is used. Note that Orthogonal Arrays refers to the CMJND sampler of [Jarosz et al. 2019], while PMJ02 refers to the method of [Christensen et al. 2018b].



(I) Cornell scene, one bounce (II) Cornell scene, two bounces (III) Cloud rendering, two bounces

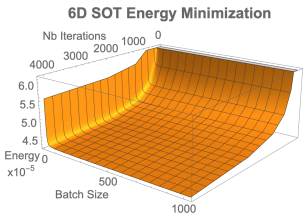
Fig. 11. Reference scenes used in our rendering experiments. For the Cornell test, we use either (I) indirect lighting with only one bounce (thus requiring 6D samples), or (II) two bounces (needing 8D samples); highlighted windows (marked as a, b, and c in each case) are 7×7-pixel regions used for convergence plots in Figs. 12 and 15 respectively. For our cloud scene with a participating medium (III), we render two light bounces in PBRT, using 8D samples (2D for rays initialization in screen-space, and 3D per bounce to sample depth and direction of each light–medium interaction); highlighted windows (marked as a, and b) are 16×16-pixels areas used for convergence plots in Fig. 17.



(a) (b) (c)

Fig. 12. Convergence plots for 6D. For the three pixel windows depicted in Fig. 11(a), we plot the graphs of absolute errors as a function of sample count.

more than 64 slices (see inset for the case of dimension 6; all other dimensions are similar up to scale). We therefore use $K = 64$ in all our results. The number of batches needed to reach a near optimal SOT energy is also quite independent of the dimension; we thus used 4096 batches for all dimensions up to 20. Higher dimensions may require, instead, a stopping criterion based on the magnitude of energy decrease for improved accuracy. With these two parameters fixed, we provide an evaluation of the typical performance of our algorithm in Fig. 18: we depict our running times as a function of the number n of samples and as a function of the spatial dimension d. We observe clearly an $O(d)$ complexity, with odd dimensions being slightly faster to handle than even dimensions because of hypergeometric function evaluations. We also see a quasilinear dependence to the number of samples as expected from our repeated 1D sorts. The time required to compute the sphere-to-cube mapping is negligible compared to the optimization steps, as it only involves a maximum of 0.01% of the total computational time. Finally, a typical running time for 16K samples in 8D using 4096 optimization batches of 64 slices each is below 10 seconds on an AMD Ryzen Threadripper 2990WX with 32 cores at 3 GHz.



4.4 Rendering with SOT sampling

Integrations performed in the context of rendering are very specific, so we proposed a variant of our sliced optimal transport approach by designing samples equidistributed in high dimensions as well as in select subspaces. A first way to evaluate the quality of the pointsets we generate for rendering purposes is to examine 2D projections of the samples in order to evaluate their equidistribution in the most relevant subspaces. Fig. 9 shows the spectra and radial spectra of each 2D projection for a 6D projective SOT sampling, generated with a weighting of respectively 2 and 1 for the full dimensional 6D space and each of the 2D projective subspaces. Compared to [Reinert et al. 2016], we observe a much improved projective behavior, with a clear absence of frequencies near DC. Note that a similar illustration in 4D is given in our supplemental material (as well as additional visualizations of 2D projections of 6D samples), and the improvements are equally clear.

We also tested the efficacy of this projective variant on three rendering scenes: the purposely simple Cornell test (Figs. 11 (I) and (II) show the two viewpoints we will use, in which small 7×7-pixel windows are marked as test regions of increasing complexity for error convergence plots), the more involved San Miguel scene (Fig. 1), and a volumetric rendering of a cloud (Fig. 11(III)). For the Cornell scene, Fig. 13 shows the spatial distribution (per-pixel

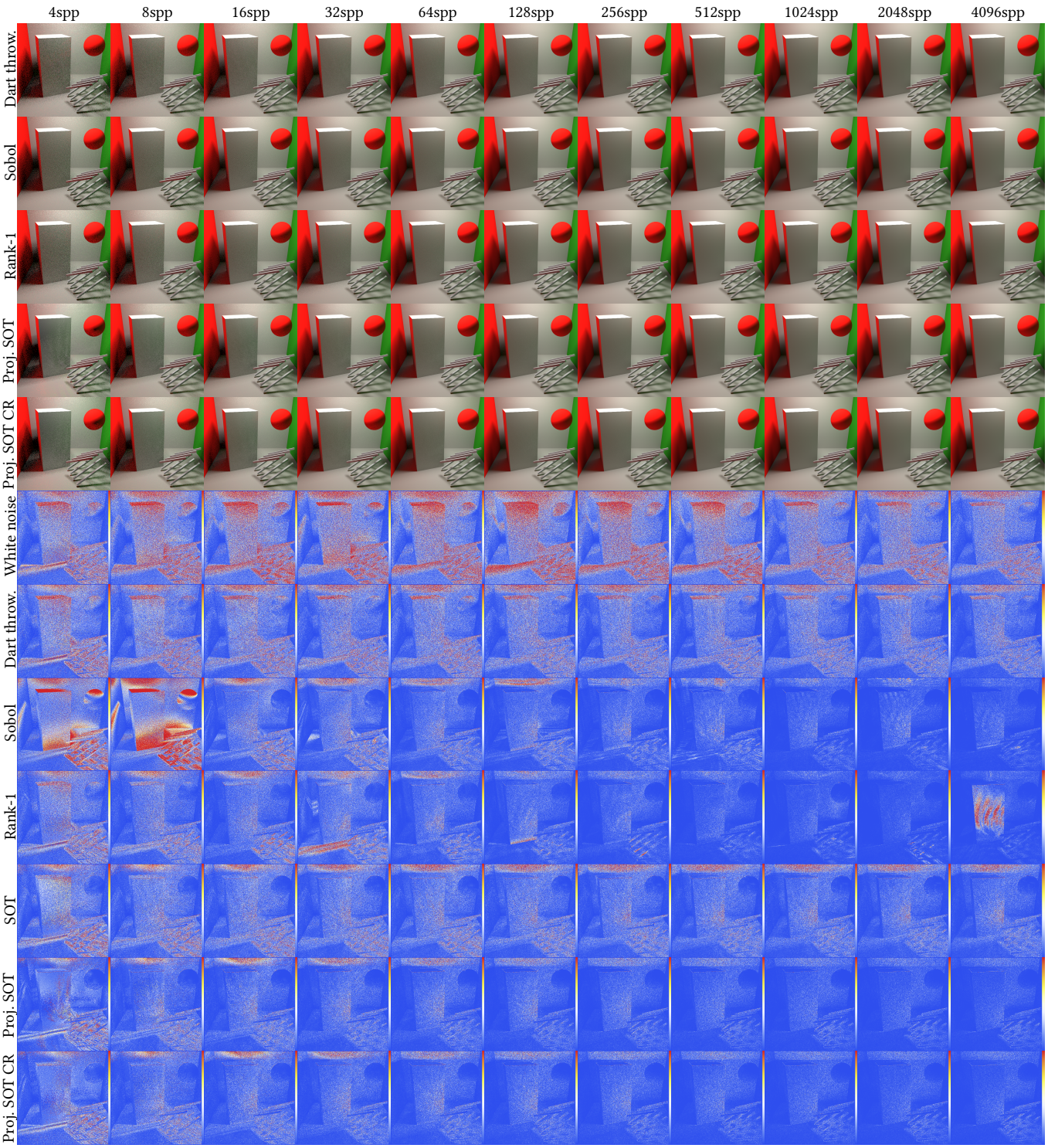


Fig. 13. Rendering results in dimension 6. We render the `Cornell` scene with one bounce of indirect lighthing using various samplers and for several sample counts. To highlight the efficiency of Monte Carlo integrations, we display the error maps (per-pixel absolute difference with reference image in Fig. 11) using pseudocolors (from blue to red, with a linear map from 0 to 0.002/spp). While SOT sampling does not help much for rendering, its projective variant is better and/or more consistent than other typical approaches, particularly so for low counts of samples per pixel.

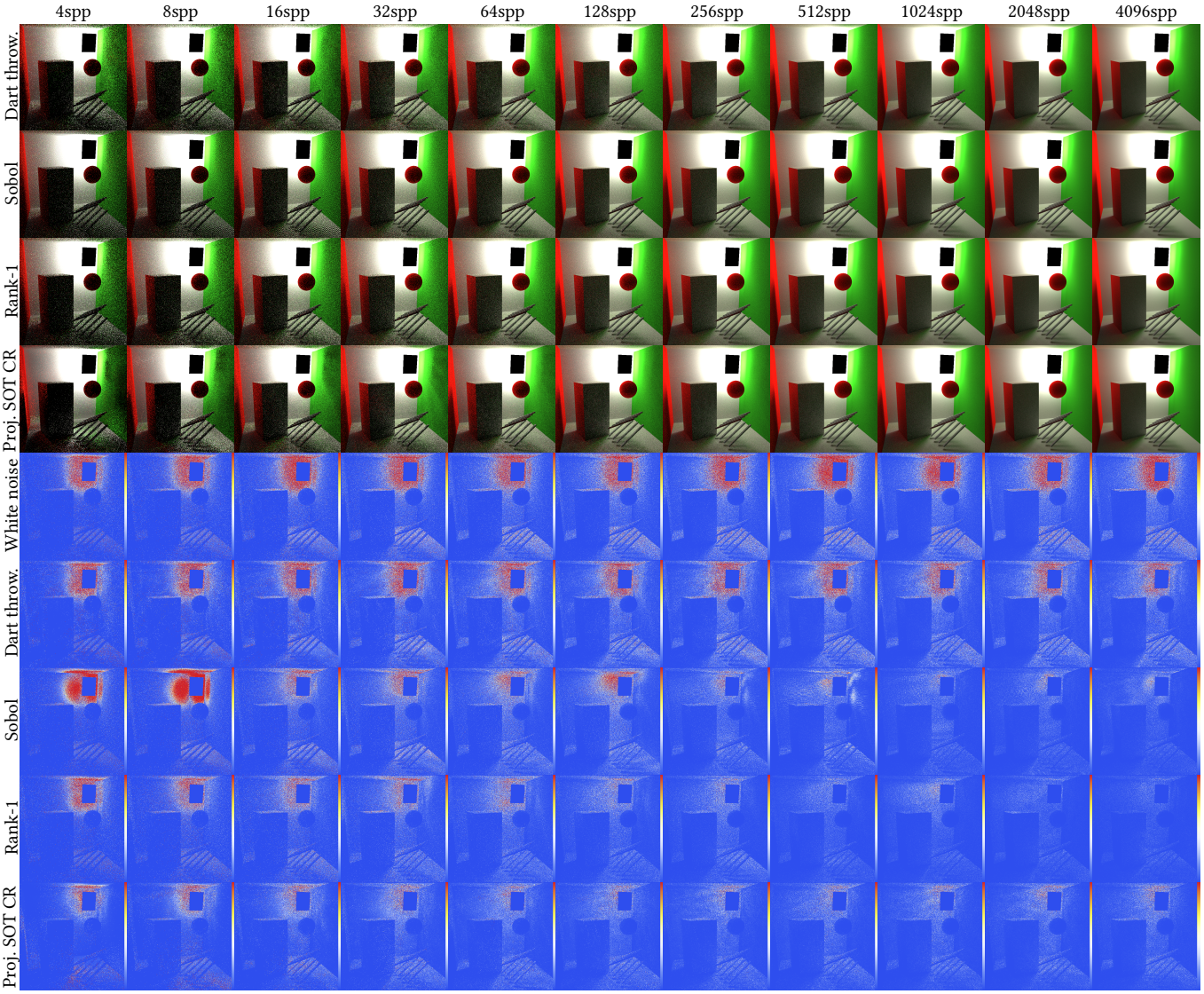


Fig. 14. Rendering results in dimension 8. We render the `Cornell` scene for two bounces of indirect lighthing using various samplers and for several sample counts. The same visualization as in Fig.13 is used, but now compared to the reference image in Fig. 11(b) and with a linear map from 0 to 0.001/spp.

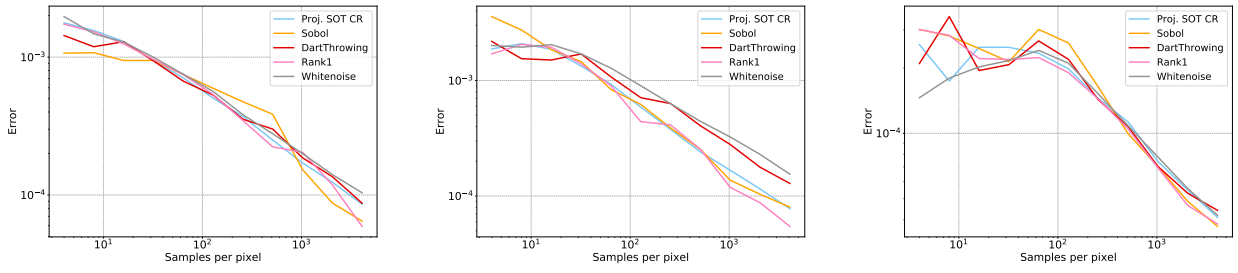


Fig. 15. Convergence plots for 8D. For the three pixel windows depicted in Fig. 11(right), we plot the graphs of absolute errors as a function of sample count.

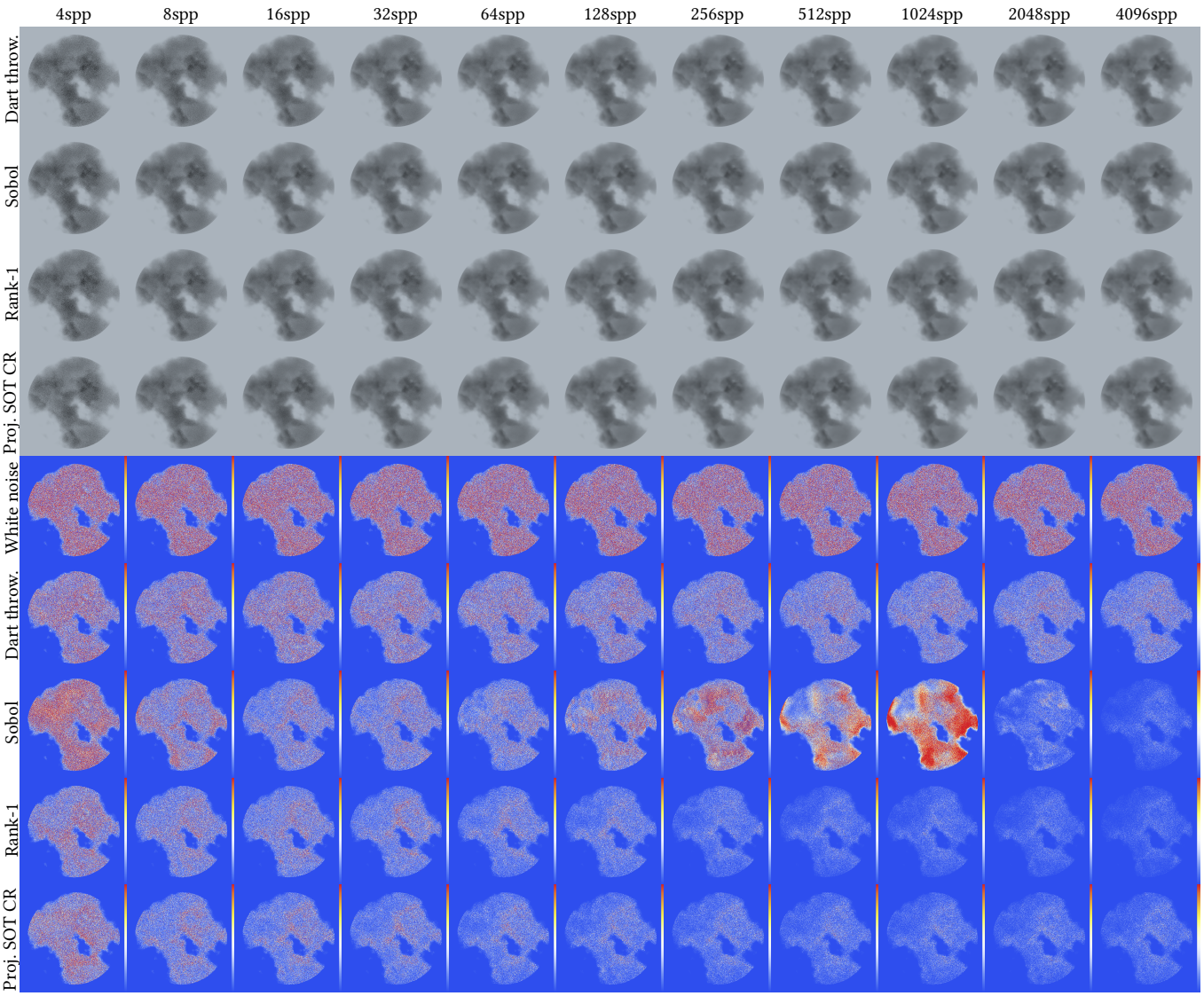
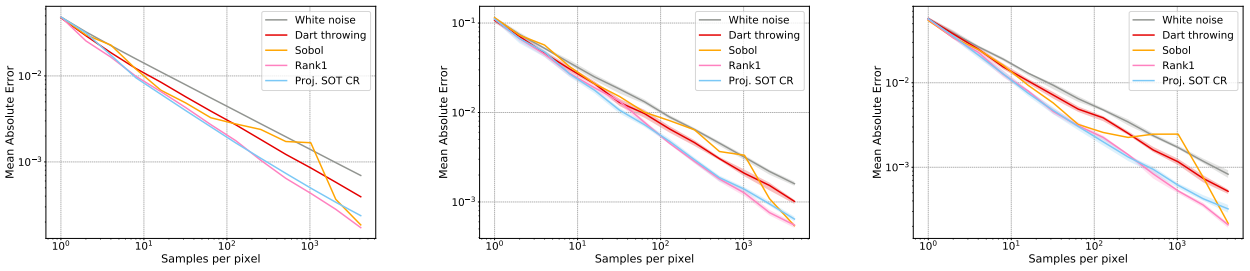


Fig. 16. Volumetric rendering results in 8 dimensions. We render a cloud-like participating medium, simulating two medium interactions (bounces) using various samplers and for several sample counts. Samples are 8-dimensional: 2D screen space and two bounces using 2+1D for each bounce direction and depth. As in Fig.14, error maps (per-pixel absolute difference with reference image in Fig. 11) use pseudocolors from blue to red, with a linear map from 0 to 0.002/spp.



Full image (a) (b)

Fig. 17. Convergence plots for 8D volumetric. From left to right: convergence of the full image, and of the 16×16-pixels areas highlighted in Fig. 11. The main curves represent the mean absolute error of 8 independent realizations (when applicable) while the encasing auras represent the bounds of the mean absolute error over the 8 independent realizations.

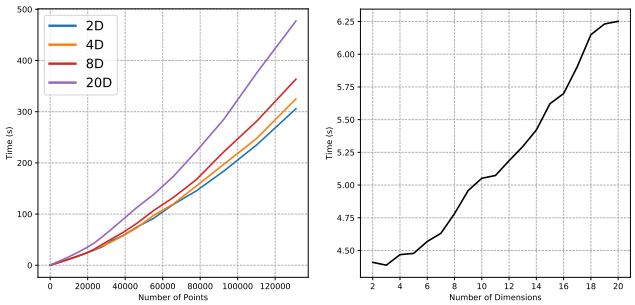


Fig. 18. Performance. To assess the efficiency of our algorithm, we plot its computational cost as a function of the number of samples (left, for dimensions 2 and 8), and as a function of the dimension (for 16K samples). We used 4096 batches of 64 slices for both graphs.

absolute difference with the reference image) of various samplers when indirect lighting with only one bounce (i,e., using samples in 6D generated as described above) is simulated, whereas Fig. 14 simulates two bounces (i.e., using samples in 8D, still with the same weighting strategy, but there is now one additional 2D subspace). For our volumetric rendering shown in Fig. 16, we also use 8D samples to handle two bounces: our projective SOT pointsets are optimized to sample both the whole 8D space as well as the 3D spaces of bounces, the 2D spaces of directions, and the 1D spaces of depths; the subspaces $(1,2,3,4,5,6,7,8)$, $(1,2)$, $(3,4,5)$, $(6,7,8)$, (3), $(4,5)$, (6), and $(7,8)$ are weighted respectively 4, 1, 2, 2, 1, 1, 1, and 1 during our projective optimization procedure. The participating medium we use has an isotropic phase function and its density field derives from a closed-form representation [Bálint and Mantiuk 2019] encased inside a spherical domain and modified to allow gaps of zero density. Such a representation allows for a closed-form expression of the transmittance over the interval of a ray–medium intersection, which in turn lets us apply closed-form tracking [Novák et al. 2018] to sample the depth of the interaction over the same finite interval. We provide results for sample counts varying from 4 to 4096. We tested our projective SOT sampling in two different ways. We first tried a straightforward strategy where each pixel randomly selects a realization of projective SOT sampling; we denote this approach as "Proj. SOT". We also tested adding a micro-Cranley-Patterson rotation per pixel, i.e., toroidal translations using random vectors of length $(2\cdot\text{spp})^{-1}$ (larger translations may alter the quality of non-toroidal samplers like ours [Singh et al. 2019]), to enrich the projective SOT realizations; this second aproach is denoted "Proj. SOT CR" in our results. The results show reduced errors for our sampling, at times significantly, compared to the traditional use of Sobol with Owen scrambling. We also rival, and sometimes outperforms, rank-1 sampling — although rarely on large sample counts. Note that our tests used one global Sobol pointset for the entire image (leveraging its stratification property), while the other samplers were used locally (per pixel), with toroidal translations for rank-1 and Proj. SOT CR.

Finally, we also provide convergence plots on three different 7×7 test windows, from both the one- and two-bounce cases of the Cornell scene in Figs. 12 and 15 respectively, and on two different 16×16 test windows from the volumetric scene in Fig. 17. These tests demonstrate that our projective SOT approach rivals optimized pointsets such as rank-1 in quality and can lead to noticeable gains compared to low-discrepancy sequences for less that 1000 samples, with the difference basically vanishing for larger sizes. While complex regions do not benefit much from our SOT sampling, less complex regions are often estimated with a lower error. Note that our supplementary material provides further visual results for completeness. We also observe uneven errors for Sobol sampling in all rendering cases: these errors are highly structured with blocky appearances that can be linked to uneven stratification in several dimension pairs for some sample numbers, see Fig. 12 in our Supplemental material. Rank-1 sampling is also uneven in quality, although less so: Fig. 13 shows that 4096spp behaves noticeably worse than 2048spp, while Fig. 14 does not show such an artifact — and in fact, outperforms projective SOT at high spp for this particular case. Our supplementary material provides visual evidence of spurious alignments in several dimension pairs as well in Figs. 19, 22 and 23.

5 CONCLUSIONS

Our proposed sliced optimal transport strategy for point sampling generation efficiently constructs, in low or high dimension, pointsets with good uniformity properties with respect to either a constant or variable density probability function. When using these samples for Monte Carlo integration, we observe reduced integration error compared to quasi-Monte Carlo approaches. Surprisingly, our approach is an order of magnitude better on smooth integrands than most low-discrepancy sequence strategies, known as the fastest variance reduction approaches for QMC.

Limitations. As we designed our sampling approach to reduce integration error, the support of the density distribution as well as the boundary of the domain of integration Ω affects our sample distributions: in a square for instance (Fig. 6, second row), we see that a layer of points has formed along the border. While this benefits the integration accuracy by improving the sampling of the "discontinuity" that the border creates, it may be detrimental if our samples are used for other purposes such as stippling. At this moment, we do not have a clear understanding of how these boundary samples affect practical applications and spectral properties.

Future work. This work raises a number of interesting questions, both on the practical front and from a theoretical standpoint. Convergence of our SOT optimization may become slow in very high dimensions (see Supplemental Material for statistics); if scalability is important, it might be interesting to explore how a faster convergence of our minimization could be obtained through more advanced numerical techniques than simple iterated averages of sliced transport optimizations. The design of other ball-to-cube maps minimizing different forms of distortion would also be of value. Additionally, while our projective sampling approach was already proven valuable for rendering applications, we believe that a more systematic evaluation of weighting strategies in our projective variant (maybe guided by the weights used in [L'Ecuyer and Munger 2016] or [Reinert et al. 2016]) could result in appreciable gains. Moreover, there may be a number of other contexts for which an extension of our projective variant could be valuable. In particular, more complex constructions of generalized projective SOT

sampling which could offer Pareto optimality of the equidistribution in each subspace are an intriguing possibility to maximize sampling quality. Finally, understanding from a theoretical perspective why transport-based sampling generation is, in practice, often better than other sampling approaches may lead to further improvements.

ACKNOWLEDGMENTS

We thank Filippo Santambrogio and the reviewers for their helpful comments. This work was partially funded by ANR-16-CE33-0026 (CALiTrOp) and ANR-16-CE23-0009 (ROOT). MD gratefully acknowledges the hospitality of ShanghaiTech University during his sabbatical.

REFERENCES

Franz Aurenhammer, Friedrich Hoffmann, and Boris Aronov. 1998. Minkowski-type theorems and least-squares clustering. *Algorithmica* 20, 1 (1998), 61–76.

Martin Bálint and Rafał K. Mantiuk. 2019. Closed Form Transmittance in Heterogeneous Media Using Cosine Noise. In *Central European Seminar on Computer Graphics*. https://www.cl.cam.ac.uk/~rkm38/pdfs/balint2019cosine_noise.pdf

Michael Balzer, Thomas Schlömer, and Oliver Deussen. 2009. Capacity-Constrained Point Distributions: A Variant of Lloyd's Method. *ACM Trans. Graph.* 28, 3 (2009), 86:1–8.

Nicolas Bonneel and David Coeurjolly. 2019. SPOT: Sliced Partial Optimal Transport. *ACM Trans. Graph.* 38, 4 (July 2019).

Nicolas Bonneel, Julien Rabin, Gabriel Peyré, and Hanspeter Pfister. 2015. Sliced and Radon Wasserstein barycenters of measures. *Journal of Mathematical Imaging and Vision* 51, 1 (2015).

Nicolas Bonnotte. 2013. *Unidimensional and evolution methods for optimal transportation*. Ph.D. Dissertation. Paris 11.

Robert Bridson. 2007. Fast Poisson Disk Sampling in Arbitrary Dimensions. In *ACM SIGGRAPH Sketches*. 22–23.

Per Christensen, Julian Fong, Jonathan Shade, Wayne Wooten, Brenden Schubert, Andrew Kensler, Stephen Friedman, Charlie Kilpatrick, Cliff Ramshaw, Marc Bannister, and et al. 2018a. RenderMan: An Advanced Path-Tracing Architecture for Movie Rendering. *ACM Trans. Graph.* 37, 3 (2018), Art. 30.

Per Christensen, Andrew Kensler, and Charlie Kilpatrick. 2018b. Progressive Multi-Jittered Sample Sequences. *Computer Graphics Forum* 37, 4 (2018), 21–33.

Robert L. Cook. 1986. Stochastic Sampling in Computer Graphics. *ACM Trans. Graph.* 5, 1 (Jan. 1986), 51–72.

Fernando De Goes, Katherine Breeden, Victor Ostromoukhov, and Mathieu Desbrun. 2012. Blue noise through optimal transport. *ACM Trans. Graph.* 31, 6 (2012), 171.

Josef Dick and Friedrich Pillichshammer. 2010. *Digital Nets and Sequences: Discrepancy Theory and Quasi-Monte Carlo Integration*. Cambridge University Press.

Mark A. Z. Dippé and Erling Henry Wold. 1985. Antialiasing through Stochastic Sampling. *Computer Graphics* 19, 3 (July 1985), 69–78.

Fredo Durand. 2011. A frequency analysis of Monte-Carlo and other numerical integration schemes. *MIT CSAIL Technical report TR-2011-052* (2011).

Raanan Fattal. 2011. Blue-Noise Point Sampling Using Kernel Density Model. *ACM Trans. Graph.* 30, 4 (July 2011), 48:1–48:12.

Iliyan Georgiev and Marcos Fajardo. 2016. Blue-Noise Dithered Sampling. 35:1–35:1. https://doi.org/10/gfznbx

Jens André Griepentrog, Wolfgang Höppner, Hans-Christoph Kaiser, and Joachim Rehberg. 2008. A bi-Lipschitz continuous, volume preserving map from the unit ball onto a cube. *Note di Matematica* 28, 1 (2008), 177–193.

Daniel Heck, Thomas Schlömer, and Oliver Deussen. 2013. Blue Noise Sampling with Controlled Aliasing. *ACM Trans. Graph.* 32, 3 (2013), 25:1–25:12.

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 Talk*.

Edmund Hlavka. 1961. Funktionen von beschränkter Variation in der Theorie der Gleichverteilung. *Annali di Matematica Pura ed Applicata* 54, 1 (1961), 325–333.

Wenzel Jakob. 2013. Mitsuba Renderer. http://www.mitsuba-renderer.org.

Wojciech Jarosz, Afnan Enayet, Andrew Kensler, Charlie Kilpatrick, and Per Christensen. 2019. Orthogonal Array Sampling for Monte Carlo Rendering. *Computer Graphics Forum* 38, 4 (2019), 135–147.

Stephen Joe and Frances Y Kuo. 2008. Constructing Sobol sequences with better two-dimensional projections. *SIAM Journal on Scientific Computing* 30, 5 (2008), 2635–2654.

Leonid Vitalevich Kantorovich. 1948. Functional analysis and applied mathematics. *Uspekhi Matematicheskikh Nauk* 3, 6 (1948), 89–185.

Leonid V. Kantorovich and Gennady S. Rubinstein. 1958. On a space of completely additive functions. *Vestnik Leningrad. Univ* 13, 7 (1958), 52–59.

Alexander Keller. 2004. Stratification by rank-1 lattices. In *Monte Carlo and Quasi-Monte Carlo Methods 2002*, Harald Niederreiter (Ed.). Springer, 299–313.

Alexander Keller. 2013. Quasi-Monte Carlo Image Synthesis in a Nutshell. In *MCQMC*, Josef Dick, Frances Y. Kuo, Gareth W. Peters, and Ian H. Sloan (Eds.). Springer, 213–249.

Johannes Kopf, Daniel Cohen-Or, Oliver Deussen, and Dani Lischinski. 2006. Recursive Wang Tiles for Real-Time Blue Noise. *ACM Trans. Graph.* 25, 3 (2006), 509–518.

Christopher Kulla, Alejandro Conty, Clifford Stein, and Larry Gritz. 2018. Sony Pictures Imageworks Arnold. *ACM Trans. Graph.* 37, 3 (2018).

Ares Lagae and Philip Dutré. 2006. An Alternative for Wang Tiles: Colored Edges versus Colored Corners. *ACM Trans. Graph.* 25, 4 (Oct. 2006), 1442–1459.

Pierre L'Ecuyer and David Munger. 2016. Lattice Builder: A General Software Tool for Constructing Rank-1 Lattice Rules. https://github.com/umontreal-simul/latnetbuilder.

Christiane Lemieux. 2009. *Monte Carlo and Quasi Monte Carlo Sampling*. Springer-Verlag New York.

Bruno Lévy. 2015. A numerical algorithm for L2 semi-discrete optimal transport in 3D. *ESAIM: Mathematical Modelling and Numerical Analysis* 49, 6 (2015), 1693–1715.

Quentin Mérigot. 2011. A multiscale approach to optimal transport. In *Computer Graphics Forum*, Vol. 30. Wiley Online Library, 1583–1592.

Don P. Mitchell. 1987. Generating Antialiased Images at Low Sampling Densities. *Computer Graphics* 21, 4 (Aug. 1987), 65–72.

Don P Mitchell. 1991. Spectrally optimal sampling for distribution ray tracing. *Computer Graphics* 25, 4 (1991), 157–164.

Patrick Mullen, Pooran Memari, Fernando de Goes, and Mathieu Desbrun. 2011. HOT: Hodge-optimized Triangulations. *ACM Trans. Graph.* 30, 4 (2011), Art. 103.

Georges Nader and Gael Guennebaud. 2018. Instant transport maps on 2D grids. *ACM Trans. Graph.* 37, 6 (2018), 13.

Jan Novák, Iliyan Georgiev, Johannes Hanika, and Wojciech Jarosz. 2018. Monte Carlo Methods for Volumetric Light Transport Simulation. 37, 2 (2018).

Victor Ostromoukhov, Charles Donohue, and Pierre-Marc Jodoin. 2004. Fast hierarchical importance sampling with blue noise properties. In *ACM Trans. Graph.*, Vol. 23. ACM, 488–495.

Art B. Owen. 1998. Scrambling Sobol's and Niederreiter-Xing Points. *Journal of Complexity* 14, 4 (1998), 466–489.

A. Cengiz Öztireli. 2016. Integration with Stochastic Point Processes. *ACM Trans. Graph.* 35, 5 (Aug. 2016), Art. 160.

Hélène Perrier, David Coeurjolly, Feng Xie, Matt Pharr, Pat Hanrahan, and Victor Ostromoukhov. 2018. Sequences with Low-Discrepancy Blue-Noise 2-D Projections. In *Computer Graphics Forum*, Vol. 37. Wiley Online Library, 339–353.

Matt Pharr, Wenzel Jakob, and Greg Humphreys. 2016. *Physically Based Rendering: From Theory to Implementation* (3 ed.). Morgan-Kaufmann.

Adrien Pilleboue, Gurprit Singh, David Coeurjolly, Michael Kazhdan, and Victor Ostromoukhov. 2015. Variance Analysis for Monte Carlo Integration. *ACM Trans. Graph.* 34, 4 (July 2015), 124:1–124:14.

Francois Pitié, Anil C. Kokaram, and Rozenn Dahyot. 2005. N-Dimensional Probablility Density Function Transfer and Its Application to Colour Transfer. In *IEEE Int. Conf. on Computer Vision (ICCV)*.

Hongxing Qin, Yi Chen, Jinlong He, and Baoquan Chen. 2017. Wasserstein blue noise sampling. *ACM Trans. Graph.* 36, 4 (2017), Art. 137.

Julien Rabin, Gabriel Peyré, Julie Delon, and Marc Bernot. 2011. Wasserstein barycenter and its application to texture mixing. In *International Conference on Scale Space and Variational Methods in Computer Vision*. Springer, 435–446.

Johann Radon. 1986. On the determination of functions from their integral values along certain manifolds. *IEEE Transactions on Medical Imaging* 5, 4 (1986), 170–176.

Bernhard Reinert, Tobias Ritschel, Hans-Peter Seidel, and Iliyan Georgiev. 2016. Projective blue-noise sampling. In *Computer Graphics Forum*, Vol. 35. Wiley Online Library, 285–295.

Mark Rowland, Krzysztof M Choromanski, François Chalus, Aldo Pacchiano, Tamas Sarlos, Richard E Turner, and Adrian Weller. 2018. Geometrically Coupled Monte Carlo Sampling. In *Advances in Neural Information Processing Systems*. Vol. 31. 195–206.

Peter Shirley and Kenneth Chiu. 1997. A low distortion map between disk and square. *Journal of Graphics Tools* 2, 3 (1997), 45–52.

Gurprit Singh and Wojciech Jarosz. 2017. Convergence Analysis for Anisotropic Monte Carlo Sampling Spectra. *ACM Trans. Graph.* 36, 4 (July 2017), 137:1–137:14.

Gurprit Singh, Kartic Subr, David Coeurjolly, Victor Ostromoukhov, and Wojciech Jarosz. 2019. Fourier Analysis of Correlated Monte Carlo Importance Sampling. *Computer Graphics Forum* 38, 1 (April 2019).

Justin Solomon, Fernando de Goes, Gabriel Peyré, Marco Cuturi, Adrian Butscher, Andy Nguyen, Tao Du, and Leonidas Guibas. 2015. Convolutional Wasserstein Distances:

Table 2. For a point $(\mathbf{x}, y) \in \mathbb{R}^d$ with $\mathbf{x} \in \mathbb{R}^{d-1}$ and $y \in \mathbb{R}$, this table summarizes the functions used in Eqs. (12) and (13) to define the mapping $\Phi_{\text{tmp}}(\mathbf{x}, y)$ of a point $(\mathbf{x}, y)$ of the unit d-ball to a point in the d-cylinder.

for $(\mathbf{x}, y) \in \mathcal{B}^{\triangledown}$	for $(\mathbf{x}, y) \in \mathcal{B}^{\triangleright\!\triangleleft}$
$h^{\triangledown}(\mathbf{x}, y) = \|(\mathbf{x}, y)\|$	$h^{\triangleright\!\triangleleft}(\mathbf{x}, y) = \|(\mathbf{x}, y)\| \displaystyle\int_0^{\arctan(y/\|\mathbf{x}\|)} \cos(\alpha)^{d-2}\, d\alpha$
$g^{\triangledown}(\mathbf{x}, y) = \begin{cases} \sqrt{\frac{y^2}{\|\mathbf{x}\|^2+1}} \left((d-1)\int_0^{\arctan(\|\mathbf{x}\|/y)} \sin(\alpha)^{d-2} d\alpha\right)^{\frac{1}{d-1}} & \text{if } \mathbf{x} \neq 0 \\ 1 & \text{if } \mathbf{x} = \mathbf{0} \end{cases}$	$g^{\triangleright\!\triangleleft}(\mathbf{x}, y) = \sqrt{1 + y^2/\|\mathbf{x}\|^2}$
$\tau(\gamma) = \left((d-1)\displaystyle\int_0^{\arctan(1/\gamma)} \sin(\alpha)^{d-2}\, d\alpha\right)^{\frac{1}{d-1}}$	$\varrho(\gamma) = \displaystyle\int_0^{\arctan(\gamma)} \cos(\alpha)^{d-2} d\alpha$

Efficient Optimal Transportation on Geometric Domains. *ACM Trans. Graph.* 34, 4 (2015), Art. 66.

Justin Solomon, Raif Rustamov, Leonidas Guibas, and Adrian Butscher. 2014. Earth Mover's Distances on Discrete Surfaces. *ACM Trans. Graph.* 33, 4 (2014), Art. 67.

Kartic Subr and Jan Kautz. 2013. Fourier analysis of stochastic sampling strategies for assessing bias and variance in integration. *ACM Trans. Graph.* 32, 4, Article 128 (2013), 12 pages.

Peter Aundal Toft and John Aasted Sørensen. 1996. The Radon transform-theory and implementation. (1996).

Salvatore Torquato, Obioma Uche, and Frank Stillinger. 2006. Random sequential addition of hard spheres in high Euclidean dimensions. *Physical Review E* 74, 6 (2006), 061308.

Robert Ulichney. 1987. *Digital Halftoning.* MIT Press, Cambridge, MA, USA.

Eric Veach. 1997. *Robust Monte Carlo methods for light transport simulation.* Stanford University PhD thesis.

Cédric Villani. 2009. *Optimal Transport: Old and New.* Fundamental Principles of Mathematical Sciences, Vol. 338. Springer-Verlag.

Li-Yi Wei. 2010. Multi-Class Blue Noise Sampling. *ACM Trans. Graph.* 29, 4 (July 2010), 79:1–79:8.

Yahan Zhou, Haibin Huang, Li-Yi Wei, and Rui Wang. 2012. Point sampling with general noise spectrum. *ACM Trans. Graph.* 31, 4 (2012), 76.

A RADON FOR UNIFORM DENSITY OVER d-BALL

In this short appendix, we provide the general form of the Radon transform of a uniform density u on a unit $d-$dimensional sphere. If $V_d(r)$ denotes the volume of a ball of radius r in dimension d, then:

$$V_d(r) \coloneqq \frac{\pi^{d/2}}{\Gamma(\frac{d}{2}+1)} r^d = K_d\, r^d \quad \text{and} \quad u(x) \coloneqq \begin{cases} 1/V_d(1) & \text{if } |\mathbf{x}| \leq 1 \\ 0 & \text{otherwise.} \end{cases}$$

By symmetry, the Radon transform R_θ of u does not depend on the chosen direction θ, so the Radon transform $Ru \coloneqq R_\theta u$ can be computed as the volume of the subspace orthogonal to any slice direction. Without loss of generality, we take $\theta \coloneqq (1, 0, 0, \ldots, 0)$. Denoting $\mathbf{x} = \{x_i\}_{i=1..d}$ the coordinates of a d-dimensional point $\mathbf{x}$, the unit ball is defined implicitly through $\{\mathbf{x} | \sum_i x_i^2 \leq 1\}$. The subspace of constant x_1 orthogonal to the chosen direction θ can thus be described through the inequality $\sum_{i=2..d} x_i^2 \leq 1 - x_1^2$. This subspace characterizes a ball of dimension $(d-1)$ and radius $\sqrt{1-x_1^2}$. We then deduce that the Radon transform of the uniform density over the $d-$dimensional ball is $Ru(s) = V_{d-1}(\sqrt{1-s^2})$, as in Eq. (8).

Our optimal transport-based solution further requires the inverse cumulative density function C_d of Ru in dimension d. We first express the cumulative density function $C_d(x)$:

$$\begin{aligned} C_d(x) &= \frac{1}{V_d(1)} \int_{-\infty}^{x} Ru(t)\, \mathrm{d}t \\ &= \frac{K_{d-1}}{K_d} \int_{-1}^{x} \left(1 - t^2\right)^{\frac{d-1}{2}} \mathrm{d}t\,. \end{aligned}$$

For odd dimensions, the above expression yields the integral of a polynomial of degree d that can be computed in closed form, while the case of even dimensions involves the use of the hypergeometric function ${}_2F_1$ — see Eq. (9) for the exact expressions.

B BALL-TO-CUBE MAP

As explained in Griepentrog et al. [2008], for a d-dimensional point $(\mathbf{x}, y) \in \mathcal{B}^{\triangledown}$ (where $\mathbf{x} \in \mathbb{R}^{d-1}$ and $y \in \mathbb{R}$), we can map it to a part of the d-dimensional cylinder using

$$\Phi^{\triangledown}_{\text{tmp}}(\mathbf{x}, y) = \left(x_1 \frac{g^{\triangledown}(\mathbf{x}, y)}{\tau(\gamma)}, \ldots, x_{d-1} \frac{g^{\triangledown}(\mathbf{x}, y)}{\tau(\gamma)}, h^{\triangledown}(\mathbf{x}, y) \right). \tag{12}$$

For a point $(\mathbf{x}, y) \in \mathcal{B}^{\Delta}$, the same expression is valid for $\Phi^{\Delta}_{\text{tmp}}$ with just a simple sign change for y, by symmetry of these two domains. Finally, for $(\mathbf{x}, y) \in \mathcal{B}^{\triangleright\!\triangleleft}$, we can use:

$$\Phi^{\triangleright\!\triangleleft}_{\text{tmp}}(\mathbf{x}, y) = \left(x_1 g^{\triangleright\!\triangleleft}(\mathbf{x}, y), \ldots, x_{d-1} g^{\triangleright\!\triangleleft}(\mathbf{x}, y), \frac{h^{\triangleright\!\triangleleft}(\mathbf{x}, y)}{\varrho(\gamma)} \right), \tag{13}$$

where the functions involved are defined in Tab. 2. In these definitions, the functions g and h are the effective maps, and ϱ and τ are just used to scale both parts of the mapping to make them coincide at their common border. The intuition behind $h^{\triangledown}$ and $g^{\triangleright\!\triangleleft}$ is to displace points from the surface of a $d-$dimensional ball onto the matching surface of the d-dimensional cylinder. $g^{\triangledown}$ and $h^{\triangleright\!\triangleleft}$ are then defined in order to attain a constant determinant of the Jacobian. As a result, these three maps creates a volume-preserving and low distortion map Φ_{tmp} from the entire sphere to the entire cube in dimension d summarized as follows: denoting $S(\mathbf{x}, y) \coloneqq (\mathbf{x}, -y)$,

$$\Phi_{\text{tmp}}(\mathbf{x}, y) = \begin{cases} \Phi^{\triangledown}_{\text{tmp}}(\mathbf{x}, y) & \text{if } (\mathbf{x}, y) \in \mathcal{B}^{\triangledown} \\ S(\Phi^{\triangledown}_{\text{tmp}}(\mathbf{x}, y)) & \text{if } S(\mathbf{x}, y) \in \mathcal{B}^{\triangledown} \\ \Phi^{\triangleright\!\triangleleft}_{\gamma_d}(\mathbf{x}, y) & \text{if } (\mathbf{x}, y) \in \mathcal{B}^{\triangleright\!\triangleleft} \\ S(\Phi^{\triangleright\!\triangleleft}_{\gamma_d}(\mathbf{x}, y)) & \text{if } S(\mathbf{x}, y) \in \mathcal{B}^{\triangleright\!\triangleleft}. \end{cases}$$

Because the d-cylinder is the tensor product of a $(d-1)$-ball and of one (e.g., the last) direction, we can compute the complete map Ball2Cube recursively through Eq. (11). Note that the original paper by Griepentrog and coauthors provide the (simple) expression for the *inverse* map from the d-cylinder to the d-sphere.