Eurographics Symposium on Rendering - Experimental Ideas & Implementations (2017)
P. Sander and M. Zwicker (Editors)

Local Quasi-Monte Carlo Exploration

Lorenzo Tessari[†], Johannes Hanika and Carsten Dachsbacher
Karlsruhe Institute of Technology, Germany.

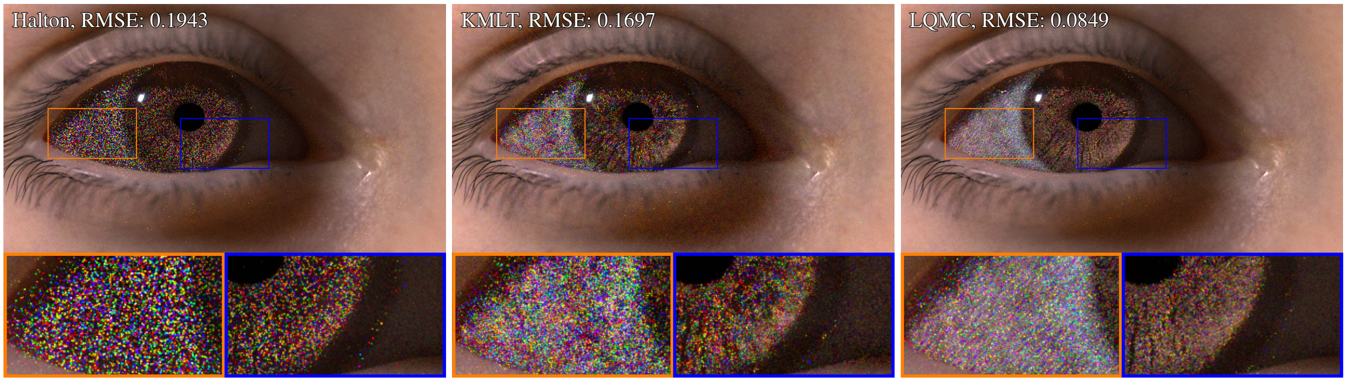

Figure 1: *Equal time comparison (15m): Regular quasi-Monte Carlo light transport (Halton points, left) can be inefficient at exploring local features such as the caustic in on the iris or the subsurface scattering in the sclera of this eye. Markov chain methods such as Kelemen Metropolis light transport (KMLT, middle) explore such features better but suffer from clumping artifacts leading to temporal inconsistency in animation. We propose a local quasi-Monte Carlo integration scheme (LQMC, right) which uses stratified point sets for local exploration of lighting features, leading to more even convergence.*

Abstract

In physically-based image synthesis, the path space of light transport paths is usually explored by stochastic sampling. The two main families of algorithms are Monte Carlo/quasi-Monte Carlo sampling and Markov chain Monte Carlo. While the former is known for good uniform discovery of important regions, the latter facilitates efficient exploration of local effects. We introduce a hybrid sampling technique which uses quasi-Monte Carlo points to achieve good stratification in both stages: we use the Halton sequence to generate initial seed paths and rank-1 lattices for local exploration. This method avoids the hard problem of introducing QMC sequences into MCMC while still stratifying samples both globally and locally. We propose perturbation strategies that prefer dimensions close to the camera, facilitating efficient reuse of transport path suffixes. This framework provides maximum control of the sampling scheme by the programmer, which can be hard to achieve with Markov chain-based methods. We show that local QMC exploration can generate results on par with state of the art light transport sampling methods, while providing more uniform convergence, improving temporal consistency.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Raytracing

1. Introduction

Physically-based image synthesis is both an important and challenging task [KFF*15]. To address this challenge, there are two main families of sampling methods: Monte Carlo and Markov chain Monte Carlo methods.

These families reflect the duality of the sampling problem, which essentially consists of two problems to solve: global discovery and local exploration. Plain Monte Carlo is typically better at global discovery of local maxima or modes of the integrand. This property can even be improved by employing quasi-Monte Carlo sampling [Nie92], reducing the Monte Carlo error.

On the other hand, Markov chain methods facilitate good local exploration by adapting sampling density to the target function without explicit inversion. This is done by keeping a previously

[†] lorenzo.tessari@kit.edu

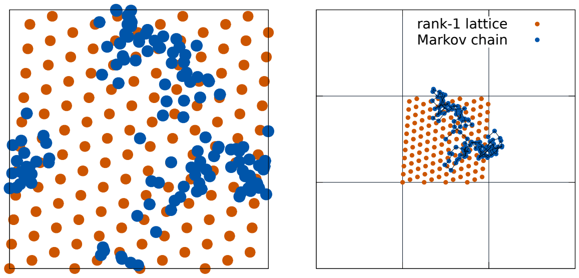

Figure 2: *Illustration how the two-dimensional unit torus (in analogy to primary sample space) is explored using $N = 128$ points, both by a rank-1 lattice with generator vector $g = (1, 12)$ [Dam09] and a Markov chain adding uniform random steps in $[-1/\sqrt{N}, 1/\sqrt{N}]$. Left: full domain, right: local resampling.*

sampled path as the state of the Markov chain, thus using more information than pure independent Monte Carlo. Often times, however, this comes at the cost of worse global discovery and stratification of samples, resulting in temporal instability of rendered frames. Also, Markov chains can be hard to control because of the iterative nature of the sampling scheme.

This work adds one more technique to the toolbox, which has properties in between those two: provide good global discovery and also stratified local exploration, with maximum control by the user. We achieve this by using regular (quasi-)Monte Carlo path tracing to sample seed paths, followed by a step where the proximity of this seed path is explored by using a stratified point set, perturbing the random number driving sequence in primary sample space. We show that this results in

- better exploration of lighting effects that are important on the image (dimensions close to camera, such as depth of field).
- efficient re-use of long paths (similar to virtual point lights).
- good stratification over pixels and improved temporal stability.
- a simple implementation, comparable to Metropolis in primary sample space.
- maximum control over the sampling process.

2. Background and Previous Work

Light transport Simulating the behaviour of light can be done by generating transport paths $\mathbf{X}$, essentially a list of path vertices where photon collision occurs. Along these, the measurement contribution function $f(\mathbf{X})$, can be evaluated. This means computing the differential flux per all area measures $d\mathbf{x}_i$ encountered on the vertices $\mathbf{x}_i$.

The rendering equation [Kaj86] computes the pixel intensity $I(p)$ for a pixel p by collecting this quantity over the path space, weighted by the pixel filter $h(p)$:

$$I(p) = \int_{\mathcal{P}} h_p(\mathbf{X}) \cdot f(\mathbf{X})\, d\mathbf{X} \tag{1}$$

Monte Carlo path sampling, such as bidirectional path tracing (BDPT) [LW93, VG94], creates a path $\mathbf{X}$ distributed according to a probability density function (PDF) $p(\mathbf{X})$. This process usually uses s-dimensional, uniformly distributed i.i.d random numbers $\mathbf{u} \in [0,1)^s$, for instance by using the inverse CDF method. The cumulative distribution function (CDF) corresponding to $p(\mathbf{X})$ is written as $P(\mathbf{X})$ and we will use $P^{-1}(\mathbf{u}) = \mathbf{X}$ to denote the path sampling process.

BDPT subsumes a set of sampling techniques t: it samples subpaths from the sensor and from the light sources and connects the two using the path segment t. We will use $p_t(\mathbf{X})$ and $P_t(\mathbf{X})$ to denote the PDF and CDF of the given technique t, respectively. To give a correct estimator, all these techniques have to be weighted together by appropriate weighting functions $w_t(\mathbf{X})$, which are best implemented using multiple importance sampling (MIS) [VG95].

Adaptive sampling To better explore difficult features of an image, often times adaptive sampling is performed, usually in image space [ZJL*15]. However, this can be inefficient: a variance buffer or similar image space considerations will only be able to guide samples in the first two dimensions of a potentially high dimensional and difficult sampling process. The most part of the path space sampling has to be performed completely randomly, usually re-tracing until a light source is found.

To achieve adaptivity in path space, guiding paths by using photon maps or dedicated guiding records [VKS*14] has been explored. This comes at a certain preprocessing and storage cost, and also needs to evaluate complete transport paths.

Distribution effects such as depth of field or motion blur, or in general effects which are easily observable because they happen close to the camera are important, but usually they require us to trace full paths to the light source. Reusing subpaths can be performed, for example, via trajectory splitting [VK16]. Our technique does this, too, but in a sense can be seen as adjoint trajectory splitting: we split the path at the camera to better explore the pixels in the image, but without starting at the light source. That is we first construct a transport path by any sampling means and then explore it after the fact.

To facilitate high-dimensional adaptive sampling in path space (i.e. importance sample the integrand which is not possible to invert directly), Markov chain methods can be used.

Markov chain methods MCMC [MRR*53, Has70] has been adopted for light transport in path space (Veach-style, VMLT) [VG97] and simplified to work on the input random numbers $\mathbf{u}$ instead of the paths directly (Kelemen-style, KMLT or primary sample space MLT) [KSKAC02].

Several specialised mutation strategies have been devised since. MMLT [HKD14] selects a technique t from the set of bidirectional methods to explore by adding knowledge about the MIS weights into the Markov chain. This resolves the issue that regular Monte Carlo estimators have when combining an excessive amount of techniques, where some of those would be inadequate and weighted close to zero by MIS, wasting computation [KGH*14].

All Markov chain-based methods usually share the drawback that they tend to thoroughly explore local regions and neglect global discovery, leading to temporal flickering.

Energy redistribution path tracing (ERPT) [CTE05], originally combined with Veach-style mutations, was the first technique to incorporate globally stratified independent samples with short bursts of local Markov chains. This is currently the state of the art when combining quasi-Monte Carlo independent samples with local exploration by a Markov chain. We are devising a technique to perform the local exploration via stratified sampling, too.

While this could be achieved with multiple-try Metropolis [LLW00] and general balance methods [ST10], we chose a different approach and are not using Markov chains at all. We compute the conditional probabilities of a dependent sample directly by replacing the Markov chain by a local lattice rule (see Fig. 2).

Quasi-Monte Carlo extensions For an excellent introduction to quasi-Monte Carlo we refer the reader to [Nie92]. Quasi-Monte Carlo point sets can be readily dropped into a light transport program by simply replacing the pseudo random number generator. Local exploration or reordering computation for tiling (for example) can be performed via sample enumeration [GRK12]. This method, however, requires good quality $(0,m,s)$-nets which are hard to come by. Also, when moving to higher dimensions ($s = 5..10$) the number of points quickly becomes a problem due to the construction of elementary intervals via a Cartesian product. We need points with good distribution properties where the number of points N is about in the same range as the number of dimensions s.

There has been work on including quasi-Monte Carlo sampling into MCMC [CMNO10], even on continuous state spaces [CDO11], using completely uniformly distributed driving sequences. To gain more control over how many samples are in every sequence based on the number of dimensions, we removed the Markov chain theory from our estimators completely.

Also from quasi-Monte Carlo literature, we will apply Cranley-Patterson rotations [CP76] to extend a lattice rule with few sample points to the full continuous sampling domain.

For our purpose, padded replications [KK02] of two- or three-dimensional point sets could potentially have been adopted. We opted to use one rank-1 lattice for all dimensions of the perturbation together instead because it is conceptually simpler to compute one single multidimensional lattice and has more freedom with respect to the number of lattice points.

Rank-1 lattices A rank-1 lattice is a s-dimensional lattice with a one dimensional indexing scheme. To be precise, rank-1 lattices are defined on the s-dimensional unit hypercube by a generator vector $\mathbf{g} \in \mathbb{N}^s$, and the j-th lattice point can be written as

$$\mathbf{r}_j = \left\{ \frac{j}{N} \cdot \mathbf{g} \right\}_1, \qquad (2)$$

where N is the number of points in the lattice and the braces indicate a modulo one operation. An example can be seen in Fig. 2 (left, the orange points). An important property for us is that the Voronoi cells around the points $\mathbf{r}_j$ are congruential (on the unit torus). To obtain good generator vectors $\mathbf{g}$ in high dimensions, we use the component-by-component construction scheme [NC06].

Path reuse Most bidirectional methods perform some kind of path reuse. Sometimes just the vertices of one light path are reused for all vertices of an eye path [LW93, VG94]. Photon maps cache many light paths and connect them to all eye paths [HPJ12, GKDS12]. Virtual point lights use extreme caching of very few light paths in the same context [DKH*13].

Multi frame rendering [MFSSK06] uses a deterministic offset in a discrete dimension, to reconnect paths to multiple frame buffers. Our technique includes this as a single dimension in our perturbation. Moreover, we propose a scheme that works on continuous dimensions such as aperture position, shutter time, outgoing directions from secondary path vertices or anything that is represented by the primary sample space formulation of light transport.

Dependent sampling Manifold next event estimation (MNEE) [HDF15] solves a related problem as we do when evaluating the PDF of a perturbed path. The main difference is that in MNEE the seed path is not an admissible sample. It would be interesting, however, to include this technique in our framework.

Hero wavelength sampling [WND*14] also traces a path with a single wavelength (the hero wavelength) as seed path, and explores a stratified set of neighbouring wavelengths, by performing an MIS combination. The theoretical basis of this is the same as what we use in this work, however it is limited to one specific dimension (wavelength). The extension to arbitrary dimensions comes with a few challenges we will address in the following. Hero wavelength sampling is a one-dimensional subset of the technique we propose.

Gradient domain path tracing [KMA*15] constructs shifted paths through neighbouring pixels. To be able to use the contribution of these dependent paths, it accumulates the seed and the offset path into a separate buffer, in form of a gradient. This gradient is used in a post process solving a Poisson equation to reconstruct the final image. The dependent samples we are using are very similar in spirit but our framework provides more freedom for sample placement: we are not bound to a tensor product construction (enumerate all neighbouring pixels) and can thus use any number of samples for the local exploration. We can also explore every dimension of path space without requiring to solve a Poisson problem on it later.

3. Stratified Local Exploration

In this section, we describe a sampling technique which will be called local quasi-Monte Carlo integration (LQMC) in this paper.

Outline A Markov chain creates correlated samples by considering a very small state space (the current path sample $\mathbf{X}_s$ only). We, on the other hand, define a local region in path space around the current path, which will be sampled by a set of stratified sample points directly. These points come from a quasi-Monte Carlo lattice rule, i.e. have very good properties with respect to minimum distance between the samples, which avoids clumping artifacts often encountered in Markov chain methods (see Fig. 2).

Although we use a deterministic lattice rule to offset the points in primary sample space, we interpret the offset from the seed path to the closest point in the lattice as a Cranley-Patterson shift [CP76], which allows us to sample the complete continuous domain.

The formal framework to derive a correct Monte Carlo estimator given this set of dependent samples is multiple importance sampling. To illustrate the idea, let us consider a case where we will explore all dimensions s of a path across the full domain size. We will extend this to adaptive dimensionality $\bar{s} \ll s$ and local exploration in the next sections.

Full perturbation of all dimensions Every path $\mathbf{X}_i$ is a member of a fixed set of paths $R = \{\mathbf{X}_0, \cdots \mathbf{X}_i \cdots, \mathbf{X}_{N-1}\}$ related to a rank-1 lattice (we will use subscript i for a specific path and j for any of those). This set is determined by considering the primary sample space location $\mathbf{u}_s$ of the seed path $\mathbf{X}_s \in R$. We find the closest lattice point $\mathbf{r}_s$ of a rank-1 lattice of choice which has the same number of dimensions as $\mathbf{u}_s$ (again, we will lift this restriction later). The primary sample space coordinates $\mathbf{u}_{j \neq s}$ of the other $N-1$ paths are then determined by the lattice points $\mathbf{r}$ as

$$\mathbf{u}_j = \mathbf{r}_j + (\mathbf{u}_s - \mathbf{r}_s). \tag{3}$$

Here, we can interpret the offset $(\mathbf{u}_s - \mathbf{r}_s)$ as a Cranley-Patterson shift. The transition to every dependent sample $\mathbf{X}_{j \neq s}$ from $\mathbf{X}_s$ is thus deterministic in primary sample space. In general, the PDF $p(\mathbf{X}_i)$ of a path sample $\mathbf{X}_i$ after one step of perturbation can be computed by a marginal integral:

$$p(\mathbf{X}_i) = \int_{\mathcal{P}} p(\mathbf{X}_i | \mathbf{X}_s) \cdot p_t(\mathbf{X}_s) \, \mathrm{d}\mathbf{X}_s, \tag{4}$$

where $p_t(\mathbf{X}_s)$ is the PDF of the technique we use to generate the seed sample $\mathbf{X}_s$. The deterministic conditional PDF of sampling any member of the set R is known in primary sample space measure $\mathrm{d}\mathbf{u}$

$$p_{\mathrm{d}\mathbf{u}}(\mathbf{u}_i | \mathbf{u}_s) = \frac{1}{N} \sum_{j=0}^{N-1} \delta_{\mathrm{d}\mathbf{u}}(\mathbf{u}_i - (\mathbf{r}_j + (\mathbf{u}_s - \mathbf{r}_s))) \tag{5}$$

$$= \frac{1}{N} \sum_{j=0}^{N-1} \delta_{\mathrm{d}\mathbf{u}}(\mathbf{u}_i - \mathbf{u}_j). \tag{6}$$

Note that $\mathbf{u}_s$ does not appear on the right hand side: the connection between $\mathbf{u}_s$ and $\mathbf{u}_j$ in this equation is that they both belong to the same set R, i.e. $\mathbf{u}_s$ implicitly appears as one element of the sum on the right hand side. To obtain a PDF in vertex area measure, we need to evaluate the Jacobian determinants:

$$p(\mathbf{X}_i | \mathbf{X}_s) = \left\| \frac{\mathrm{d}\mathbf{u}_i}{\mathrm{d}\mathbf{X}_i} \right\| \cdot \frac{1}{N} \sum_{j=0}^{N-1} \delta_{\mathrm{d}\mathbf{u}}(\mathbf{u}_i - \mathbf{u}_j) \cdot \left\| \frac{\mathrm{d}\mathbf{X}_j}{\mathrm{d}\mathbf{u}_j} \right\|. \tag{7}$$

Due to the Dirac delta $\delta_{\mathrm{d}\mathbf{u}}$, the integral in Eq. (4) reduces to a sum:

$$p(\mathbf{X}_i) = \left\| \frac{\mathrm{d}\mathbf{u}_i}{\mathrm{d}\mathbf{X}_i} \right\| \frac{1}{N} \sum_{j=0}^{N-1} \left\| \frac{\mathrm{d}\mathbf{X}_j}{\mathrm{d}\mathbf{u}_j} \right\| p_t(\mathbf{X}_j). \tag{8}$$

This also degenerates gracefully for the special case of $i = j$ when the index i matches the seed sample s in set R. Finally we compute:

$$I_i = \frac{w_t(\mathbf{X}_i) \cdot f(\mathbf{X}_i)}{\left\| \frac{\mathrm{d}\mathbf{u}_i}{\mathrm{d}\mathbf{X}_i} \right\| \sum_{j=0}^{N-1} \left\| \frac{\mathrm{d}\mathbf{X}_j}{\mathrm{d}\mathbf{u}_j} \right\| p_t(\mathbf{X}_j)} \quad \forall \, \mathbf{X}_i \in R \tag{9}$$

for all $\mathbf{X}_i \in R$ and accumulate these N contributions to their respective pixel locations. Here, $w_t(\mathbf{X}_i)$ represents the MIS weight for sampling the perturbed path $\mathbf{X}_i$. Note that the factor $\frac{1}{N}$ in Eq. (8) cancels out with the normalisation over all N sample contributions.

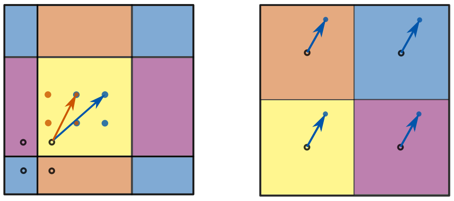

Figure 3: *Left: a point set which is not a lattice cannot be used in our framework, since it does not reproduce consistent Cranley-Patterson shift vectors due to Voronoi cells which are not congruential. Right: A lattice fulfills these requirements.*

We will derive the Jacobians for the early-reconnection strategies we use in practise in the next sections. Using this sum of PDFs to derive the estimator equates to the balance heuristic [VG95]. Note that this is very similar to the formalism used in multi frame rendering [MFSSK06] and hero wavelength sampling [WND*14, Eq. (6)], only that our approach works for both discrete dimensions (for instance frame buffer selection) as well as continuous ones (aperture position, time, or outgoing directions).

A note on lattices Ideally, we would like to have the freedom to choose any point set for the local exploration. One reason why we use rank-1 lattices is that they allow us to circumvent the curse of dimensionality that would come with Cartesian product construction: we can choose N independent of dimension $\bar{s}$. Another important aspect is that they allow us to find the Cranley-Patterson shift by simply searching for the closest point in the lattice from the seed path coordinate: the shift will be the same when recomputing it for a different seed path which is a member of the same set R. This is because the Voronoi cells around each rank-1 lattice point are congruential. Compare the two point sets illustrated in Fig. 3: In the left set, two points in the shifted set R (marked as blue circles) will find the same closest lattice point (empty circle) because they are in the same Voronoi cell. This will result in an inconsistent Cranley-Patterson shift (the orange and blue arrows do not agree). In the right set, all members of the offset lattice agree on the shift after finding their closest unshifted lattice point.

3.1. Enforcing subpath reuse and locality

Tiling the s-dimensional domain and sampling all individual tiles via a lattice rule will eventually fill the full domain with lattice points. In Fig. 4, this corresponds to stacked copies of the orange region until the whole domain (along the y-axis in the figure) is filled. Likewise, the orange region will span all the way to the right, if we are exploring all dimensions of the sampling domain.

Our approach makes sense only because of two additional aspects: a) we only explore a subset of dimensions $\bar{s} \ll s$ to facilitate path reuse, by breaking the path into a static part and one that is to be explored (see Fig. 5). And b) we want to achieve high MIS weights (similar to acceptance rate in MCMC) by increased locality of the samples within the same set R. As a side effect locality helps memory accesses and speeds up memory-heavy renders.

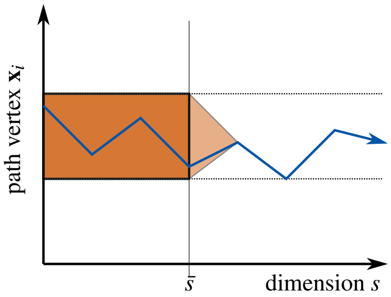

Figure 4: *Parallel coordinates visualisation (akin to [Kaj86, p. 148]) of path space and the subdomain we explore by a lattice rule (orange). To increase locality, we aim for both small $\bar{s}$ and small tile size (vertical size of the orange region in the illustration).*

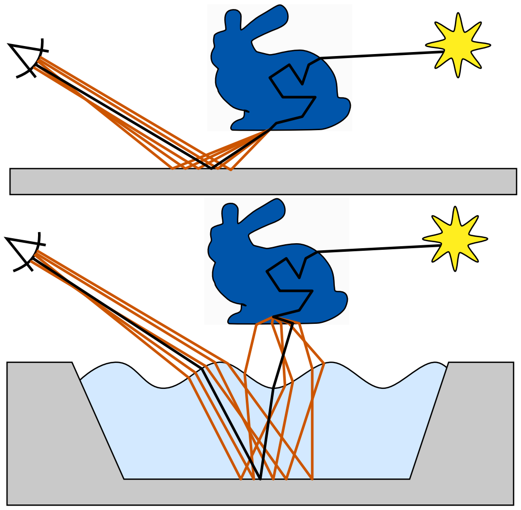

Figure 5: *Illustration of exploring the first dimensions using a local rank-1 lattice. Top: simple case where only a few dimensions are needed (wavelength, time, subpixel, aperture position and pixel coordinate). Bottom: more complex example with paths that encounter near-specular events. The subsurface scattering subpath through the bunny is reused in both cases.*

Sub path reuse We want to only explore a small subset of dimensions $\bar{s} \ll s$, corresponding to a lower dimensional vector of random numbers in primary sample space $\bar{\mathbf{u}}$. To this end, we choose a breakup point $\mathbf{x}_b$ from the path vertices (similar to [KMA*15, HKD15]). To maximise path reuse, it should be as close to the camera as possible (see Fig. 5).

We start to number vertices at the camera, so $b = 0$ means we reconnect only to a perturbed position on the camera. This can include dimensions for wavelength, aperture position, time, and frame buffer in case of multi-frame rendering ($\bar{s} = 5$). In case $b = 1$, the first point in the scene is perturbed, too, which includes up to three additional dimensions in the lattice ($\bar{s} = 8$): sensor position, which results in a changed outgoing direction, and distance for free path sampling in the medium, if applicable. Every following vertex will consume up to three additional random numbers: two for outgoing direction if the material is not specular, and one for free path sampling, if the connecting edge is in the volume. We do not change the random numbers of the seed path responsible for Russian roulette or BSDF layer selection.

The Jacobians in Eq. (8) are computed as follows. In simple cases, such as for a breakup point $b = 0$, perturbing the path vertex on the aperture, both primary sample space and path space are equally distributed and the Jacobians are not necessary. In the other extreme, when resampling all dimensions, i.e. if $\bar{s} = s$, the Jacobian from primary sample space to path space is just the PDF of the primary path sampler $\left\| \mathrm{d}\mathbf{u} / \mathrm{d}\mathbf{X} \right\| = p_t(\mathbf{X})$ and Eq. (8) becomes, only for this special case

$$p(\mathbf{X}_i) = p_t(\mathbf{X}_i) \cdot M, \quad \text{if } \bar{s} = s, \tag{10}$$

where $M \leq N$ is the number of paths $\mathbf{X}_j$ in the set with $p_t(\mathbf{X}_j) > 0$.

Analogously to how the PDF of a complete path defines the Jacobian from primary sample space to path space, we are interested in computing this density transform only up to the breakup point $\mathbf{x}_b$, i.e. we compute the Jacobian for the subpath $\bar{\mathbf{X}}$ that is defined by the subset of the random numbers $\bar{\mathbf{u}}$:

$$\left\| \mathrm{d}\bar{\mathbf{u}} / \mathrm{d}\bar{\mathbf{X}} \right\| = p_t(\bar{\mathbf{X}}). \tag{11}$$

We require that this subpath be constructed from the eye, such that it can be recomputed from $\bar{\mathbf{u}}$ (and the additional fixed random numbers for Russian roulette and layer selection, as mentioned above).

Locality To force the perturbed samples to be spatially close together, we use an implicit tiling of the primary sample space and make sure a perturbation stays within this $\bar{s}$-dimensional tile. Formally, we define a function that maps a primary sample space point $\bar{\mathbf{u}}$ to the normalised space of the tile containing $\bar{\mathbf{u}}$. The coordinates $[0,1)^{\bar{s}}$ will only address the tile in question (Fig. 2, right):

$$T : \bar{\mathbf{u}} \mapsto (\bar{\mathbf{u}} - \mathbf{o}) / \mathbf{k}, \tag{12}$$

where all operations are performed component-wise. Since the tile offset $\mathbf{o}$ is chosen such that $\bar{\mathbf{u}}$ is contained in the tile, $T(\bar{\mathbf{u}}) \in [0,1)^{\bar{s}}$. The number of tiles per dimension $\mathbf{k} \in \mathbb{N}^{\bar{s}}$ is a user parameter (we will discuss the impact and possible choices later).

3.2. Algorithm

The core technique is laid out in Alg. 1. As input, it requires the random numbers $\mathbf{u}$ and the technique t to construct the seed path from. This listing assumes a pre-selected rank-1 lattice configuration. That is, the breakup point b has already been chosen for this type of input path, determining the adaptive dimensionality $\bar{s}$. Also the tiling parameters $\mathbf{k}$ are fixed, so we can evaluate $T(.)$, and the number of points in the lattice N is known. We can thus select a rank-1 lattice from a set of precomputed generator vectors.

First, we determine the index i of the rank-1 lattice point which is closest to the truncated random numbers $\bar{\mathbf{u}}$, transformed to tile space $T(\bar{\mathbf{u}})$. After that, we iterate over all N points in the lattice and compute the random number vector $\mathbf{u}_j$ by applying the Cranley-Patterson shift in tile space and backtransforming to primary sample space (line 7). From these, we can create a new path prefix $\bar{\mathbf{X}}_j$

which is then reconnected to the static postfix of the seed path (connecting the breakup point $\mathbf{x}_b$ to $\mathbf{x}_{b+1}$, if the path has any more vertices left after the breakup point). To accumulate the final contribution of the path, we need to evaluate Eq. (8). For this we first accumulate the sum into p_s and then run the loop again to compute the final contribution value.

In theory, any path sampling technique t can be used as seed sampler. The only requirement in this formulation is that it provides enough random numbers for a path tracing prefix up to the chosen breakup point. We could lift this restriction, however, by applying a reverse mapping in an initial step to obtain the random numbers [Pan17].

Note that if the seed sampling technique t handed to Alg. 1 is combined with others in an MIS context, the individual contributions of the perturbed paths have to be weighted accordingly. This is done by the multiplication by the MIS weight $w_t(\mathbf{X}_j)$ in line 15.

Algorithm 1 Local quasi-Monte Carlo sampling

```
 1: procedure SAMPLE($\mathbf{u}, t$) // random numbers and technique
 2:     // get index of closest lattice point in tile determined by $\bar{\mathbf{u}}$
 3:     $i \leftarrow \mathrm{argmin}_j \, \|\mathbf{r}_j - T(\bar{\mathbf{u}})\|$
 4:     $p_s \leftarrow 0$
 5:     for $j = 0..N-1$ do
 6:         // create offset sample, modulo tile containing $\bar{\mathbf{u}}$
 7:         $\mathbf{u}_j \leftarrow T^{-1}\left(\{\mathbf{r}_j + (T(\bar{\mathbf{u}}) - \mathbf{r}_i)\}_1\right)$      // Eq. (3) on tile
 8:         // create path prefix from random numbers
 9:         $\bar{\mathbf{X}}_j \leftarrow P_t^{-1}(\bar{\mathbf{u}}_j)$
10:         $\mathbf{X}_j \leftarrow$ reconnect $\bar{\mathbf{X}}_j$ to static path postfix
11:         $J_j \leftarrow \left\| \mathrm{d}\bar{\mathbf{u}}_j / \mathrm{d}\bar{\mathbf{X}}_j \right\|$          // collect parts of Eq. (8)
12:         $p_s \leftarrow p_s + p_t(\mathbf{X}_j)/J_j$
13:     end for
14:     for $j = 0..N-1$ do
15:         splat path $\mathbf{X}_j$ with $w_t(\mathbf{X}_j) \cdot f(\mathbf{X}_j)/(J_j \cdot p_s)$     // Eq. (9)
16:     end for
17: end procedure
```

3.3. Performance considerations

Adaptive number of samples Local exploration without adaptive choice of the number of samples N would just amount to a reordering of computations. While this can be useful, our technique shows much more benefit when dynamically choosing between a set of rank-1 lattices with different sample count N for different tiles.

While we would like to pick the number N stochastically, this would lead to a more complex computation of the marginal PDF: we would need to recreate all possible seed paths of all lattices that may be connected to a certain sample path.

We can, however, choose N based on a class of paths. All paths in the set R must fall into the same class, such that all of them, if used as seed path, will decide for the same number of lattice points N. This is very similar to the degrees of freedom we have when choosing the number of mutations for ERPT: in both cases perturbed paths which fall out of the class need to be rejected.

Adaptive dimensionality As pointed out in Sec. 3.1, we want to facilitate efficient subpath reuse and thus in general choose $\bar{s} \ll s$. We want to make the choice stochastic: to sample b, we build a CDF based on the roughness of the vertices $\mathbf{x}_b$ and $\mathbf{x}_{b+1}$.

Explore only if needed Some paths have low variance contribution and are acceptable to be accumulated to the frame buffer directly. In simple cases such as three-vertex paths with next event estimation which are in focus, we avoid local exploration altogether, speeding up the render (see the next section for details).

3.4. Implementation Details

Jittered tiling Exploring each tile of the truncated primary sample space with a separate lattice rule may lead to visible changes in noise characteristics at tile boundaries (see Fig. 6, left). Also, assuming that image features are locally Gaussian in shape, using a fixed tiling pattern does not guarantee a good windowing of such a feature, since it may be cut off in the middle. For these reasons, we apply a global shift vector to the tiling pattern, which is randomised after every progression in the render (i.e. after realising one sample per pixel). This $\bar{s}$-dimensional offset is $\in [0, 1/\mathbf{k})$. Intuitively, this corresponds to shifting the vertical bounds of the orange box in Fig. 4 up and down for every dimension individually, resulting in a first-order b-spline smoothing (applying two random shifts would be second order [SSA05]). See Fig. 6 (right) for the effect of this smoothing.

Seed sampling To drive the seed sampling, we use the Halton sequence (as implemented by [GRK12]) up to 256 dimensions and apply the Mersenne twister after that. While it is easily possible to create adverse effects by combining multiple different quasi-Monte Carlo point sets, we did not observe any bad correlation artefacts coming from this. We believe this is due to the almost random nature of the Halton sequence and the fact that we use them in the context of a Cranley-Patterson rotation.

Local sampling We precompute rank-1 lattice generator vectors using the Matlab code provided with the component-by-component construction [NC06] for $N = 7, 23, 97$ and 313, each for $\bar{s} = 1 \dots 21$. To find the closest lattice point (as required by line 3 in Alg. 1) we perform a brute force search with early out per dimension. This process may potentially be sped up using a reduced Minkowski basis, as done for 2D textures [Dam09]. This has not been a performance concern for us. For instance, in the door scene using $N = 313$, the closest point search amounts to 0.21% of the runtime).

Parameters To determine the parameters for LQMC, we first classify transport paths into four categories: *normal, caustic, depth of field, next event estimation*. Caustic paths are detected by a (near) diffuse interaction followed by a (near) specular chain towards the light source. Depth of field is selected if the first vertex in the scene is out of focus, determined by the circle of confusion of the camera and a threshold value. If the seed path sampling technique successfully used next event estimation, we are very careful not to explore the sample (except for depth of field and complex cases such as the door scene).

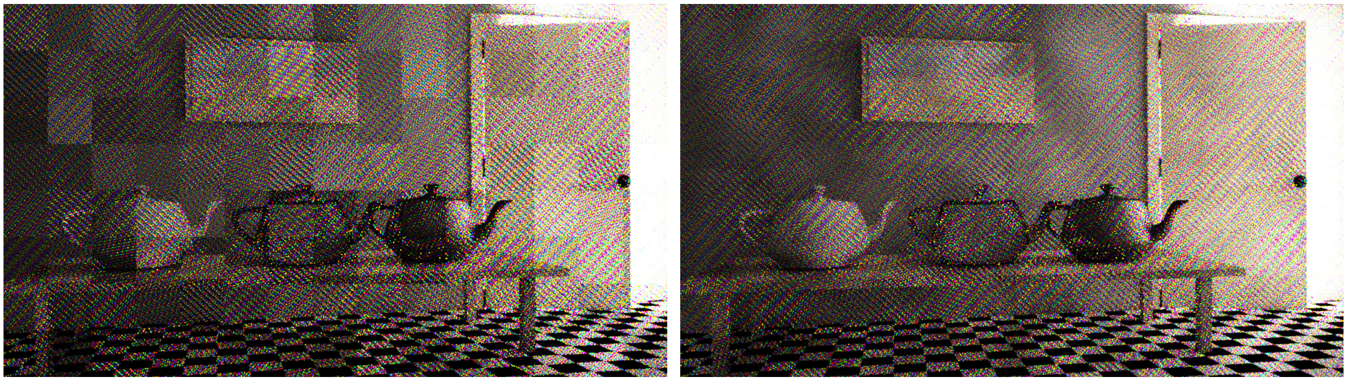

Figure 6: *Effect of jittering the tiling pattern. Left: without, right: with jittering enabled. While the left image does converge to the right solution, the noise differences at tile boundaries may remain visible for a long time without jittering and show objectionable shower door artefacts in animations (since the tile boundaries would stay in the exact same spot, distracting from the image content).*

To choose the breakup point b (and determine the truncated dimensionality $\bar{s}$), we construct a CDF based on the minimum roughness of the path vertices $\mathbf{x}_i$ and $\mathbf{x}_{i+1}$, for all inner vertices $\mathbf{x}_i$. Depth of field paths will have a higher probability to connect to $b = 0$ and caustic paths will have a high probability to connect only after the specular chain, which often means b is the end vertex of the path on the light source. This case means our technique degenerates to a full multichain perturbation or KMLT seeded by plain path tracing.

In this first implementation of LQMC, we also use these categories to determine tile sizes and sample counts. Our strategy to select tile sizes roughly corresponds to the mutation step sizes per dimension in our implementation of KMLT. Normal paths divide the aperture dimensions into 2×2 tiles, the image plane into 24 tiles in width and a value corresponding to this and the aspect ratio in height. The wavelength domain only uses one tile, and thus always mutates in the full spectral range. We divide the outgoing direction into 50–150 tiles depending on whether the path has been classified as caustic or not. Remember that tile space is in primary sample space and actual outgoing directions will be computed by importance sampling the BSDF. Caustic paths also use a finer subdivision of the image plane, we use 48 tiles here. Distances in participating media are explored in very small steps, for our renderings we use 1000 tiles in this dimension.

As for sample counts, we use $N = 23$ for normal and depth of field paths, while caustics receive an increased sample count of $N = 97$. The *ajar door* scene shown in the results uses a fixed N as indicated in the figure.

4. Results

Fig. 7 shows a simple diffuse scene to illustrate the effect of exploring dimensions one by one, including wavelength [WND*14] as a subset, as well as camera motion blur. The latter is essentially multi-frame rendering [MFSSK06] on a continuous scale, we could have connected to a different frame buffer just the same. Note that our current implementation only supports camera motion blur. Moving geometry would require to ray trace all path segments again, to keep visibility consistent.

Fig. 8 is the classic *ajar door* scene featuring difficult visibility between the light source in one room and the camera in another. Our technique, though very simple, delivers output of quality comparable to top-of-the-line rendering algorithms. It manages to efficiently reuse path postfixes to the light source and explore them close to the camera. We use $N = 313$ in this render and pick the breakup point $1 \leq b \leq 4$, depending on the surface roughness. Due to the large dynamic range of this scene the plain root mean squared error (RMSE) values were dominated by few bright pixels, and we report the mean of absolute differences (MAD) instead.

Fig. 9 shows a scene with heavy blur due to depth of field. LQMC splits the seed path trajectory at the camera and reuses long transport paths efficiently.

Figs. 1 and 10 show a difficult sampling configuration: The caustic on the displaced iris make uniform exploration of the path space very hard for all techniques. This is easier to see at low sample counts, where pure MC methods have problems finding the right configuration and MCMC remains stuck with clumping artefacts. Our simple heuristic is able to detect caustic paths and increase N to better explore the local feature, while adjusting the tile sizing accordingly. This allows us to obtain a visually pleasant image and a low RMSE even from the start of the computation. Fig 11 shows the sample density of LQMC. It finds the important regions of the image, while keeping overall stratification. Note that the eye scene is using non-reciprocal BSDFs both on the skin and the sclera, so no bidirectional methods are demonstrated here.

Fig. 12 demonstrates a volumetric caustic. In this setting sampling benefits from light tracing, all relevant parts of the path space are well explored by this technique (as is evident from the bidirectional methods BDPT and MMLT [HKD14] in the picture). We show that, even starting from a bad sampling strategy (plain path tracing in our case) LQMC is able to reconstruct and search important regions that dominate the integral.

To evaluate temporal stability, we provide a video with equal time comparisons of unconverged renders of the depth of field test scene and the eye as supplemental material (for each scene looping ten identical frames with different random seeds).

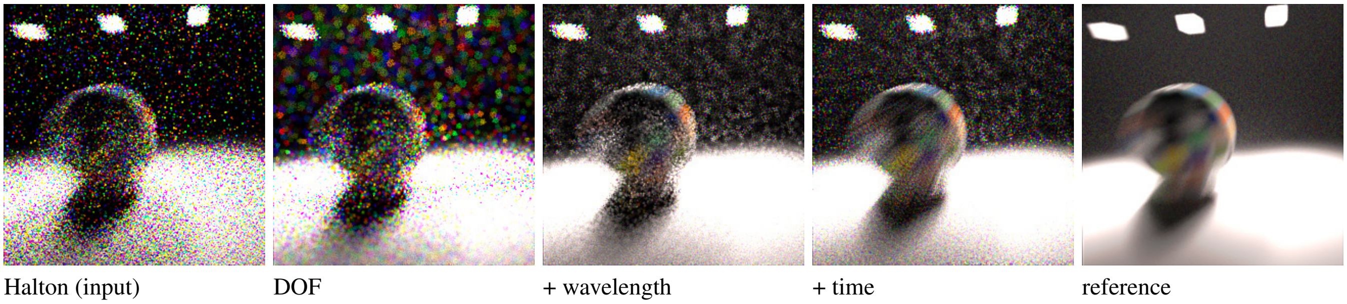

Figure 7: *Simple diffuse test scene to demonstrate distribution effects. Left to right: input Halton points, aperture only ($\bar{s} = 2$) shows circles of confusion in the same colour dictated by the constant wavelength, hero wavelength and depth of field ($\bar{s} = 3$), all that plus time ($\bar{s} = 4$). This scene is dominated by direct light, so we do not demonstrate outgoing direction here. All images use 8 spp and $N = 23$.*

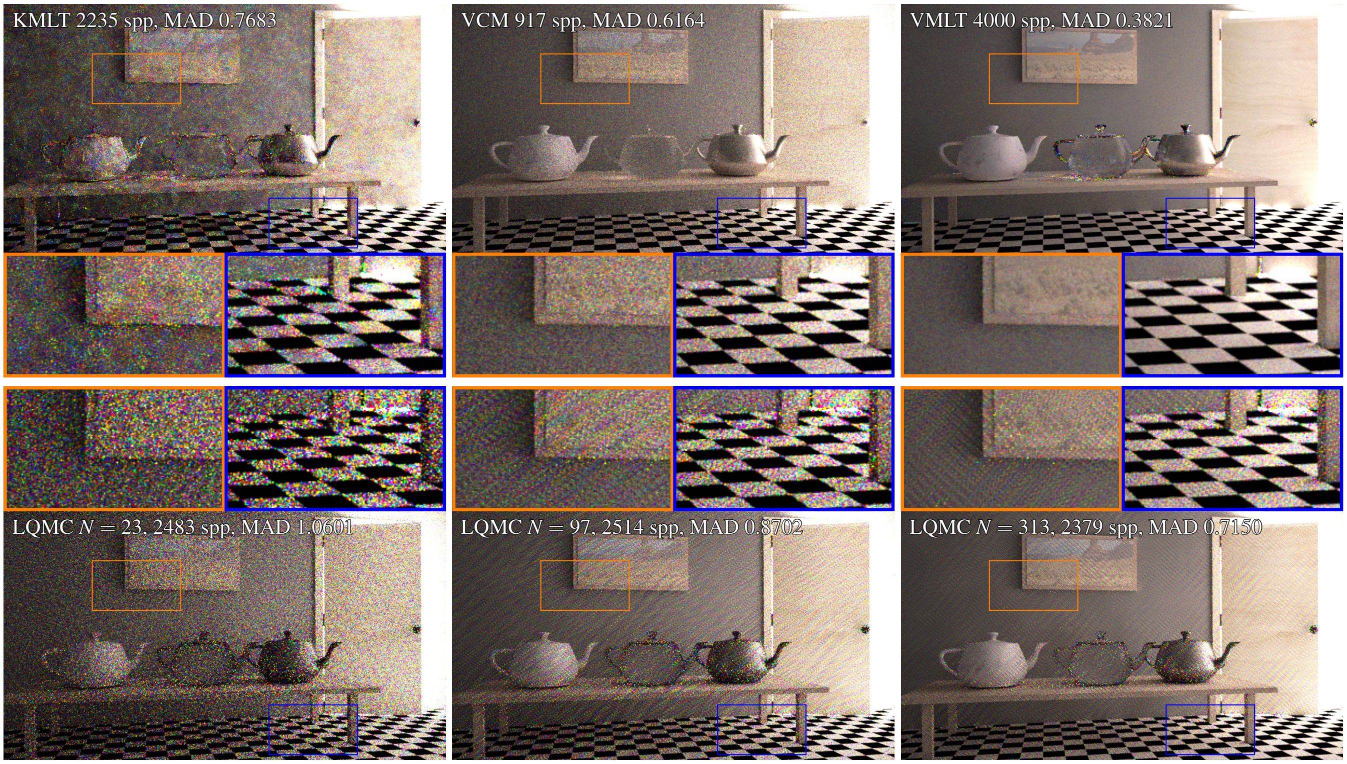
KMLT 2235 spp, MAD 0.7683
VCM 917 spp, MAD 0.6164
VMLT 4000 spp, MAD 0.3821
LQMC N = 23, 2483 spp, MAD 1.0601
LQMC N = 97, 2514 spp, MAD 0.8702
LQMC N = 313, 2379 spp, MAD 0.7150

Figure 8: *Equal time comparison in the* ajar door *scene (roughly 17 core hours on an Intel Xeon 2.50GHz). VCM used 70MB of additional memory for the photon map. We used $N = 23, 97, 313$, which means that for every successful sample in the 2379 samples per pixel, LQMC does N reconnections to the camera. Due to locality and reuse of computations, these are relatively cheap to evaluate. More local exploration can more efficiently reuse long transport paths in this scene, but also make rank-1 lattice structures show more. Because of high dynamic range and firefly outliers, we report the mean absolute deviation (MAD) as error measure for this scene.*

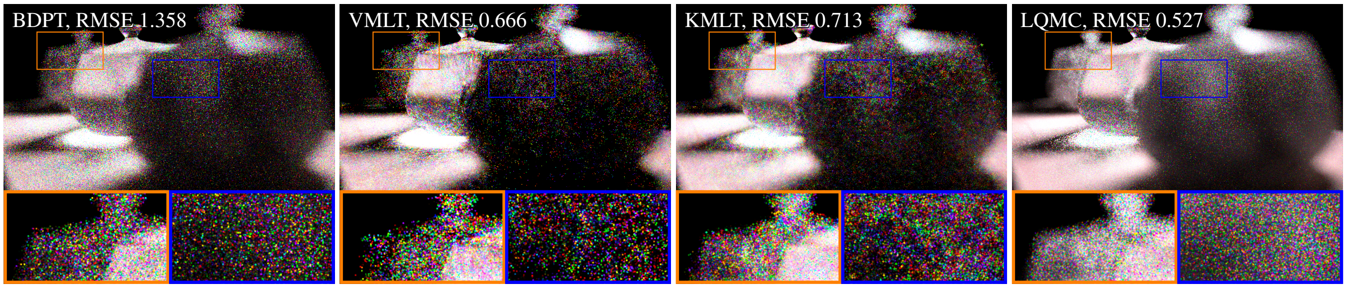

Figure 9: *Three rough translucent teapots with shallow depth of field (equal time comparison after one minute). LQMC is able to efficiently reuse the subsurface scattering paths through the teapots by early reconnecting them to the aperture. We recommend looking at the images electronically on screen.*

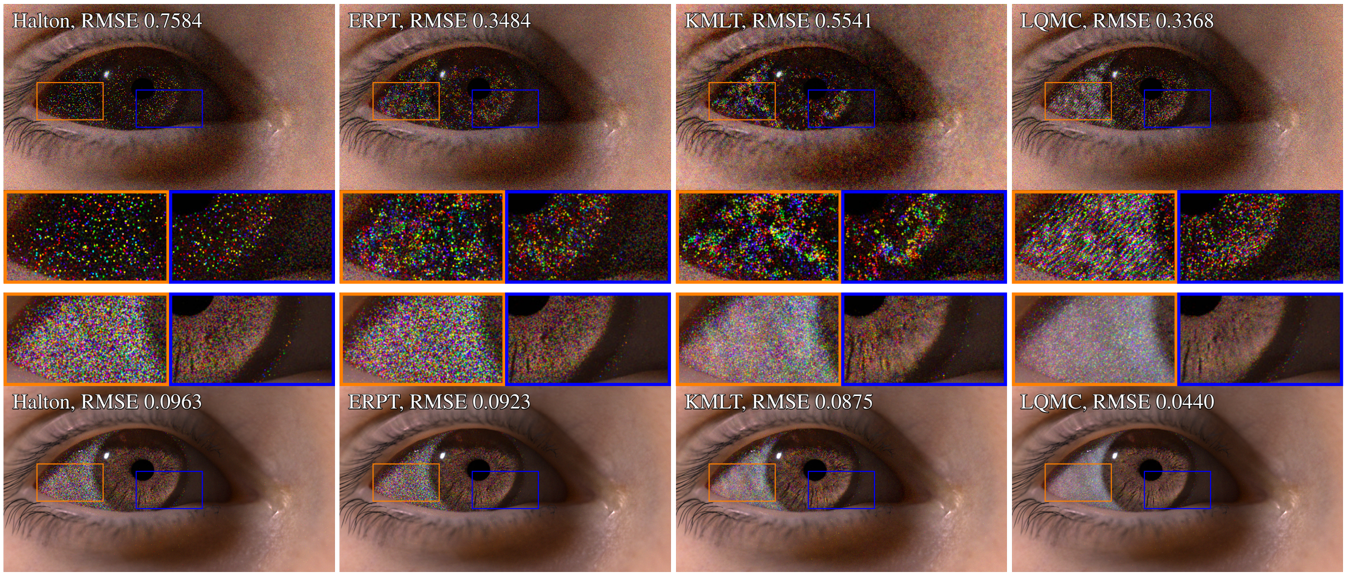

Figure 10: *An eye with difficult caustic paths and displacement on the iris. Left to right: path tracing with next event estimation, ERPT, KMLT, local quasi-Monte Carlo (LQMC, ours), all images are equal time. Top: one minute, bottom: one hour. Our stratification pays off from the start of the rendering process.*

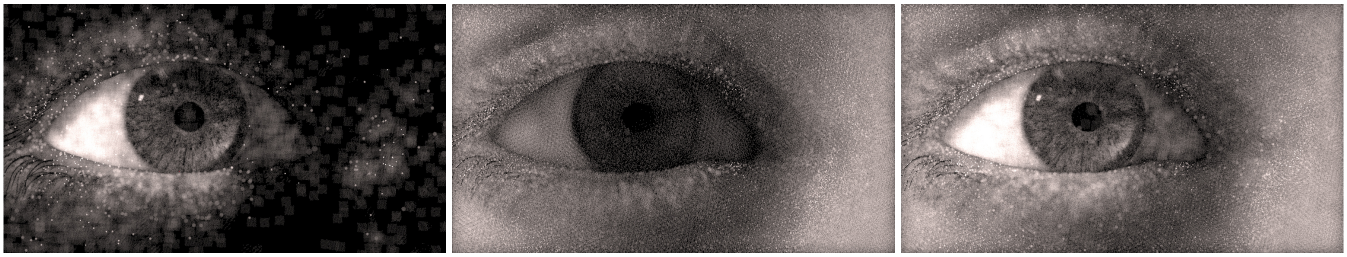

Figure 11: *LQMC sample density on the eye scene after 160 spp: it adapts to the brightness, exploring the most important regions of the image. Left to right: caustic paths, all other paths, both combined. Circular splats correspond to breakup point* $b = 0$*, square splats to* $b = 1$*, higher ones are decorrelated and have a randomised pattern. Images have been gamma-mapped for display.*

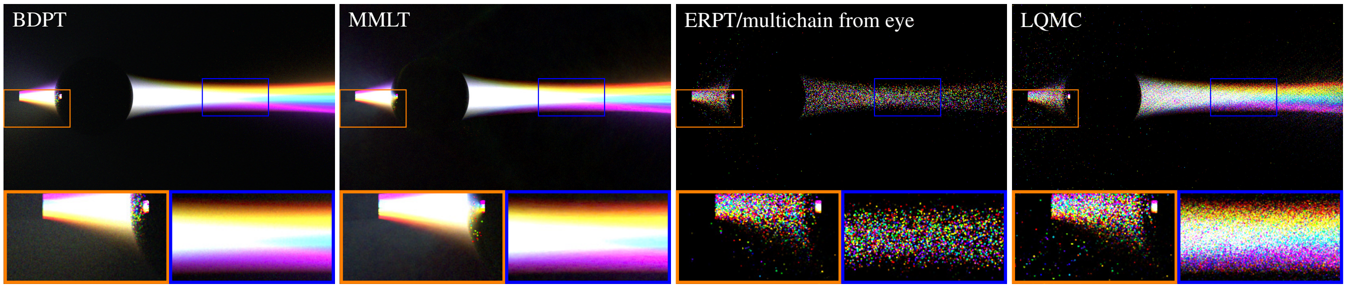

Figure 12: *In an infinite, homogeneous participating medium, a directed, coloured beam of light is refracted through a smooth dielectric sphere (equal time renders, 15 minutes). This scene is well handled by light tracing, and MIS robustly detects this for both Monte Carlo (BDPT) and Markov chain Monte Carlo (in particular, here: MMLT [HKD14]). The two other techniques here (ERPT and LQMC) use only a uni-directional technique from the eye, always tracing up to the light source. Even starting with suboptimal sampling, our technique manages to reconstruct the important features of the path space.*

5. Discussion and Future Work

Lattice structure As N increases, we start to see lattice structures in the render (see Fig. 8). A possible workaround might be to scramble the rank-1 lattice dimensions by randomly permuting the mapping from primary sample space to rank-1 lattice dimensions. It may also be possible to exploit the clear structure to guide reconstruction filters.

Trade-offs compared to Markov chains Temporal stability is governed by stratification over image plane, but the overall sampling efficiency is governed by the acceptance of samples (i.e. it depends on a large MIS weight): larger tiles stratify better, smaller tiles lead to higher weights. This is similar to Markov chain-based methods, only that small step sizes there can lead to clumping artifacts and large step sizes may lead to uncontrolled fireflies due to stuck chains. In contrast, in our setting the magnitude of the seed sample is fixed. In the worst case, when tile sizes are much too large, the seed path will stand as a single, unexplored Monte Carlo sample and some computation will be wasted.

One difference to Kelemen Metropolis is when the exploration takes place. In standard Kelemen MLT the mutation is done on the whole group of paths in case there are bifurcations (such as for bidirectional path tracing or next event estimation in combination with forward scattering). In our current implementation we perform the perturbations every time a complete path is found, i.e. each time a full contribution would be added to the pixel. This can yield better exploration results but comes with a speed impact due to the large number of these splat events. On the other hand, we don't explore paths which are classified as simple, and we don't always need to re-trace the whole path, leading to more efficiency.

Parameter choice Gaining control over the sampling process is a double-edged sword: Taking away automatic mechanisms from the Markov chain means we need to actively decide for exploration strategies, which leads to a larger set of parameters to tweak. We have many possibilities when to explore a path more, given that the classification can be done as a group decision on the set R. We can pick tile sizes, breakup points, and number of samples N based on surface interaction type (transmit vs. reflect) or based on tagged geometry, for instance. This is very flexible but it may be possible that future work will find general ways of adapting sampling in this framework, similar to what was done for KMLT [ZSK13] or, based on ray differentials, for VMLT [HKD15].

Bidirectional path construction Our current implementation requires the path prefix to be perturbed to be constructed via path tracing from the eye. Using full generic bidirectional techniques would be possible, too, by first computing the corresponding random numbers from the path using the CDF [Pan17].

Conversely, if desired, the proposed technique also works purely uni-directionally by just tracing rays from the eye. This may make implementations in production environments easier, where some tricks result in non-reciprocal BSDFs or smoothed surface roughness for bounces far away from the eye, or where texture lookups or level of detail selection depend on pixel footprints from the eye which are yet unknown when starting at the light source.

6. Conclusion

We presented a simple technique that yields considerable quality improvements over plain Monte Carlo sampling by adding in purely deterministic local exploration. Even in hard scenes it reaches similar levels of quality as the best state-of-the-art algorithms (for instance VMLT or VCM in the *ajar door* scene). At the same time, it is simple to implement and to control, and no additional memory is required. We showed how this framework subsumes previous work such as hero-wavelength sampling and path reuse for multi-frame rendering. As future work, we think lifting the restriction on the regular lattice structure will be important, as well as automatic means to adapt tile sizes and sample counts to scene features.

7. Acknowledgements

The first author's work has been funded by the HITS gGmbH.

References

[CDO11] Chen S., Dick J., Owen A. B.: Consistency of Markov chain quasi-Monte Carlo on continuous state spaces. *The Annals of Statistics 39*, 2 (04 2011), 673–701. 3

[CMNO10] Chen S., Matsumoto M., Nishimura T., Owen A.: New inputs and methods for Markov chain quasi-Monte Carlo. In *Monte Carlo and quasi-Monte Carlo Methods* (2010). 3

[CP76] Cranley R., Patterson T.: Randomization of number theoretic methods for multiple integration. *SIAM Journal of Numerical Analysis 13*, 6 (1976), 904–914. 3

[CTE05] Cline D., Talbot J., Egbert P. K.: Energy redistribution path tracing. *ACM Trans. on Graphics (Proc. SIGGRAPH) 24*, 3 (2005), 1186–1195. 3

[Dam09] Dammertz S.: *Rank-1 Lattices in Computer Graphics*. PhD thesis, Ulm University, 2009. 2, 6

[DKH*13] Dachsbacher C., Krivanek J., Hasan M., Arbree A., Walter B., Novak J.: Scalable realistic rendering with many-light methods. *Computer Graphics Forum 33*, 1 (2013), 88–104. 3

[GKDS12] Georgiev I., Krivanek J., Davidovic T., Slusallek P.: Light transport simulation with vertex connection and merging. *ACM Trans. on Graphics (Proc. SIGGRAPH Asia) 31*, 6 (2012), 192:1–192:10. 3

[GRK12] Grünschloss L., Raab M., Keller A.: Enumerating quasi-Monte Carlo point sequences in elementary intervals. In *Monte Carlo and Quasi-Monte Carlo Methods 2010* (2012), Wozniakowski H., Plaskota L., (Eds.), Springer-Verlag. 3, 6

[Has70] Hastings W. K.: Monte Carlo sampling methods using Markov chains and their applications. *Biometrika 57*, 1 (1970), 97–109. 2

[HDF15] Hanika J., Droske M., Fascione L.: Manifold next event estimation. *Computer Graphics Forum (Proc. Eurographics Symposium on Rendering) 34*, 4 (June 2015), 87–97. 3

[HKD14] Hachisuka T., Kaplanyan A. S., Dachsbacher C.: Multiplexed Metropolis light transport. *ACM Trans. on Graphics (Proc. SIGGRAPH) 33*, 4 (2014), 100:1–100:10. 2, 7, 10

[HKD15] Hanika J., Kaplanyan A., Dachsbacher C.: Improved half vector space light transport. *Computer Graphics Forum (Proceedings of Eurographics Symposium on Rendering) 34*, 4 (June 2015), 65–74. 5, 10

[HPJ12] Hachisuka T., Pantaleoni J., Jensen H. W.: A path space extension for robust light transport simulation. *ACM Trans. on Graphics (Proc. SIGGRAPH Asia) 31*, 6 (2012), 191:1–191:10. 3

[Kaj86] Kajiya J. T.: The rendering equation. *Computer Graphics (Proc. SIGGRAPH)* (1986), 143–150. 2, 5

[KFF*15] Keller A., Fascione L., Fajardo M., Christensen P., Hanika J., Eisenacher C., Nichols G.: SIGGRAPH 2015 course notes: The path-tracing revolution in the movie industry, Aug. 2015. 1

[KGH*14] Krivanek J., Georgiev I., Hachisuka T., Vevoda P., Sik M., Nowrouzezahrai D., Jarosz W.: Unifying points, beams, and paths in volumetric light transport simulation. *ACM Trans. on Graphics (Proc. SIGGRAPH) 33*, 4 (July 2014). 2

[KK02] Kollig T., Keller A.: Efficient bidirectional path tracing by randomized quasi-Monte Carlo integration. In *Monte Carlo and Quasi-Monte Carlo Methods 2000* (Berlin, 2002), K.-T. Fang F. H., Niederreiter H., (Eds.), Springer-Verlag, pp. 290–305. 3

[KMA*15] Kettunen M., Manzi M., Aittala M., Lehtinen J., Durand F., Zwicker M.: Gradient-domain path tracing. *ACM Trans. on Graphics (Proc. SIGGRAPH) 34*, 4 (2015). 3, 5

[KSKAC02] Kelemen C., Szirmay-Kalos L., Antal G., Csonka F.: A simple and robust mutation strategy for the Metropolis light transport algorithm. *Computer Graphics Forum 21*, 3 (2002), 531–540. 2

[LLW00] Liu J. S., Liang F., Wong W. H.: The multiple-try method and local optimization in Metropolis sampling. *Journal of the American Statistical Association 95*, 449 (Mar. 2000), 121–134. 3

[LW93] Lafortune E., Willems Y.: Bi-directional path tracing. In *Proc. of COMPUGRAPHICS* (1993), pp. 145–153. 2, 3

[MFSSK06] Mendez-Feliu A., Sbert M., Szirmay-Kalos L.: Reusing frames in camera animation. *Journal of the WSCG 14* (2006). 3, 4, 7

[MRR*53] Metropolis N., Rosenbluth A. W., Rosenbluth M. N., Teller A. H., Teller E.: Equation of state calculations by fast computing machines. *The Journal of Chemical Physics 21*, 6 (1953), 1087–1092. 2

[NC06] Nuyens D., Cools R.: Fast component-by-component construction of rank-1 lattice rules with a non-prime number of points. *Journal of Complexity 22*, 1 (Feb. 2006), 4–28. 3, 6

[Nie92] Niederreiter H.: *Random Number Generation and Quasi-Monte Carlo Methods*. SIAM, Philadelphia, 1992. 1, 3

[Pan17] Pantaleoni J.: Charted Metropolis light transport. *Transactions on Graphics (Proceedings of SIGGRAPH) 35*, 4 (Aug. 2017), 1–15. 6, 10

[SSA05] Stark M., Shirley P., Ashikhmin M.: Generation of stratified samples for b-spline pixel filtering. *Journal of Graphics Tools 10*, 1 (2005), 39–48. 6

[ST10] Suwa H., Todo S.: Markov chain Monte Carlo method without detailed balance. *Phys. Rev. Lett. 105*, 12 (2010). 3

[VG94] Veach E., Guibas L.: Bidirectional estimators for light transport. In *Proc. Eurographics Workshop on Rendering* (1994), pp. 147–162. 2, 3

[VG95] Veach E., Guibas L. J.: Optimally combining sampling techniques for Monte Carlo rendering. *Proc. SIGGRAPH* (1995), 419–428. 2, 4

[VG97] Veach E., Guibas L. J.: Metropolis light transport. *Proc. SIGGRAPH* (1997), 65–76. 2

[VK16] Vorba J., Krivanek J.: Adjoint-driven Russian roulette and splitting in light transport simulation. *ACM Trans. on Graphics (Proc. SIGGRAPH) 35*, 4 (2016). 2

[VKS*14] Vorba J., Karlik O., Sik M., Ritschel T., Krivanek J.: On-line learning of parametric mixture models for light transport simulation. *ACM Trans. on Graphics (Proc. SIGGRAPH) 33*, 4 (aug 2014). 2

[WND*14] Wilkie A., Nawaz S., Droske M., Weidlich A., Hanika J.: Hero wavelength spectral sampling. *Computer Graphics Forum (Proceedings of Eurographics Symposium on Rendering) 33*, 4 (July 2014), 123–131. 3, 4, 7

[ZJL*15] Zwicker M., Jarosz W., Lehtinen J., Moon B., Ramamoorthi R., Rousselle F., Sen P., Soler C., Yoon S.-E.: Recent advances in adaptive sampling and reconstruction for Monte Carlo rendering. *Computer Graphics Forum 34*, 2 (2015), 667–681. 2

[ZSK13] Zsolnai K., Szirmay-Kalos L.: Automatic parameter control for Metropolis light transport. In *Eurographics short papers* (2013), pp. 53–56. 10