 https://doi.org/10.3311/PPee.40410

Periodica Polytechnica Electrical Engineering and Computer Science, 69(3), pp. 334–345, 2025

Vector Coupled Map Lattice PRNG for Monte Carlo Rendering

Attila Kárpáti[1*], Viktória Kárpáti[1], László Szécsi[1]

[1] Department of Control Engineering and Information Technology, Faculty of Electrical Engineering and Informatics, Budapest University of Technology and Economics, Műegyetem rkp. 3., H-1111 Budapest, Hungary
[*] Corresponding author, e-mail: karpati@iit.bme.hu

Received: 11 March 2025, Accepted: 07 June 2025, Published online: 25 June 2025

Abstract

In this paper we propose a straightforward method to generate random points uniformly distributed on the unit sphere or following a 3D Gaussian distribution. For that, we use a small Coupled Map Lattice (CML), which is similar to a cellular automaton but with cells containing arbitrary variables in place of states from a finite set. Our lattice variables are 3D unit vectors. We use this setup to solve the otherwise challenging task of generating uniformly distributed direction vectors on the unit sphere without resorting to rejection sampling. We also generate samples of a 3D Gaussian distribution with sufficient accuracy by summing several of the above random vectors. To showcase the possible uses of this method, we introduce a new Bidirectional Reflection Distribution Function (BRDF) model that is physically plausible and features: perfect importance sampling, only needing a few intuitive parameters, not rejecting samples, and supporting anisotropy. The sampling process is generalized by projecting 3D Gaussian samples to 2D direction space. The resulting probability density function over directions is obtained in a closed form. We also demonstrate the capabilities of our lattice Pseudo-Random Number Generators (PRNG) by creating an especially fast Lambertian path tracer and a volumetric scattering effect.

Keywords

BRDF, PRNG, ray tracing

1 Introduction

Pseudo-Random Number Generators (PRNGs), are ubiquitously used in science and engineering, with different fields posing varied requirements on the randomness qualities of the generated number sequences. These requirements include [1] that the numbers: have *k*-uniform distribution, are uncorrelated, have a long period before repeating, satisfy statistical tests of randomness, can be changed by choosing an initial seed value, can be generated rapidly and use limited computer memory.

In the case of Monte Carlo methods, suitable random numbers are essential, as k-uniform distribution is crucial to avoid bias. Low correlation, i.e., the independence of consecutive random numbers from each other is especially important when dealing with higher-dimensional integrals. Using correlated values for different sample dimensions results in a non-uniform sampling of the multi-dimensional integration domain, introducing bias again. Rapid generation is often the least concern, as the evaluation of light path contributions via ray casting is always more expensive. However, the memory footprint of the method can be a serious bottleneck when using GPU hardware, as processor-local memory is a very limited resource. If GPU threads use more local memory, less of them can be executed on a multiprocessor concurrently, reducing occupancy and thus, performance. Also, a long period may not be important when a PRNG is executed on a GPU thread, building a random walk from only a handful of randoms generated from an externally randomized seed. Meeting other randomness statistics is, in some cases, even considered detrimental. Deterministic sequences with low discrepancy (e.g., the Halton or Hammersley sequences) are used in what are called Quasi-Monte Carlo methods, obtaining better stratification and thus, less error in the estimates. These sequences are certainly not random in a statistical sense. We should note that multiple sequences with different prime bases must be used to provide sample coordinates in different dimensions, which may become cumbersome when the dimension count gets high.

PRNGs are most often constructed to produce a uniform distribution. Other desired distributions may be obtained using transformations. However, this is not always a trivial task, as exemplified by the plethora of methods devised to produce Gaussian random numbers [2].

Cite this article as: Kárpáti, A., Kárpáti, V., Szécsi, L. "Vector Coupled Map Lattice PRNG for Monte Carlo Rendering", Periodica Polytechnica Electrical Engineering and Computer Science, 69(3), pp. 334–345, 2025. https://doi.org/10.3311/PPee.40410

In formulating Monte Carlo random walk image synthesis methods, the rendering equation can be written as:

$$L(x,\omega_v) = \int_\Omega L(x,\omega_l)\cos\theta_l f_r(\omega_l,x,\omega_v)\,d\omega_l, \quad (1)$$

where $L(x, \omega_v)$ is the radiance function at position x and direction ω_v, x is the shaded surface point, ω_v is the outgoing direction for the computed outgoing radiance, ω_l is the integration variable, i.e., the incoming light direction, over Ω on the hemisphere above the shaded surface, θ_l is the angle between the incoming light direction and the surface normal and f_r is the Bidirectional Reflection Distribution Function (BRDF) that describes the reflection properties of the surface, giving the outgoing radiance in response to unit irradiance.

This integral should be evaluated using the Monte Carlo technique, that is, generating random samples according to a probability density p, and averaging sampled integrand values divided by the corresponding value of p. The estimator is going to have low variance if p is mimicking the integrand, which is the principle of importance sampling. In case of direct lighting, the sampling can be performed most straightforwardly according to the emitted radiances, meaning that the BRDF has to be evaluated for a pair of incoming and outgoing directions. For indirect lighting, however, importance sampling can generally only be performed according to the BRDF. Therefore, BRDFs are required for which the samples can be generated perfectly or almost perfectly following its distribution. Meeting these two requirements at the same time (the BRDF can be efficiently evaluated and also sampled with perfect importance sampling) poses stringent criteria on possible BRDF models.

Physically based rendering requires BRDFs that obey physical laws. These are positivity, energy conservation, and reciprocity. Out of these, reciprocity, formally the requirement that satisfies:

$$f_r(\omega_l,\omega_v) = f_r(\omega_v,\omega_l), \quad (2)$$

is quite challenging to meet. This is because reciprocity implies that the directional sampling process must rely on a random variable that is independent of the outgoing direction. Often not all values of the random variable can be mapped to an incoming direction, as area-preserving mappings between spherical regions can be costly to compute [3], especially so when BRDF anisotropy requires scaling spherical circles into spherical ellipses. Therefore, some samples must be rejected. This has to be accounted for in the estimator's generation probability factor. However, the evaluation of the rejection probability can lead to another costly elliptic integral.

In this research, we provide an algorithm that directly produces random samples of desired distributions with the following characteristics:

- generates independent samples matching a desired distribution,
- starts from an externally randomized initial state,
- runs in deterministic constant time (without rejections),
- has a low memory footprint,
- can be implemented simply and efficiently on any GPU.

In addition, we illustrate the potential applications of this method in three distinct cases: superior to state-of-the-art Lambertian Monte Carlo sample generation on the GPU, subsurface scattering simulation, and a novel BRDF model that facilitates the use of our PRNG and aims to achieve the following properties:

- cheap to evaluate,
- perfect importance sampling,
- rejected samples,
- physically plausible,
- has only a few, intuitive parameters,
- supports anisotropy.

2 Previous work

2.1 Generation of normally distributed random numbers

The generation of random numbers with a one-dimensional Gaussian distribution is already a challenging task addressed by numerous solutions, categorized by Thomas et al. [2] as CDF inversion techniques, transformation methods, rejection methods, sum-of-uniforms methods and the recursive method.

CDF inversion and transformation methods are mostly overlapping. In both cases, uniformly distributed samples need to be transformed into a normal distribution. As the CDF of the Gaussian cannot be written in terms of elementary functions, it is not straightforward to evaluate within the constraints PRNGs operate under and certainly unfeasible in GPU shaders.

The Box-Muller transform, however, exploits the fact that the radial CDF of a two-dimensional normal distribution can indeed be written in a closed form, providing a way to transform two uniformly distributed random numbers ξ_0 and ξ_1 into a 2D Gaussian. In polar coordinates, the equation is:

$$r = \sqrt{-2\ln\xi_0},\ \ \phi = 2\pi\xi_1. \quad (3)$$

As the Gaussian is separable, this provides two independent random variables with a normal distribution:

$$\gamma_0 = r\cos\phi,\ \ \gamma_1 = r\sin\phi. \tag{4}$$

This method is exact and simple, and objections against it can only be raised when the tail of the Gaussian beyond five standard deviations must be accurately sampled and number representation issues become relevant. However, it still needs good quality uniform randoms, that may themselves be challenging or costly to obtain, e.g., in a GPU thread. Rejection methods are a poor fit for parallel single-instruction-multiple-threads architectures, where threads that have finished iteration must wait for those still working to find an acceptable sample. Most widespread and powerful algorithms investigated by Thomas et al. [2] rely on dynamic flow control of this kind.

The sum-of-uniforms methods are also known as the Central Limit Theorem (CLT) approach, as the theorem implies that the distribution of the average of multiple random variables converges to the Gaussian as the number of variables increases. Convergence is especially fast for uniformly distributed variables (the Irwin–Hall distribution), at least initially. However, even with thousands of variables summed, error does not completely vanish, and CLT methods are discarded as only approximate while also expensive. Notably, the generated numbers always fall in a finite range, between the sum of the minima and maxima of the uniform distribution ranges, and thus sampling of the tails of the normal distribution is always lacking. Still, the CLT method works where accuracy requirements are not especially strict. The presence of low-precision floating-point arithmetic can be such a qualifying factor. There are some ancient tricks [4] to further reduce the error by transforming the generated samples.

The Wallace [5] method is referred to as recursive, as it relies on a large set of already normally distributed samples and combines them in random permutations to obtain new ones. The quality of samples is not comparable to more refined methods, but it is extremely fast and well-suited to parallel architectures. However, the method does not scale down to work with a small memory footprint, which would be used when generating samples in thread-local or group-local memory in GPUs. Indeed, the Wallace method relies on perturbations over the entire pool of samples, accessing random locations in global memory.

2.2 Cellular automata as PRNGs

Cellular automata were shown to be useful as PRNGs, producing results with good randomness qualities [6]. When using non-commutative algebra in PRNGs, cellular automata over quaternions were identified as the most promising candidates for application in cryptography [7].

The PRNG concept proposed in this paper is inspired by Quaternionic Cellular Automata (QCA) [8]. QCA generators load m cells with initial values (the seed) $x_1^0, \ldots, x_m^0 \in H_{Z_m} \setminus 0$, meaning that the quaternion coefficients are integers. This represents stage zero. To compute stage $n + 1$ from stage n, one can use the rule:

$$x_s^{n+1} = x_{s-1}^n \cdot x_{s+1}^n, \tag{5}$$

where $s \in [0, z-1]$ is the cell index out of z cells. The arithmetic on the subscripts takes place in Z_m. The output at each stage is the element of a fixed but arbitrary cell. Cell count m can be as low as 5, and a PRNG with good quality metrics can be obtained [7], when the quaternions are in H_{Z_m}, where all coefficients are modulo m integers.

2.3 Coupled Map Lattice

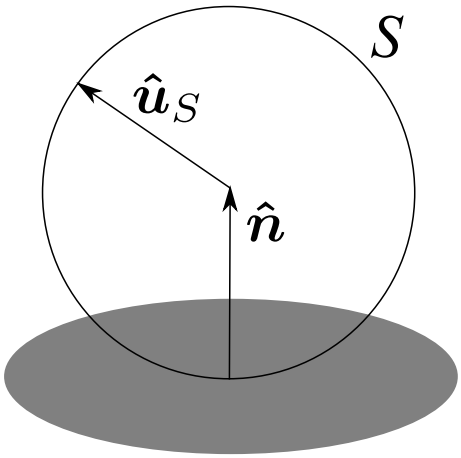
A Coupled Map Lattice (CML) is different from a Cellular Automaton in that its cells (or rather sites) may contain arbitrary variables in place of states from a finite set. CMLs can be used to model non-linear systems and often exhibit chaotic dynamics [9].

2.4 BRDF

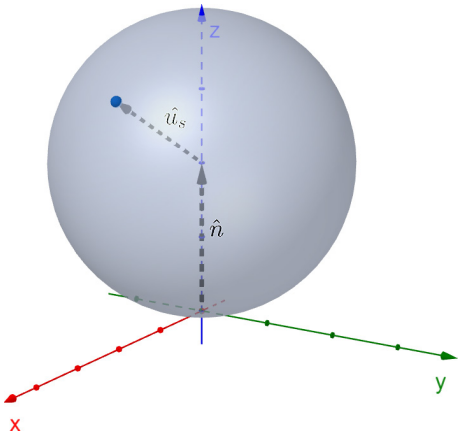
Over decades of computer graphics research, numerous BRDF models and representations have been proposed for designing and capturing material surface reflectance properties and rendering scenes by evaluating the resulting functions. The most basic out of those, the diffuse Lambertian, is not only very simple and efficient to evaluate but also lends itself to simple and efficient importance sampling. As the BRDF itself is constant, only the cosine term needs to be considered. This can be done by transforming a pair of uniformly distributed random numbers onto the directional hemisphere.

For our work, the sampling scheme proposed by Tobler et al. [10] is more relevant. Instead of generating cosine-distributed points on the directional hemisphere, samples are taken with a uniform distribution on a unit radius sphere sharing a tangent plane with the surface at the shaded point (Fig. 1).

We will continue to refer to this sphere as the *tangent sphere S*. When these points are centrally projected through the surface point to a double-radius hemisphere, we obtain cosine-distributed directional samples. This allows for extremely elegant implementation if samples $\boldsymbol{u}_S$, uniformly distributed on the unit sphere, are readily available. The sampled direction $\boldsymbol{l}$ is described by:



(a)



(b)

Fig. 1 Cosine sampling using the tangent sphere: (a) 2D representation; (b) 3D representation

$$\boldsymbol{l} = \left(\boldsymbol{n} + \boldsymbol{u}_S\right)^{\wedge}, \tag{6}$$

where $\boldsymbol{n}$ is the surface normal and the center of the tangent sphere, and the ^ superscript denotes normalization.

BRDFs that go beyond the Lambertian need to capture dependence on directions $\boldsymbol{v}$ and $\boldsymbol{l}$, using the unit vectors to represent directions. In order to ensure reciprocity, the BRDF is most often written as a function of the halfway vector $\boldsymbol{h}$ defined as:

$$\boldsymbol{h} = \left(\boldsymbol{v} + \boldsymbol{l}\right)^{\wedge}, \tag{7}$$

which is what the normal vector should be to produce ideal reflection from $\boldsymbol{v}$ to $\boldsymbol{l}$ or vice versa.

2.5 Physically based BRDFs

Popular physically based BRDFs are based on the microfacet model, assuming a distribution of small ideal reflectors along material interfaces. The BRDF is determined by the probability density of hitting a small surface element with normal $\boldsymbol{h}$, so a random distribution of normals must be prescribed. The challenge is that not all microfacet orientations produce valid reflections, as occlusion and multiple reflections can and do happen in reality. The domain of valid microfacet normals is a cone, in which the integral of the scattering density is typically an elliptic integral, which cannot be evaluated analytically. Therefore, if being physically based means that a microfacet distribution is sampled and invalid samples are rejected, then the rejection probability can only be approximated through expensive numerical methods, like the one presented by Guillén et al. [3] Constructing BRDF-specific helper structures, as proposed by Montes et al. [11], can improve the unpredictable runtime of rejection sampling; nevertheless, this method requires more GPU memory, which we would prefer to avoid.

2.6 Ward and Schlick models

To exemplify the above concerns and provide a baseline, we consider two well-known models. Among the countless existing BRDFs, here we concentrate on the Ward [12, 13] and Schlick [14] models in particular. These are used in Monte Carlo image synthesis extensively and are momentous milestones in BRDF theory and practice. The Ward BRDF is an empirical anisotropic BRDF that employs an elliptic Gaussian distribution on the directional hemisphere for the distribution of normals. The method proposed for importance sampling does not achieve perfect cancellation of factors, and thus weighting (or renormalization) of the random samples is required. The renormalization factor is not simple or transparent, underscored by the fact that several corrections to the original formula were published more than a decade later [13, 15].

Schlick's BRDF is considered a physically based model, and it incorporates occlusion and self-shading terms to complement the microfacet model while maintaining relatively few, reasonably intuitive parameters. Schlick's BRDF can be importance sampled perfectly, with one caveat. Both the Ward and Schlick BRDFs sample the microfacet normal distribution and, especially for grazing angles, are prone to generating invalid reflection directions.

These are considered to be self-obstructed by the surface. The photons are assumed to undergo multiple scattering, and they are re-emitted in a diffuse way. This requires additional random sampling that happens only on a condition, which is not ideal on parallel GPU hardware. On a more theoretical note, the probability of self-obstruction happening

cannot be written exactly in terms of elementary functions, but a rational fraction approximation is given by Schlick.

It can be said that the Ward BRDF is a great practical choice for most rendering tasks, while the widespread use of the Schlick formulas in high-fidelity renderers is justified by their qualities. Indeed, the only requirement we stated in the introduction of this paper that the Schlick BRDF does not meet is that it should not reject samples, which may be a minor or even irrelevant issue, depending on the hardware architecture. Still, both BRDFs use relatively complicated formulas that are empirical or approximate, removing the implementation from physical intuition.

2.7 Tobler, Neumann, Sbert, and Purgathofer model

BRDFs that are physically plausible (i.e., reciprocal), but not strictly microfacet-based, also exist. The previously cited work by Tobler et al. [10] does not only address sampling the diffuse BRDF in a new and simple way but also handles specular reflections. The BRDF is defined in terms of bidirectional distributions over the tangent sphere *S*. The benefit is that when this distribution is sampled, the cosine term does not need to be considered, as it is covered by the projection from the tangent sphere to the directional hemisphere. Also, all directions on the tangent sphere correspond to valid directions on the directional hemisphere, avoiding the rejection of obstructed samples. The benefit is that when this distribution is sampled, the cosine term does not need to be considered, as it is covered by the projection from the tangent sphere to the directional hemisphere. Instead of uniformly distributed samples on *S*, the samples need to be concentrated around the ideal reflection direction. For this, the authors propose using the analogy of the hat-box theorem, orienting a cylinder around the tangent sphere so that its axis coincides with the ideal reflected direction. Samples on the cylinder are generated from two uniformly distributed random variables, sampling the azimuth angle uniformly and the axial position according to an arbitrary PDF with an analytically invertible CDF. This construction again satisfies all of our stated requirements, including perfect importance sampling and no sample rejections, except for the inclusion of anisotropy. As the BRDF is written in terms of the reflected direction and not microfacet normals, relation to tangential directions is not straightforward to incorporate.

Our work, inspired by the above results, explores further options for reciprocal sampling using the tangent sphere, eventually solving the anisotropy problem.

3 The Unit-Quaternionic CML

We tested the idea that the scheme of the Quaternionic Cellular Automaton could work with unit quaternions with real coefficients, a setup we refer to here as the Unit-Quaternionic CML, or UQCML. In order to avoid the eventual degradation due to numerical errors in the realization, we formulate the rule as:

$$x_s^{n+1} = \frac{x_{s-1}^n \cdot x_{s+1}^n}{\left| x_{s-1}^n \cdot x_{s+1}^n \right|}, \tag{8}$$

where superscripts denote iteration indices, and we added a theoretically superfluous normalization to maintain the unit norm of the quaternion in every cell.

The straightforward alternative to UQCML would be to use Shoemake's formula [16] on uniform pseudo-random numbers provided by a standard high-quality PRNG. In general, the Mersenne Twister [17] is the de facto standard solution. When memory usage is at a premium, but PRNG cycle length is a secondary concern, as in GPU kernels or shaders, its variant TinyMT can be used [18].

To evaluate the resulting pseudo-random sequence of rotations, we computed the rotation axes as unit vectors. If these are evenly distributed on the unit sphere, all their components need to be uniformly distributed, as evident from the hat-box theorem [19]. We quantized them to 8 bits and benchmarked the resulting byte stream using the ENT PRNG test suite [20]. We compared the results against the standard C random generator, against the uniform random numbers produced by the TinyMT PRNG, and direction components/polar angles we got by mapping randoms produced by TinyMT to the unit quaternions by Shoemake's formula. Results in Table 1 [20] show that UQCML achieved the same quality as TinyMT if the size of the lattice is at least 5 and an odd number. Even-sized lattices separate their internal state into two disjunct subsets, therefore, even-numbered lattices produce good-quality randoms from size 10. Interestingly, increasing the UQCML site count did not improve randomness. Overall, the statistics are excellent, and we find it remarkable that this extremely simple setup compares equally to the quite refined Mersenne Twister solution. TinyMT uses 127 bits for internal storage, while UQCML-5 stores five quaternions using single-precision floats in 640 bits. This is redundant, as all quaternions are unit length. Later solutions in this paper improve this number.

It may be noted that there are obviously flawed starting states, e.g., if all cells hold an identity quaternion, when the output will not be random. When we set the starting states

Table 1 ENT [20] randomness and uniformity test results over 10 million elements for the UQCML and RL direction components (X, Y, Z) / polar angle(atan) distribution

	Entropy	Chi square dist	Mean	Monte Carlo Pi	Pi error (%)	Serial correlation coeff.
C Rand	7.999993	93.79	127.4905	3.141471657	0.00	0.000096
TinyMT float	7.999979	284.22	127.4467	3.141987657	0.01	0.000107
TinyMT X	7.999981	261.49	127.5023	3.141894057	0.01	−0.000015
TinyMT atan (X,Y)	7.999984	228.26	127.4968	3.141685257	0.00	0.000009
UQCML31 X	7.999984	219.86	127.491	3.141802857	0.01	−0.000285
UQCML31 Y	7.99998	279.58	127.4702	3.141558057	0.00	−0.000195
UQCML31 Z	7.999979	291.74	127.4698	3.143108457	0.05	−0.000021
UQCML31 atan (X,Y)	7.999982	244.51	127.4897	3.142330857	0.02	−0.000035
UQCML31 atan (X,Z)	7.999982	254.84	127.5126	3.141174056	0.01	−0.000203
UQCML31 atan (Y,Z)	7.999982	250.79	127.4706	3.142078857	0.02	−0.000408
UQCML5 X	7.999975	353.07	127.7271	3.128658051	0.41	−0.014072
UQCML5 Y	7.999983	241.25	127.533	3.139750856	0.06	0.000698
UQCML5 Z	7.999975	353.07	127.7271	3.128658051	0.41	−0.014072
UQCML5 atan (X,Y)	7.999976	326.91	127.6642	3.131007652	0.34	−0.008352
UQCML5 atan (X,Z)	7.999976	332.84	127.6801	3.130122052	0.37	−0.008284
UQCML5 atan (Y,Z)	7.99998	283.17	127.5324	3.139736456	0.06	0.000125
RL31 X	7.999982	253.71	127.5184	3.139710056	0.06	−0.000225
RL31 Y	7.999979	294.22	127.478	3.141332457	0.01	0.000408
RL31 Z	7.99998	279.55	127.5291	3.140694056	0.03	−0.000311
RL31 atan (X,Y)	7.999983	235.75	127.5073	3.140653256	0.03	0.000002
RL31 atan (X,Z)	7.999983	233.16	127.5025	3.142323657	0.02	−0.000519
RL31 atan (Y,Z)	7.999983	241.08	127.4668	3.141666057	0.00	0.000056
RL5 X	7.99998	280.81	127.487	3.139707656	0.06	−0.000302
RL5 Y	7.999981	260.62	127.55	3.140074856	0.05	0.000371
RL5 Z	7.99998	275.1	127.4933	3.140082056	0.05	−0.000191
RL5 atan (X,Y)	7.999981	268.29	127.498	3.141567657	0.00	−0.00049
RL5 atan (X,Z)	7.999981	269.98	127.4813	3.140842856	0.02	−0.000358
RL5 atan (Y,Z)	7.999982	252.49	127.5796	3.139119656	0.08	0.000075

using uniform random rotations generated by Shoemake's formula, we never encountered any flawed configurations.

4 The Reflection Lattice

While UQCML can be used to generate uniform rotations, a more ubiquitous problem is the generation of random directions, e.g., in the context of random walks in Monte Carlo image synthesis algorithms. Our idea was to construct a CML with 3D vectors instead of quaternions but still using non-commutative operations. Obvious candidates are the cross product and the reflection of a vector onto another vector. These operations are implemented in shader languages as the cross and reflect intrinsics, respectively:

$$\boldsymbol{v}_s^{n+1} = \boldsymbol{v}_{s-1}^n \times \boldsymbol{v}_{s-1}^n, \tag{9}$$

$$\boldsymbol{v}_s^{n+1} = \boldsymbol{v}_{s-1}^n - 2\left(\boldsymbol{v}_{s-1}^n \cdot \boldsymbol{v}_{s+1}^n\right)\boldsymbol{v}_{s+1}^n. \tag{10}$$

The stage transition rule is Eq. (9) or Eq. (10). As before, the results can be normalized to avoid number-representation inaccuracies.

The alternatives would be using uniform randoms mapped to spherical coordinates, using the Box-Muller transform, then normalizing, or using rejection sampling on uniforms in 3D. Rejection sampling is not done in constant time and is thus poorly suited to GPU contexts. We compare the Reflection Lattice (RL) to using random spherical coordinates and the Box-Muller transform (Table 1). The reference methods need uniform randoms, which we supplied with the TinyMT PRNG.

Results were similar or better than those of the UQCML, with both uniform distribution on the sphere and low serial correlation. Visually, the results of UCQML, RL, and the reference methods are similar. Again, using more than five

sites did not improve statistics. The five-site Reflection Lattice (RL5) stores its state on 480 bits, which can be reduced to 320 bits by exploiting the fact that all vectors are of unit length. This is still more than the 127 bits of TinyMT, but the implementation is simpler, and the results are somewhat better. Also, the RL can work on hardware where integer operations are not available or in OpenGL ES 2.0 contexts.

5 The summed RL

Generating samples from a three-dimensional normal distribution is possible if we sum the output of our CMLs, following the sum-of-uniforms (or CLT) scheme of generating Gaussian samples. Although the values at the sites cannot be regarded as independent, the Shapiro–Wilk [21] test shows that the samples can be accepted as that of a normal distribution with high confidence, especially at higher site counts (Table 2) [21]. In Table 2 each row shows results for a given lattice size, each column gives the value for a given sample count. A value below 0.05 means that the hypothesis that the samples were taken from a normal distribution is rejected. Note that the reference value in the table does not actually depend on z, its variation is only random, highlighting how the test values should not be given more significance beyond passing or failing the test. In fact, we generate samples of the Irwin–Hall distribution, which can be used as an approximation of the Gaussian. As the variance of the Irwin–Hall is $z/12$, with z being the number of uniform distributions added, the summed and normalized distribution is limited to $\left[-\sqrt{12z},+\sqrt{12z}\right]$, where z is the size of the lattice. Even for a lattice size of 5, these are beyond 7 standard deviations.

Even so, compared to more refined normal distribution generation methods, tails are sampled relatively poorly, which shows up in the Shapiro test at higher sample counts. In practice, if samples are generated for the purpose of Monte Carlo rendering in a per-thread or per-thread-block fashion, the required sample counts can be expected to remain under 100.

Using Sherman's upper bound [22] for the approximation of the normal CDF using the Irwin–Hall distribution, we can say that for a lattice size of 5, we get a worst-case integration error of less than 2.5%. The same value for $z = 31$ is 0.0024%.

We can conclude that for the purposes of Monte Carlo rendering, our SRL-5 provides an approximate distribution that leads to slightly biased estimates, but SRL-31 can be considered error-free. On GPU hardware with 32- or 64-lane architecture, it would be ideal to implement a per-block PRNG with parallel execution.

6 Group shared memory utilization

Group shared memory utilization significantly reduces the per-thread memory requirement, which can greatly enhance computation performance on the GPU. Compute shader threads were dispatched in groups of 32. Each group maintains a CML, with every thread being responsible for updating one cell. The calculation step can be executed simultaneously for every instance of the compute shader group without expensive code branching. This way the RL is able to generate a new state in every cell while executing only a reflection and a normalization step and only requiring 12 bytes of memory per shader thread.

Table 2 Results of the Shapiro–Wilk [21] test for samples generated by the summed RL

z	Reference	500	1000	1500	2000	2500	3000	3500	4000	4500	5000
3	0.36	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	0.20	0.19	0.12	0.16	0.10	0.06	0.05	0.08	0.01	0.01	0.00
7	0.16	0.25	0.32	0.29	0.33	0.24	0.02	0.01	0.02	0.01	0.00
9	0.14	0.46	0.42	0.30	0.15	0.06	0.02	0.05	0.04	0.02	0.01
11	0.07	0.27	0.61	0.26	0.27	0.41	0.41	0.38	0.46	0.40	0.36
13	0.41	0.81	0.48	0.26	0.21	0.31	0.15	0.26	0.18	0.20	0.13
15	0.75	0.43	0.20	0.89	0.94	0.95	0.88	0.68	0.55	0.33	0.44
17	0.87	0.51	0.33	0.50	0.49	0.40	0.54	0.40	0.49	0.12	0.09
19	0.99	0.60	0.62	0.22	0.28	0.52	0.33	0.33	0.09	0.13	0.40
21	0.77	0.48	0.09	0.09	0.51	0.65	0.59	0.42	0.36	0.64	0.60
23	0.68	0.74	0.66	0.63	0.55	0.84	0.74	0.64	0.56	0.56	0.52
25	0.84	0.53	0.49	0.38	0.59	0.87	0.80	0.46	0.39	0.32	0.29
27	0.42	0.46	0.36	0.31	0.51	0.67	0.23	0.38	0.33	0.73	0.52
29	0.26	0.16	0.44	0.57	0.08	0.29	0.46	0.33	0.23	0.15	0.23
31	0.56	0.48	0.34	0.49	0.70	0.41	0.16	0.14	0.16	0.17	0.46

7 Sampling on the tangent sphere using uniform directional samples

The hat-box mapping for generating samples on the tangent sphere S is a convenient method when uniformly distributed random numbers are available. However, as random number generation is challenging on some hardware, it is possible that pre-generated randoms are used, possibly the same or correlated ones in all pixels. In such a setup, pre-generating random numbers already uniformly distributed on the sphere means no overhead. Alternatively, methods that generate random directions directly can be used, such as the one described in this paper. Therefore, it is of interest how these uniformly distributed directions can be transformed into non-uniform sampling on the tangent sphere.

The hat-box analogy can still be used, but there is no need to transform uniform variables onto the cylinder. Let $\boldsymbol{r}_S \in S$ be the fixed point (or pivot) of the transformation. Pivot $\boldsymbol{r}_S$ can certainly be the normal itself, e.g., to produce a distribution of halfway vectors around the normal, or it could be an ideal reflection direction to produce samples of a specular lobe, as chosen by Tobler et al. [10]

The axis coordinate on the hat-box cylinder would be:

$$z = \boldsymbol{q}_S \cdot \boldsymbol{r}_S. \tag{11}$$

Transformed by an arbitrary PDF, we can obtain z'. Then the transformed sample is:

$$\boldsymbol{q}'_S = \boldsymbol{r}_S \cdot z' + \left(\boldsymbol{q}_S - \boldsymbol{r}_S \cdot z\right) \cdot \sqrt{\frac{1-z^2}{1-z'^2}}. \tag{12}$$

8 Halfway-on-tangent-sphere BRDF

The shortcoming of the tangent-sphere sampling method is that it does not work with a microfacet normal distribution, and therefore cannot incorporate the anisotropy of these microfacets. We could define:

$$\boldsymbol{h}_S = \left(\boldsymbol{v}_S + \boldsymbol{l}_S\right)^{\wedge}, \tag{13}$$

where $\boldsymbol{v}_S$ and $\boldsymbol{l}_S$ are related to $\boldsymbol{v}$ and $\boldsymbol{l}$ as given by Tobler et al. [10] and shown in Fig. 2:

$$\boldsymbol{v}_S = 2\left(\boldsymbol{n}\cdot\boldsymbol{v}\right)\boldsymbol{v} - \boldsymbol{n}, \;\; \boldsymbol{l}_S = 2\left(\boldsymbol{n}\cdot\boldsymbol{l}\right)\boldsymbol{l} - \boldsymbol{n}. \tag{14}$$

Then, it is a logical variant to formulate the BRDF in terms of $\boldsymbol{h}_S \cdot \boldsymbol{n}$. Samples of $\boldsymbol{h}_S$ can be generated using the method in Section 7 or on an elliptic cylinder that has the normal as its axis, and projected onto S. Then, a random sample for $\boldsymbol{v}$ can be generated by projecting it onto the tangent sphere to get:

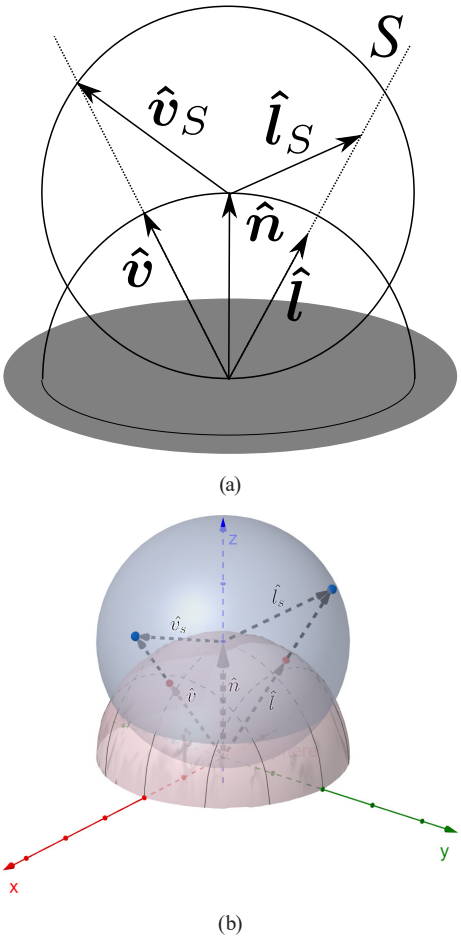


Fig. 2 Cosine sampling using the tangent sphere: (a) 2D representation; (b) 3D representation

$$\boldsymbol{v}_S = 2\left(\boldsymbol{n}\cdot\boldsymbol{v}\right)\boldsymbol{v} - \boldsymbol{n}, \tag{15}$$

reflecting this on $\boldsymbol{h}_S$ gives:

$$\boldsymbol{l}_S = -\boldsymbol{v}_S + 2\boldsymbol{n}\left(\boldsymbol{h}_S \cdot \boldsymbol{v}_S\right). \tag{16}$$

Finally, $\boldsymbol{l}_S$ is projected back to the directional hemisphere as:

$$\boldsymbol{l} = \left(\boldsymbol{n} + \boldsymbol{l}_S\right)^{\wedge}. \tag{17}$$

But the problem with this idea, and the likely reason the original authors did not consider this, is that isotropic lobes on the tangent sphere are not isotropic on the direction sphere. As circles are mapped to ellipses, blurring along the plane spanned by vectors $\boldsymbol{v}$ and $\boldsymbol{n}$ appears. Note that this is not the regular anisotropy induced by the scratches on the surface of the material but a viewpoint-dependent blur.

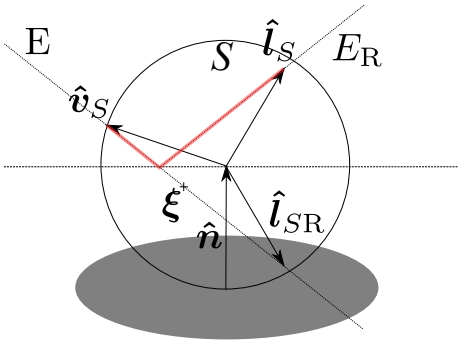
9 Volumetric sampling

As the halfway-on-tangent-sphere BRDF does not provide the required behaviour, it is logical to look for another way

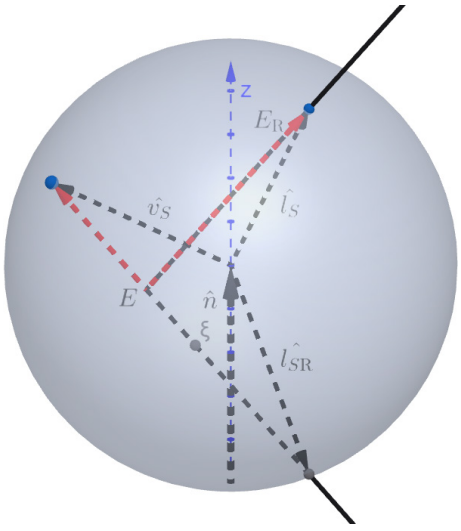
to relate $\boldsymbol{v}_S$ and $\boldsymbol{l}_S$ in a symmetric manner. If distribution on the surface of the tangent sphere does not work, a distribution over the volume of the tangent sphere could. Follow the process in Fig. 3.

If we select a point $\boldsymbol{\xi}$ in the tangent sphere, then drawing the line E through $\boldsymbol{v}_S$ and $\boldsymbol{\xi}$ intersects S at another point $\boldsymbol{l}_{SR}$. This direction is not near the ideal reflection direction, but if it is reflected onto the normal to get $\boldsymbol{l}_S$, that one is. The process is entirely symmetric only if we restrict the sampled distribution to be symmetric along the normal. Then, the integrals along lines E and E_R are equivalent. The sampled light path can be seen to go from $\boldsymbol{v}_S$ to $\boldsymbol{l}_S$ while being reflected from the equatorial plane of S (highlighted in red in Fig. 3).

This way, if the distribution is concentrated around the center of S, the sampled directions will be close to the ideal reflected direction. On the other hand, a uniform distribution on S gives the diffuse BRDF. The distribution has to be so that it can be both efficiently sampled and the probability of generating a sample can be determined.



(a)



(b)

Fig. 3 Volumetric sampling of directions using the tangent sphere: (a) 2D representation; (b) 3D representation

All points along line E map to the same sample, so we have to integrate the volume PDF along E. This needs to be the same path if $\boldsymbol{v}_S$ and $\boldsymbol{l}_S$ are switched, which is true if the volume distribution is symmetric on the equatorial plane of S. This integral can be very challenging to evaluate, with a closed-form solution not existing for most distributions.

If the PDF is constant or piecewise constant, then the length of the path intersecting a domain of constant value could be found. However, we cannot expect that to lead to natural BRDF models. If the distribution is a 3D normal distribution, the integral is again easy to evaluate. The easiest case is what we call the *isotropic full Gaussian* version. In this case we do not limit samples to the inside of S, and the line E along which the same directional sample is obtained is an infinite line, not a line segment. A 2D projection of a 3D normal distribution is also a normal distribution, so we only have to find the distance between E and the center of S, and obtain the value of the projected distribution there.

We first project $\boldsymbol{v}$ and $\boldsymbol{l}$ to $\boldsymbol{v}_S$ and $\boldsymbol{l}_S$, respectively, using the same formulas as before. Then the distance is:

$$t = \sqrt{\frac{1+\boldsymbol{v}_S\cdot\boldsymbol{l}_S-2(\boldsymbol{v}_S\cdot\boldsymbol{n})(\boldsymbol{l}_S\cdot\boldsymbol{n})}{2}}, \tag{18}$$

which is notably symmetric in $\boldsymbol{v}_S$ and $\boldsymbol{l}_S$. The BRDF is:

$$f_{r,isoGauss} = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{t^2}{2\sigma^2}}. \tag{19}$$

This, however, produces realistic reflection only as long as the variance of the Gaussian is small, and samples outside of the tangent sphere are not generated with a significant probability. As the variance increases, the distribution of the direction of the line E converges to a uniform distribution. This leads to a non-uniform distribution on the tangent sphere, featuring stronger retro-reflection. Instead, a BRDF that converges to the diffuse case is preferable. We call this version the *isotropic clipped Gaussian*, where Gaussian samples generated outside of the tangent sphere are simply projected onto the sphere to produce a uniform distribution, exactly recreating the diffuse case. In this case, however, the BRDF consists of a diffuse and a specular component. The probability of selecting the specular component, i.e., a point within the sphere along line E is the integral of the Gaussian distribution along line E between the entry and exit points. With the half-length of the segment:

$$\alpha = 1-t^2, \tag{20}$$

the integral can be expressed using the Gaussian error function, making the weight of the specular:

$$\zeta = \frac{\alpha}{\sqrt{2}\sigma} - \frac{-\alpha}{\sqrt{2}\sigma}. \tag{21}$$

The BRDF is:

$$f_{r_i soGauss_{clipped}} = \frac{1}{\sqrt{2\pi}\sigma}\zeta + \frac{1-\zeta}{\pi}, \tag{22}$$

where the second term captures the diffuse component.

It is possible to apply any affine transformation on the Gaussian distribution, maintaining the ability to evaluate the PDF (as lines are transformed to lines). The transformation cannot depend on $\boldsymbol{v}$, as that breaks reciprocity, but it can depend on model tangents and binormals, as in Wards and Schlick's models, introducing material anisotropy. If we combine this with the clipping of the Gaussian distribution to the tangent sphere, the limits of the integration change, as the tangent sphere, in the primary distribution space, is an ellipsoid. In this case, the BRDF can be evaluated by transforming $\boldsymbol{v}_S$ and $\boldsymbol{l}_S$ to tangent space, scaling them with the anisotropy factor to primary distribution space, and evaluating ζ with the respective values of the 1D error function at those locations. Note that this also makes the formerly diffuse component anisotropic.

10 Volumetric scattering

Another possible application of the presented lattice PRNG is creating volumetric scattering effects. Volumetric scattering, in contrast to the aforementioned applications, requires random distance in addition to the random direction vector to replicate the random walk inside solid bodies. In this scenario, we demonstrate that our RL can create directions so effectively that it is more beneficial to use RL to generate directions and TinyMT to generate the distance simultaneously than it is to use TinyMT exclusively to generate both.

11 Results

In order to demonstrate the various application possibilities of our novel PRNG, we constructed four path tracing scenarios:

1. The first is a simple Cornell box to demonstrate the efficiency of the RL (Fig. 4). We measured a 10% FPS increase when generating the directions with RL compared to TinyMT without any perceptible visual change. We also implemented our Gaussian distribution-based BRDF with both random generation methods and compared them to Schlick and Ward alternatives. The efficiency of our BRDF improved by 25% FPS with RL while visually remaining the same. Our BRDF looked different from Ward and Schlick, but all three seemed plausible. The Schlick version was 18% while the Ward version was 26% faster than our BRDF, but when the generated direction was invalid, we reverted to diffuse sampling. If we were to reject samples in this case, then we could achieve varying results depending on the number of surfaces that are shown from sharp viewing angles.
2. In the second scenario we showcase that our BRDF provides a perfectly continuous transition between ideal mirror and completely diffuse surfaces while also allowing anisotropy (Fig. 5).
3. In the third scene we can perceive that our BRDF can create similar visuals to the references in complex environments (Fig. 6). We think that the only big

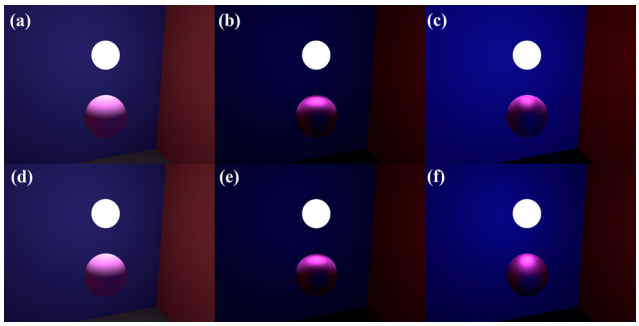


Fig. 4 Simple Cornell boxes: (a) Completely diffuse with TinyMT 36 FPS, (b) our BRDF with TinyMT 31 FPS, (c) Ward 49 FPS, (d) completely diffuse with RL32 40 FPS, (e) our BRDF with RL32 39 FPS, (f) Schlick 46 FPS

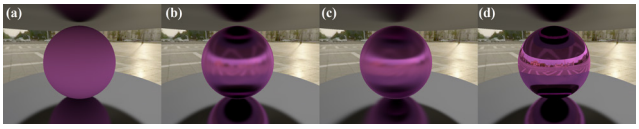


Fig. 5 Our BRDF with shininess: (a) 0, (b) 0.7 (isotropic), (c) 0.7 (anisotropic), (d) 1

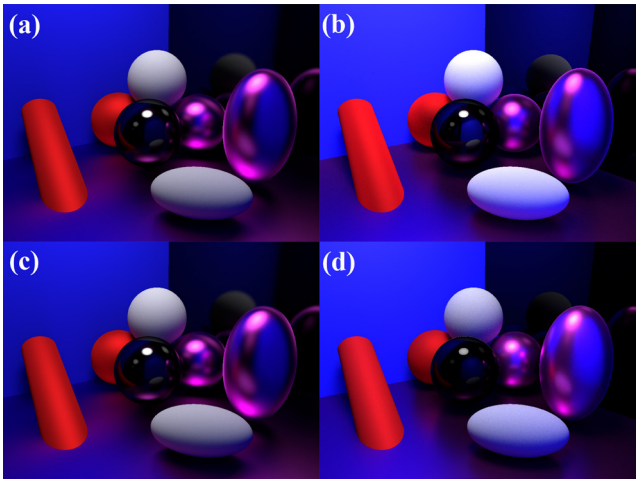
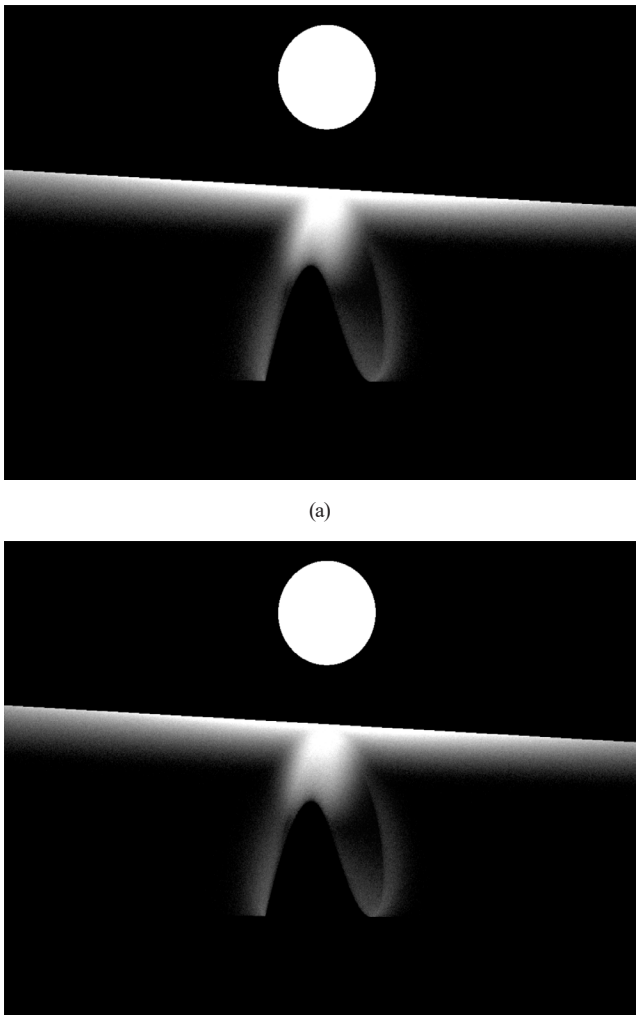


Fig. 6 Scene with multiple objects: (a) our BRDF with TinyMT 9 FPS, (b) Ward 11 FPS, (c) our BRDF with RL32 10 FPS, and (d) Schlick 11 FPS

differences between our BRDF and the references are on the edges of shiny surfaces, where the view angle is almost parallel with the surface. In this case, Ward and Schlick BRDFs can easily generate invalid angles that cause a darkened contour in the original Ward and diffuse colour dominancy over specular colour in the later Ward variants and Schlick BRDF. We think that our solution handles this behaviour in a more correct way. In this scene our BRDF was 10% faster with RL compared to TinyMT, and Ward and Schlick were even faster by 10% when rejection was circumvented.

4. The last scenario is a volumetric scattering test case to prove that our lattice can be used in a wide range of applications (Fig. 7). The reference volumetric scattering generated the directions and the distance with TinyMT. In our version the direction generation was changed to RL, but the distance generation remained unchanged. We measured a 10% FPS increase applying RL while the visual changes were very minor.

We implemented our ideas into practice in a DirectX 11 environment. Our technical parameters were: 1024px × 1024px render target, NVIDIA GeForce GTX 1080 Ti graphics card and Intel i7 4790k processor. The compute shader instances of the path tracers calculated 32 different routes until the 8th depth per frame. The volumetric scattering was evaluated until the 64th step in the inside of the objects.



(a)

(b)

Fig. 7 Subsurface scattering: (a) TinyMT 45 FPS, (b) RL32 and TinyMT 56 FPS

12 Conclusions

Our proposed PRNG performs around 10% faster and requires 24% less memory when generating directions compared to the de facto standard TinyMT version while providing correct distributions without any visual glitches. We also measured the correctness of this PRNG with ENT and the derived 3D Gaussian distribution with Shapiro–Wilk test, achieving good results. Our BRDF model improves on the model of Tolber et al. by allowing for anisotropy, and it is better suited to GPU kernels than the Ward or Schlick BRDFs because it does not reject samples. As results in Fig. 5 show, a smooth transition from the ideal mirror to a completely diffuse appearance is supported. Both the sampling process and the BRDF evaluation are relatively simple, and no renormalization is needed. The visual quality of the new BRDF is very similar to Ward and Schlick models but more vivid on sharp angles when the sample generation of these alternatives could fail. We also proved that our lattice can achieve the aforementioned improvements even if used together with alternative random generator solutions.

References

[1] Coddington, P. D. "Random Number Generators for Parallel Computers", Syracuse University, Syracuse, NY, USA, No. 1.1, 1997. [online] Available at: https://surface.syr.edu/cgi/viewcontent.cgi?article=1012&context=npac [Accessed: 11 May 2025]

[2] Thomas, D. B., Luk, W., Leong, P. H. W., Villasenor, J. D. "Gaussian random number generators", ACM Computing Surveys, 39(4), 11, 2007. https://doi.org/10.1145/1287620.1287622

[3] Guillén, I., Ureña, C., King, A., Fajardo, M., Georgiev, I., López-Moreno, J., Jarabo, A. "Area-Preserving Parameterizations for Spherical Ellipses", Computer Graphics Forum, 36(4), pp. 179–187, 2017. https://doi.org/10.1111/cgf.13234

[4] Teichroew, D. "Distribution sampling with high speed computers", PhD thesis, North Carolina State College, 1953.

[5] Wallace, C. S. "Fast Pseudorandom Generators for Normal and Exponential Variates", ACM Transactions on Mathematical Software, 22(1), pp. 119–127, 1996. https://doi.org/10.1145/225545.225554

[6] Tomassini, M., Sipper, M., Perrenoud, M. "On the Generation of High-Quality Random Numbers by Two-Dimensional Cellular Automata", IEEE Transactions on Computers, 49(10), pp. 1146–1151, 2000. https://doi.org/10.1109/12.888056

[7] McKeever, B. M. "The use of non-commutative algebra in cryptographically secure pseudo-random number generators", BSc thesis, University of Richmond, 1996. [online] Available at: https://scholarship.richmond.edu/cgi/viewcontent.cgi?article=1586&context=honors-theses [Accessed: 11 May 2025]

[8] Greenfield, G. R. "On Quaternionic Pseudo-Random Number Generators", University of Richmond, Richmond, VA, USA, Rep. TR-96-01, 1996. [online] Available at: https://scholarship.richmond.edu/cgi/viewcontent.cgi?article=1023&context=mathcs-reports [Accessed: 11 May 2025]

[9] Chazottes, J.-R., Fernandez, B. "Dynamics of Coupled Map Lattices and of Related Spatially Extended Systems", Springer Berlin, Heidelberg, 2005. ISBN 978-3-540-24289-5 https://doi.org/10.1007/b103930

[10] Tobler, R. F., Neumann, L., Sbert, M., Purgathofer, W. "A new Form Factor Analogy and its Application to Stochastic Global Illumination Algorithms", In: Eurographics Workshop on Rendering Techniques (EGSR 1998), Vienna, Austria, 1998, pp. 35–44. ISBN 978-3-7091-6453-2 https://doi.org/10.1007/978-3-7091-6453-2_4

[11] Montes, R., Ureña, C., García, R., Lastra, M. "An Importance Sampling Method for Arbitrary BRDFs", In: VISIGRAPP: International Joint Conference on Computer Vision, Imaging and Computer Graphics (VISIGRAPP 2008), Funchal, Madeira, Portugal, 2008, pp. 41–54. ISBN 978-3-642-10226-4 https://doi.org/10.1007/978-3-642-10226-4_4

[12] Ward, G. J. "Measuring and Modeling Anisotropic Reflection", In: SIGGRAPH ,92: Proceedings of the 19th Annual Conference on Computer Graphics and Interactive Techniques, Vancouver, Canada, 1992, pp. 265–272. ISBN 978-0-89791-479-6 https://doi.org/10.1145/133994.134078

[13] Walter, B. "Notes on the Ward BRDF", Cornell Program of Computer Graphics, Ithaca, NY, USA, Rep. PCG-05-06, 2005. [online] Available at: https://www.graphics.cornell.edu/pubs/2005/Wal05.pdf [Accessed: 11 May 2025]

[14] Schlick, C. M. "An Inexpensive BRDF Model for Physically-based Rendering", Computer Graphics Forum, 13(3), pp. 233–246, 1994. https://doi.org/10.1111/1467-8659.1330233

[15] Dür, A. "An Improved Normalization for the Ward Reflectance Model", Journal of Graphics Tools, 11(1), pp. 51–59, 2006. https://doi.org/10.1080/2151237X.2006.10129215

[16] Shoemake, K. "III.6 - Uniform Random Rotations", In: Kirk, D. (ed.) Graphics Gems III (IBM Version), Morgan Kaufmann, 1992, pp. 124–132. ISBN 978-0-12-409673-8 https://doi.org/10.1016/B978-0-08-050755-2.50036-1

[17] Matsumoto, M., Nishimura, T. "Mersenne Twister: A 623-Dimensionally Equidistributed Uniform Pseudo-Random Number Generator", ACM Transactions on Modeling and Computer Simulation (TOMACS), 8(1), pp. 3–30, 1998. https://doi.org/10.1145/272991.272995

[18] Saito, M., Matsumoto, M., Roca, V., Baccelli, E. "RFC 8682 TinyMT32 Pseudorandom Number Generator (PRNG)", Internet Engineering Task Force (IETF), Wilmington, DE, USA, 2020. https://doi.org/10.17487/RFC8682

[19] Cundy, H., Rollett, A. "Sphere and Cylinder-Archimedes' Theorem", In: Mathematical Models, Tarquin Publications, 1989, pp. 172–173. ISBN 0906212200

[20] Walker, J. "ENT-A Pseudorandom Number Sequence Test Program", [online] Available at: https://www.fourmilab.ch/random/ [Accessed: 11 May 2025]

[21] Shapiro, S. S., Wilk, M. B. "An analysis of variance test for normality (complete samples)", Biometrika, 52(3–4), pp. 591–611, 1965. https://doi.org/10.1093/biomet/52.3-4.591

[22] Sherman, R. "Error of the normal approximation to the. sum of N random variables", Biometrika, 58(2), pp. 396–398, 1971. https://doi.org/10.1093/biomet/58.2.396