EUROGRAPHICS 2002 / G. Drettakis and H.-P. Seidel
(Guest Editors)

Volume 21 (2002), Number 3

Efficient Multidimensional Sampling

Thomas Kollig and Alexander Keller

Department of Computer Science, Kaiserslautern University, Germany

Abstract

Image synthesis often requires the Monte Carlo estimation of integrals. Based on a generalized concept of stratification we present an efficient sampling scheme that consistently outperforms previous techniques. This is achieved by assembling sampling patterns that are stratified in the sense of jittered sampling and N-rooks sampling at the same time. The faster convergence and improved anti-aliasing are demonstrated by numerical experiments.

Categories and Subject Descriptors (according to ACM CCS): G.3 [Probability and Statistics]: Probabilistic Algorithms (including Monte Carlo); I.3.2 [Computer Graphics]: Picture/Image Generation; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism.

1. Introduction

Many rendering tasks are given in integral form and usually the integrands are discontinuous and of high dimension, too. Since the Monte Carlo method[22] is independent of dimension and applicable to all square-integrable functions, it has proven to be a practical tool for numerical integration. It relies on the point sampling paradigm and such on sample placement. Increasing the uniformity of the samples is crucial for the efficiency of the stochastic method and the level of noise contained in the rendered images.

The most popular uniform sampling schemes in graphics are jittered and Latin hypercube sampling. Jittered sampling[2] profoundly has been analyzed by Mitchell[13] and in fact can only improve efficiency. Chiu et al.[1] joined the concepts of jittered and Latin hypercube sampling obtaining an increased uniformity of the samples, but no minimum distance property can be guaranteed that has been proved to be useful in graphics[2]. In consequence care of the choice of the strata has to be taken manually, since warping[19] these point sets in order to e.g. sample long thin light sources can dramatically reduce the benefits of stratification.

We present an unbiased Monte Carlo integration scheme that consistently outperforms the previous approaches, is trivial to implement, and robust to use even with warping. This is obtained by an even more general concept of stratification than just joining jittered and Latin hypercube sampling. Since our samples are highly correlated and satisfy a minimum distance property, noise artifacts are attenuated much more efficiently and anti-aliasing is improved.

2. Monte Carlo Integration

The Monte Carlo method of integration estimates the integral of a square-integrable function f over the s-dimensional unit cube by

$$\int_{[0,1)^s} f(x)dx \approx \frac{1}{N}\sum_{i=0}^{N-1} f(\xi_i) \ , \qquad (1)$$

where the $\xi_i \in [0,1)^s$ are independent uniform random samples. The efficiency of the stochastic method is inversely proportional to the variance σ^2_{MC} of the estimator (1). Among many variance reduction techniques[22, 23, 11], increasing the uniformity of the samples by stratification has been proven to be beneficial in graphics[13, 2]. We briefly review the facts relevant to this paper; for a more complete survey we refer to e.g. Glassner's book[6] or the notes[9] of the course 'Beyond Monte Carlo'.

2.1. Jittered Sampling

For jittered sampling[2] the unit cube is subdivided into N cubes of equal measure $\frac{1}{N}$, where in each cube one

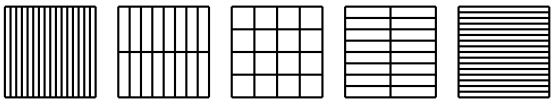

Figure 1: *All elementary intervals in base $b = 2$ and dimension $s = 2$ with volume $\lambda_2(E) = \frac{1}{16}$.*

random sample is taken (see figure 2 (a)). It is simple to show[7] that the variance of the resulting estimator never can be higher than σ^2_{MC}.

2.2. Latin Hypercube Sampling

The idea of Latin hypercube sampling (N-rooks sampling) is to subdivide the unit cube into N intervals along each coordinate. Then the samples are chosen randomly such that each interval contains exactly one point (see figure 2 (c)). Since there are more restrictions in the placement of Latin hypercube samples in comparison to jittered sampling, the variance

$$\sigma^2_{\mathrm{LHS}} \leq \left(\frac{N}{N-1}\right)^{\min\{s-1,1\}} \cdot \sigma^2_{\mathrm{MC}}$$

can slightly increase. Nevertheless it never can be much higher and often is reduced in practical application.

3. Uniform Samples from (t, m, s)-Nets

Chiu et al.[1] combined jittered and Latin hypercube sampling in order to achieve more uniformity. An even more general concept of stratification has been developed by Sobol'[21] that finally yielded the so-called (t, m, s)-nets and (t, s)-sequences[14].

In order to explain the concept, the notion of the elementary interval

$$E := \prod_{j=1}^{s} \left[\frac{a_j}{b^{l_j}}, \frac{a_j+1}{b^{l_j}}\right) \subseteq [0,1)^s$$

is required, where $0 \leq a_j < b^{l_j}$ and $0 \leq l_j$ are integers. Consequently the volume of E is

$$\lambda_s(E) = \prod_{j=1}^{s} \frac{1}{b^{l_j}} = b^{-\sum_{j=1}^{s} l_j} \quad .$$

As an example figure 1 shows the structure of all elementary intervals with the volume $\lambda_2(E) = \frac{1}{16}$ in base $b = 2$ for dimension $s = 2$.

Given two integers $0 \leq t \leq m$ a set of $N = b^m$ s-dimensional points x_i is called a (t, m, s)-net in base b if every elementary interval with volume $\lambda_s(E) = b^{t-m}$ contains exactly b^t points.

t can be considered as a quality parameter that is best if chosen small. For $t = 0$ each elementary interval contains exactly $b^0 = 1$ point. Consequently the b^{ks} points of a $(0, ks, s)$-net in base b with $k \in \mathbb{N}$ are stratified like both jittered and Latin hypercube sampling points at the same time as can be seen in figure 2 (b). In addition the structure of the elementary intervals imposes even more stratification resulting in an increased uniformity of the samples.

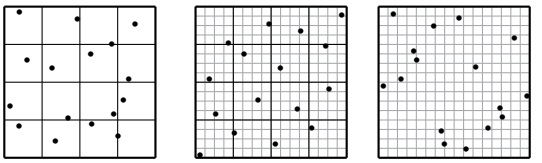

Figure 2: *Realization of (a) jittered and (c) Latin hypercube sampling. The realization of a $(0, 4, 2)$-net in base 2 in (b) not only combines both sampling techniques, but imposes even more stratification as can be seen from the corresponding dyadic elementary intervals in figure 1.*

In the sequel we explain how to efficiently construct such point sets suited for unbiased Monte Carlo integration.

3.1. Deterministic Generation

(t, m, s)-nets are much more uniformly distributed than random samples can be. This is exploited by quasi-Monte Carlo integration[15], where deterministic (t, m, s)-nets are used for the estimator (1): For certain, very restricted function classes a quadratically faster convergence can be guaranteed as compared to random sampling.

Most deterministic constructions of (t, m, s)-nets are based on (t, s)-sequences: For an integer $t \geq 0$ an infinite point sequence $(y_i)_{i=0}^{\infty}$ is called a (t, s)-sequence in base b, if for all $k \geq 0$ and $m > t$ the point set $\{y_{kb^m}, \ldots, y_{(k+1)b^m-1}\}$ is a (t, m, s)-net.

Consequently the first b^m points of a (t, s)-sequence form a (t, m, s)-net. A second approach is to add the component $\frac{i}{b^m}$ to the first b^m points of a (t, s)-sequence always yielding a $(t, m, s + 1)$-net.

Since explaining explicit constructions is beyond the scope of this paper, we refer to Niederreiter's book[15] and provide the compact implementation (section 7) of three $(0, 1)$-sequences that can be used to generate a $(0, 2)$-sequence and $(0, m, 2)$-nets.

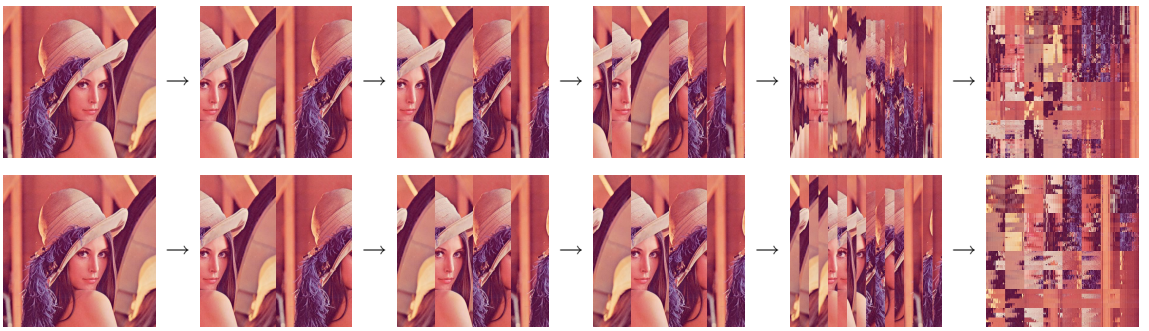

Figure 5: *Owen scrambling (top row) and random digit scrambling (bottom row) in base 2. A difference is hardly perceivable. First intervals are swapped horizontally; the final image then includes the permutations along the vertical direction, too.*

Figure 3: *The effect of a Cranley-Patterson rotation by the random vector ξ.*

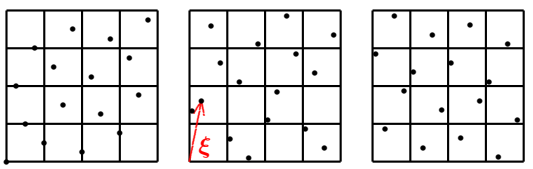

Figure 4: *Randomizing the $(0, 4, 2)$-net in base 2 in a) by a Cranley-Patterson rotation can degrade the uniformity as shown in b), whereas c) random digit scrambling preserves the properties of the net.*

3.2. Randomized Generation

The quasi-Monte Carlo method yields consistent but biased estimators. However, it is possible to randomize a (t, m, s)-net $P := \{a_0, a_1, \ldots, a_{N-1}\}$ in such a way that

a) the randomized point set $X := \{x_0, x_1, \ldots, x_{N-1}\}$ remains a (t, m, s)-net (with probability 1) and
b) x_i is uniformly distributed in $[0, 1)^s$ for $i = 0, 1, \ldots, N-1$.

Condition b) is sufficient to make (1) an unbiased estimator for all square-integrable functions[16, 8]. Preserving the uniformity properties of the samples by condition a) allows one to benefit from the improved convergence of the quasi-Monte Carlo method. The resulting variance reduction technique belongs to the domain of randomized quasi-Monte Carlo integration[18, 10].

3.2.1. Cranley-Patterson Rotations

Cranley and Patterson[3] randomized a point set P by just adding the same random shift ξ to each point $a_i \in P$ modulo 1 as illustrated in figure 3. Originally developed for point sets that tile periodically, applying a so-called Cranley-Patterson rotation to a (t, m, s)-net can destroy its stratification structure (see figure 4) thus violating condition a).

3.2.2. Owen Scrambling

Owen's randomization scheme preserves the structure of (t, m, s)-nets in base b (with probability 1). For the (involved) formulas we refer to the original work[16]. The actual algorithm, however, is simple to explain. Starting with $H = [0, 1)^s$ the following steps are applied to each coordinate (see figure 5):

1. Slice H into b equal volumes $H_1, H_2, \ldots, H_b$ along the coordinate.
2. Randomly permute these volumes in an independent way.
3. For each volume H_h recursively repeat the procedure starting out with $H = H_h$.

Owen[17] proved that using an Owen-scrambled $(0, m, s)$-net in (1) yields the upper bound

$$\sigma^2_{\mathrm{OS}} \leq \left(\frac{b}{b-1}\right)^{\min\{s-1,m\}} \cdot \sigma^2_{\mathrm{MC}}$$

for the variance σ^2_{OS} of the resulting estimator. For $b = N$ this $(0, m, s)$-net sampling degenerates to Latin hypercube sampling. Decreasing the base b implies more restrictions to the sample placement resulting in an increased variance bound. Although this variance bound is strict[17], for most functions to be integrated the variance is reduced.

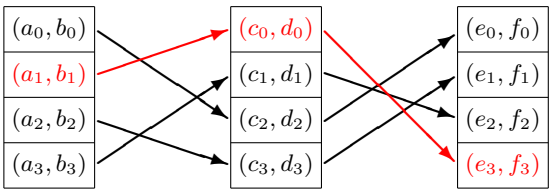

Figure 6: *Multidimensional sampling. The highlighted sample $(a_1, b_1, c_0, d_0, e_3, f_3)$ is padded from the stratified patterns (a_i, b_i),(c_i, d_i), and (e_i, f_i) using random permutations.*

Due to the finite precision of computer arithmetic the infinite scheme in fact becomes a finite algorithm. Nevertheless the number of required random permutations behaves exponentially in the precision so that an efficient implementation remains quite challenging[5].

3.2.3. Random Digit Scrambling

Instead of using independent random permutations in each level of the recursion of Owen scrambling, only one random permutation can be used (see the bottom row of figure 5). This subset of the original method obviously still fulfills the conditions of section 3.2, but requires only a number of permutations linear in the precision. Opposite to Owen's scrambling method, using random digit scrambling preserves minimum distance properties contained in the net to be scrambled.

A highly efficient implementation becomes available for (t, m, s)-nets in base $b = 2$, where a permutation simply can be realized by the `XOR` operation[4, 5]: Each coordinate of the point set is randomized by just performing a bitwise `XOR` of one random bit vector (i.e. a random integer) and the components of the point set (for the trivial realization see section 7).

4. Multidimensional Sampling

Typically the integrands in image synthesis expose high correlation with respect to certain low-dimensional projections, e.g. the pixel area, lens area, or area light sources. Therefore high-dimensional samples are padded using low-dimensional stratified patterns[20]. Correlation artifacts are avoided by randomly permuting the sample order of the low-dimensional patterns (see figure 6). Additionally the number of samples becomes independent of dimension making this approach more practical than jittered sampling.

Although constructions of (t, m, s)-nets exist for any dimension, choosing the optimal quality parameter $t = 0$ requires $b \geq s - 1$ for $m \geq 2$. For $s > 3$ this prohibits to use the extraordinarily efficient vectorized implementations in base $b = 2$. However, using the simple algorithms from section 7, it is possible to pad high-dimensional samples in an even simpler way: Instead of using random permutations we just pad independent realizations[10] of randomly digit scrambled nets (or Owen-scrambled nets). Since condition (2) (section 3.2) holds for the low-dimensional realizations, each resulting high-dimensional sample x_i is uniformly distributed in $[0, 1)^s$ for $i = 0, 1, \ldots, N - 1$, too, guaranteeing an unbiased estimate (1).

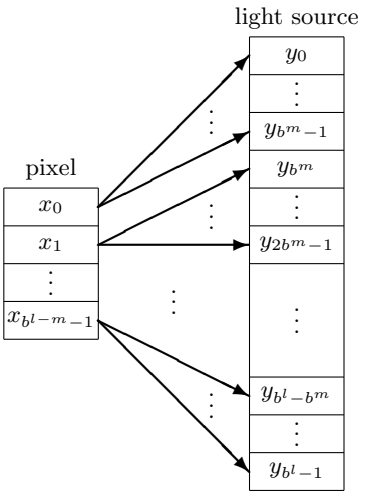

Figure 7: *Trajectory splitting, see the explanation in section 4.1.*

4.1. Trajectory Splitting

Considering the example of distribution ray tracing[2] splitting trajectories[8], e.g. tracing multiple shadow rays for one eye ray, can increase efficiency depending on the correlation coefficient with respect to the split dimensions[22].

From the definition in section 3.1 it follows that the first b^l points of a (t, s)-sequence $(y_j)_{j=0}^{\infty}$ are a (t, l, s)-net. In addition each point set $\{y_{ib^m}, \ldots, y_{(i+1)b^m-1}\}$ is a (t, m, s)-net for $0 \leq i < b^{l-m}$. This observation can be used to realize trajectory splitting by extending the scheme from the previous section:

For the example of pixel anti-aliasing and illumination by an area light source two independent realizations are required: An instance of a randomized $(0, l - m, 2)$-net of b^{l-m} samples x_i in the pixel and the first $b^l = b^{l-m} \cdot b^m$ samples y_j of an instance of a randomized $(0, 2)$-sequence on the area light source. For the i-th sample in the pixel then b^m shadow rays have to be traced towards the samples

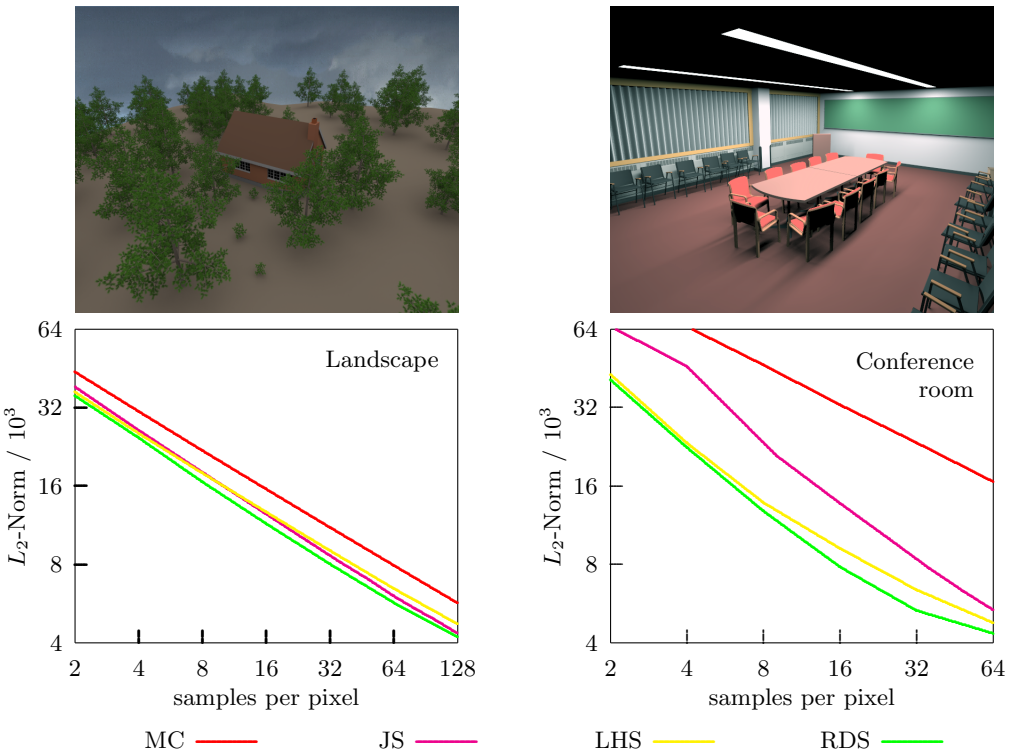

Figure 8: *Comparison of pure random (MC), jittered (JS), and Latin hypercube (LHS) sampling with our approach using random digit scrambling (RDS).*

$\{y_{ib^m}, \ldots, y_{(i+1)b^m-1}\}$ on the light source (see figure 7) yielding the estimator

$$\int_{[0,1)^2}\int_{[0,1)^2} f(x,y)dydx \approx \frac{1}{b^{l-m}}\sum_{i=0}^{b^{l-m}-1}\frac{1}{b^m}\sum_{j=ib^m}^{(i+1)b^m-1} f(x_i,y_j)\ . \quad (2)$$

By using the subsequent $(0,m,2)$-nets of a $(0,2)$-sequence to realize trajectory splitting, the samples on the light source itself form a $(0,l,2)$-net obtaining superior stratification properties in a fully automatic way. This would be rather costly to achieve by jittered or Latin hypercube sampling.

5. Numerical Results

For the application examples two representative settings were selected: An overcast sky model daylight simulation and an indoor scene with very long and thin light sources. The resulting four-dimensional integrals compute pixel anti-aliasing with direct illumination.

The new scheme (2) with x_i and y_i from the algorithms in section 7 is compared to pure random, jittered, and Latin hypercube sampling. In the experiments a splitting rate of 4 was used, i.e. for each eye ray 4 shadow rays were traced. For each pixel an independent realization of the sampling scheme was used.

Trajectory splitting for jittered and Latin hypercube sampling was realized by generalizing the multidimensional sampling scheme[20] in a straightforward way: N samples and $4N$ samples were generated on the pixel and the light source, respectively. Then the set of $4N$ points randomly is split into N sets of 4 points and each set is assigned a pixel sample in canonical order.

The error graphs in figure 8 are determined by computing the L_2-norm of a measurement to a converged master image. For the case of the hemispherical overcast sky integral our scheme slightly outperforms jittered and Latin hypercube sampling, is much simpler to implement, and saves about 10–15% of the total number of rays to be traced in order to obtain the same quality. Due to the complex shadowing the overall gain by stratification is small.

Warping the samples onto the long thin light sources in the conferences room scene exposes the projection regularity of the samples. Therefore Latin hypercube

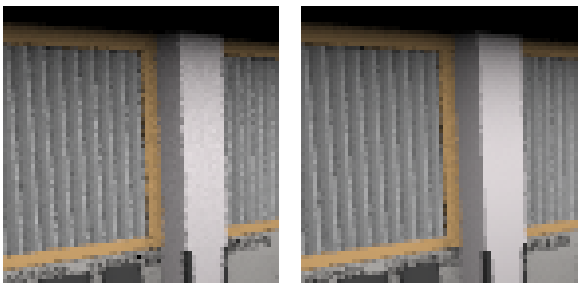

Figure 9: *Improved anti-aliasing and noise reduction for the conference room scene with long light sources. Latin hypercube sampling in the left image and our new sampling scheme on the right.*

sampling significantly outperforms jittered sampling. The samples from the new scheme, however, are stratified in a more general way and satisfy a minimum distance property reducing the error by approximately 15% as compared to Latin hypercube sampling.

Comparing the zoomed images in figure 9 shows that the high correlation of the samples from the new scheme results in superior anti-aliasing and noise reduction as compared to Latin hypercube sampling. This becomes even more apparent in animations, where uncorrelated noise causes distracting flicker.

6. Conclusion

We presented new algorithms for efficiently generating high-dimensional uniform samples yielding unbiased Monte Carlo estimators. The implementation of the highly correlated sampling scheme is extremely simple and due to the generalized concept of stratification previous patterns are outperformed consistently.

7. Appendix: Algorithms

Using the following code fragments it is possible to verify the results of the paper with any ray tracer in a very short amount of time. The routines `RI_vdC`, `RI_S`, and `RI_LP` implement the radical inverse functions by van der Corput[15], Sobol'[21], and Larcher and Pillichshammer[12], respectively, which are $(0,1)$-sequences in base $b = 2$ (see section 3.1). Randomized digit scrambling (section 3.2.3) is realized by just calling the routines with a random integer instead of the default parameter `uint r = 0`. Completing `RI_vdC` with the component $\frac{i}{2^m}$ yields the famous Hammersley point set, which in fact is a $(0, m, 2)$-net. Using $x_i = \left(\frac{i}{2^m}, \texttt{RI\_LP}(i)\right)$ instead, however, results in a $(0, m, 2)$-net of much higher quality. Combining $y_i = (\texttt{RI\_vdC}(i), \texttt{RI\_S}(i))$ results in the first two components of the Sobol' sequence, which form a $(0, 2)$-sequence as used in section 4.1.

```
typedef unsigned int uint;

double RI_vdC(uint bits, uint r = 0)
{
  bits = ( bits                << 16)
         | ( bits                >> 16);
  bits = ((bits & 0x00ff00ff) <<  8)
         | ((bits & 0xff00ff00) >>  8);
  bits = ((bits & 0x0f0f0f0f) <<  4)
         | ((bits & 0xf0f0f0f0) >>  4);
  bits = ((bits & 0x33333333) <<  2)
         | ((bits & 0xcccccccc) >>  2);
  bits = ((bits & 0x55555555) <<  1)
         | ((bits & 0xaaaaaaaa) >>  1);

  bits ^= r;

  return (double) bits / (double) 0x100000000LL;
}

double RI_S(uint i, uint r = 0)
{
  for(uint v = 1<<31; i; i >>= 1, v ^= v>>1)
    if(i & 1)
      r ^= v;

  return (double) r / (double) 0x100000000LL;
}

double RI_LP(uint i, uint r = 0)
{
  for(uint v = 1<<31; i; i >>= 1, v |= v>>1)
    if(i & 1)
      r ^= v;

  return (double) r / (double) 0x100000000LL;
}
```

Acknowledgements

The first author has been funded by the Stiftung Rheinland-Pfalz für Innovation.

References

1. K. Chiu, C. Wang, and P. Shirley. Multi-Jittered Sampling. In S. Heckbert, editor, *Graphics Gems IV*, pages 370–374. Academic Press, 1994. 1, 2
2. R. Cook, T. Porter, and L. Carpenter. Distributed Ray Tracing. In *Computer Graphics (SIGGRAPH 84 Conference Proceedings)*, pages 137–145, 1984. 1, 4
3. R. Cranley and T. Patterson. Randomization of number theoretic methods for multiple integration. *SIAM Journal on Numerical Analysis*, 13:904–914, 1976. 3

4. I. Friedel. Abtastmethoden für die Monte-Carlo und Quasi-Monte-Carlo-Integration. Universität Kaiserslautern, 1998. 4

5. I. Friedel and A. Keller. Fast generation of randomized low-discrepancy point sets. In H. Niederreiter, K. Fang, and F. Hickernell, editors, *Monte Carlo and Quasi-Monte Carlo Methods 2000*, pages 257–273. Springer, 2002. 4

6. A. Glassner. *Principles of Digital Image Synthesis*. Morgan Kaufmann, 1995. 1

7. M. Kalos and P. Whitlock. *Monte Carlo Methods, Volume I: Basics*. J. Wiley & Sons, 1986. 2

8. A. Keller. Trajectory Splitting by Restricted Replication. Interner Bericht 316/01, Universität Kaiserslautern, 2001. 3, 4

9. A. Keller. Beyond Monte Carlo – Course Material. Interner Bericht 320/02, University of Kaiserslautern, 2002. Lecture at the Caltech, July 30th – August 3rd, 2001. 1

10. T. Kollig and A. Keller. Efficient bidirectional path tracing by randomized quasi-Monte Carlo integration. In H. Niederreiter, K. Fang, and F. Hickernell, editors, *Monte Carlo and Quasi-Monte Carlo Methods 2000*, pages 290–305. Springer, 2002. 3, 4

11. E. Lafortune. *Mathematical Models and Monte Carlo Algorithms for Physically Based Rendering*. PhD thesis, Katholieke Universitiet Leuven, Belgium, 1996. 1

12. G. Larcher and F. Pillichshammer. Walsh Series Analysis of the L_2-Discrepancy of Symmetrisized Point Sets. *Monatsh. Math.*, (132):1–18, 2001. 6

13. D. Mitchell. Consequences of Stratified Sampling in Graphics. In *SIGGRAPH 96 Conference Proceedings*, Annual Conference Series, pages 277–280, 1996. 1

14. H. Niederreiter. Point sets and sequences with small discrepancy. *Monatsh. Math.*, 104:273–337, 1987. 2

15. H. Niederreiter. *Random Number Generation and Quasi-Monte Carlo Methods*. SIAM, Pennsylvania, 1992. 2, 6

16. A. Owen. Randomly Permuted (t, m, s)-Nets and (t, s)-Sequences. In H. Niederreiter and P. Shiue, editors, *Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing*, volume 106 of *Lecture Notes in Statistics*, pages 299–315. Springer, 1995. 3

17. A. Owen. Monte Carlo Variance of Scrambled Net Quadrature. *SIAM J. on Numerical Analysis*, 34(5):1884–1910, 1997. 3

18. A. Owen. Monte Carlo Extension of Quasi-Monte Carlo. In *Winter Simulation Conference*, pages 571–577. IEEE Press, 1998. 3

19. P. Shirley. Nonuniform Random Point Sets via Warping. In D. Kirk, editor, *Graphics Gems III*, pages 80–83. Academic Press Professional, 1992. 1

20. P. Shirley. *Realistic Ray Tracing*. AK Peters, Ltd., 2000. 4, 5

21. I. Sobol'. On the Distribution of Points in a Cube and the approximate Evaluation of Integrals. *Zh. vychisl. Mat. mat. Fiz.*, 7(4):784–802, 1967. 2, 6

22. I. Sobol'. *A Primer for the Monte Carlo Method*. CRC Press, 1994. 1, 4

23. E. Veach. *Robust Monte Carlo Methods for Light Transport Simulation*. PhD thesis, Stanford University, 1997. 1