

# 経路誘導による多光源照明計算の効率化

杉田 俊平<sup>1,a)</sup> 藤代 一成<sup>2,b)</sup>

**概要:** 映画・建築・デザイン分野では、大域照明を考慮した、きわめて写実性の高いレンダリング技術が利用されている。その計算過程においてモンテカルロ推定を利用する場合、出力画像には最終的な収束値から大きく外れた点を多数含んでしまう。そのため、低サンプル数でかつ低分散な出力画像が得られる最適化手法が検討されている。経路誘導は強度分布を事前推定し、効率的な経路を構築することで分散低減を実現する。しかし多光源シーンでは各光源からの寄与を得ることが困難であるため、強度分布の収束が遅く分散低減効果が低いという課題がある。そこで本研究では、光源サンプリングを改善することで多光源照明計算における経路誘導の計算効率を高める最適化手法について検討を行う。

## Improving Efficiency of Many-light Global Illumination with Path Guiding

SUGITA SHUMPEI<sup>1,a)</sup> FUJISHIRO ISSEI<sup>2,b)</sup>

**Abstract:** Photorealistic rendering has been used in the fields of film production, architecture, and design to account for global illumination. When using Monte Carlo estimation for such rendering, the resulting images may contain some points that deviate significantly from a final convergence value. An optimization method is needed to obtain output images with a small number of samples and low variance. Path guiding is a method to reduce dispersion by estimating the intensity distribution in advance and constructing efficient paths. However, in many-lights scenes, it is difficult to obtain contributions from each of the light sources, resulting in slow convergence of the intensity distribution and limited effectiveness of dispersion reduction. This research, therefore, aims to improve the computational efficiency of path guiding in many-lights illumination by sophisticating light source sampling.

### 1. 背景と目的

映画・建築・デザイン分野では、大域照明を考慮した、きわめて写実性の高いレンダリング技術が利用されている。大域照明計算の主流手法の一つであるモンテカルロ・レイトレーシング法 [3] では、経路積分の計算にモンテカルロ法を利用するため、サンプル数が十分でない場合、出力画像は最終的な収束値から大きく外れた点を多数含んでしまう。そのため、低サンプル数でかつ低分散な出力画像が得られる最適化手法が検討されている。

経路誘導 (path guiding) は強度分布を事前推定し、効率的な経路を構築することで分散低減を実現する最適化手法であり、映画産業で注目されている。[10]. この最適化手法は、単光源からのコースティクスを含むシーン等では高い分散低減効果が確認されている [5]. しかし、多光源シーンでは各光源からの寄与を得ることが困難であるため、強度分布の収束に時間を要する。そのため、粗い強度分布により構築される経路は効率が低く、分散低減効果が上がらないという課題がある。そこで、多光源シーンで経路誘導を適用するには、光源からの寄与を効率的に獲得する適切な光源サンプリング手法との組合せが必要であると考えられる。本研究では光源サンプリングを改善することで多光源照明計算における経路誘導の計算効率を高める最適化手法について検討を行う。提案手法の概略を図 1 に示す。

<sup>1</sup> 慶應義塾大学 大学院理工学研究科  
Graduate School of Science and Technology, Keio University

<sup>2</sup> 慶應義塾大学 理工学部  
Faculty of Science and Technology, Keio University

a) shumpei.sugita@fj.ics.keio.ac.jp

b) fuji@ics.keio.ac.jp

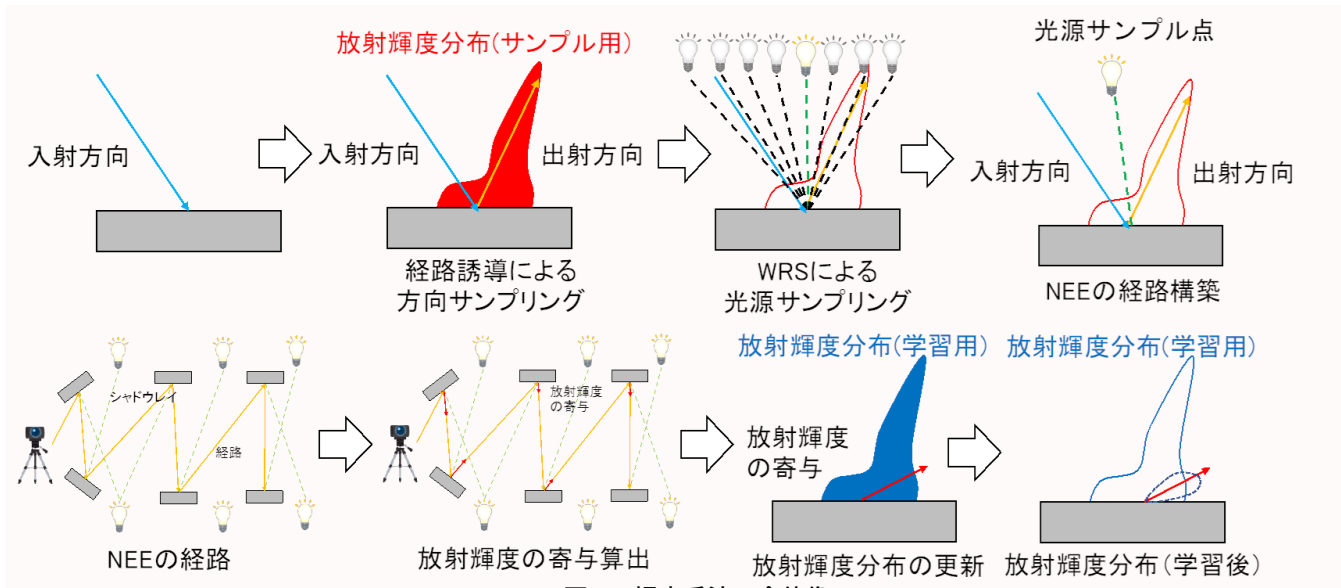


図 1: 提案手法の全体像

## 2. 関連研究

経路誘導と光源サンプリングの最適化に関連する研究を紹介し、提案手法の位置づけを明確にする

### 2.1 経路誘導

Müller [5] は、経路誘導におけるデータ構造として SD-Tree を採用し、出力画像の推定と強度分布の推定を同時に行うオンライン学習によってパストレーシングの分散低減を行う手法を提案した。SD-Tree は図 2 で示すように、構造格子に似た特性をもつデータ構造であり、追加のフィッティング処理なしで分布の格納とサンプリングが可能である。SD-Tree を用いた経路誘導の処理は、強度分布を推定する学習段階と、推定済み分布を用いて効率的に経路構築を行う描画段階に分かれる。これらの処理を並行して実行することで、分布推定と経路構築が相互に効率化されて、分散を効果的に低減できる。

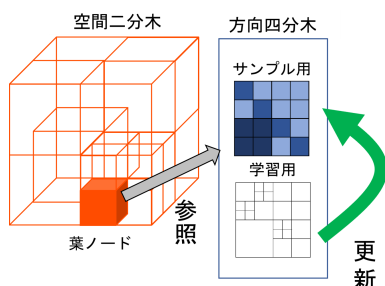


図 2: Müller が提案した SD-Tree の構造。空間二分木と葉ノードに紐づけられた方向四分木 (サンプル用, 学習用) から構成される。十分な学習後、学習用の方向四分木をサンプル用として利用する。

### 2.2 RIS と光源サンプリングの効率化

RIS (**R**esampled **I**mportance **S**ampling) [9] は、候補分布を用いて複数の点を生成し、標的分布と生成分布から求まる重みによる確率的選択によって標的分布に限りなく近い点を生成するサンプリング手法である。通常の重点サンプリングと比較して、逆関数の記述が困難な分布を扱うことが可能である点で、RIS は適用範囲が広い。ただし、標的分布に近いサンプリングを行うには十分な候補サンプル数が必要である反面、候補サンプル数ごとに候補サンプル生成と重み評価を実行するため、計算量が大い点が課題として挙げられる。Bitterli ら [1] は、時間的・空間的相関を考慮した候補サンプル点の再利用を行うことで生成回数を可能な限り抑えながら RIS を光源サンプリングに適用し、リアルタイム多光源直接照明計算を実現する手法を提案した。ただし、Bitterli らの提案する空間的再利用は、スクリーン空間上で行われるため、そのままでは直接照明計算にしか適用できない。Ouyang ら [7] は、Bitterli らの手法を拡張して大域照明を扱うことを提案しているが、スクリーン空間上での相関が失われていくため、長い経路長を扱うことが困難である。そこで Boissé [2] は、スクリーン空間ではなくワールド空間上で再利用を行い、間接照明計算に適用する手法を提案した。この手法はワールド空間上で再利用が行われるため、経路長に依存せず他の最適化手法との組合せも比較的容易に行える。これらの手法における時間的・空間的再利用の処理は GPU 上で並列実行可能であり、最先端の GPU ハードウェアに最適化されている。

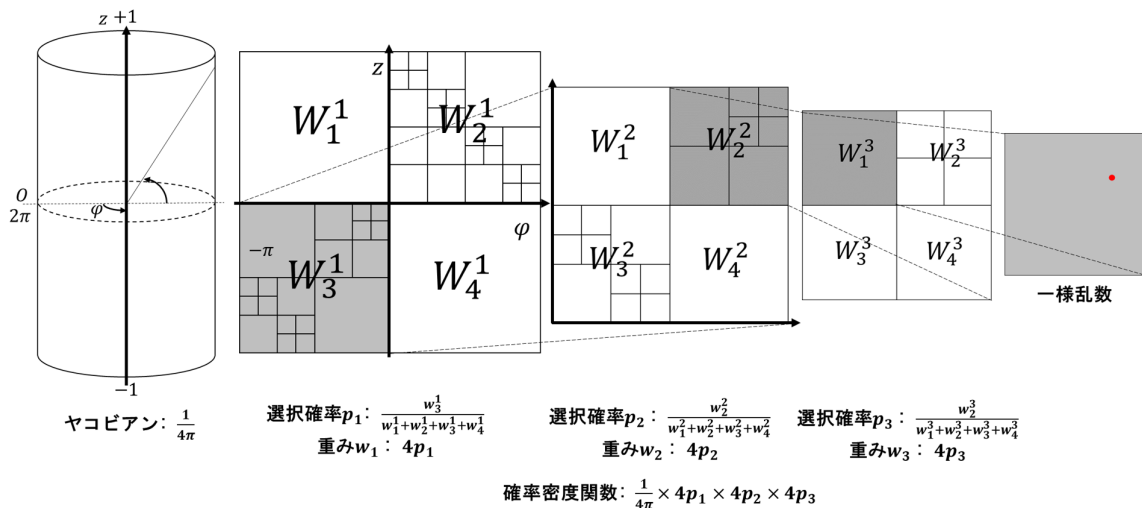


図 3: SD-Tree の方向サンプリング. SD-Tree の方向四分木は円筒座標系を投影したものとなっており、サンプリングでは重みに基づき各層のノードを選択しながら走査していき、葉ノードで一様分布から点生成を行うことで最終的な方向を決定する。ここで、最終的な確率密度関数は各層におけるノードの選択確率と円筒座標系へのヤコビアンとの積をとったものとなる。

### 3. 提案手法

提案手法で利用する経路誘導と光源選択のアルゴリズムについて以下で述べる。図 1 で示したように、提案システムは、経路誘導として SD-Tree を利用し、効率的な光源サンプリングにより強度分布と出力画像の収束を同時に効率化する。前節で述べたように、RIS を用いた光源サンプリングは時間的・空間的再利用に適しているが、本研究では通常の RIS による効率化を行う。

#### 3.1 SD-Tree による経路誘導

入射方向が与えられたとき、次の経路を決定するため、次の方向と対応する確率密度関数の値を求める必要がある。本研究では、強度分布に対して重点サンプリングを行うため、現在の経路点に対応する方向四分木を空間二分木内で検索して、対応するサンプリング用の四分木からサンプリングを行う。図 3 に方向サンプリングの生成過程を示した。ただし SD-Tree の確率密度分布は方向四分木の葉ノードの値が 0 であるとき値 0 となるため、特定の方向について寄与が 0 になる恐れがある。そこで実際には、方向四分木単体による方向サンプリングではなく、マテリアルに沿った方向サンプリングと合わせた多重的重点サンプリングを行う点に注意する必要がある。次に各経路点における強度が確定した段階で、その寄与を SD-Tree に学習する。ここでは経路点に対応する学習用の四分木を方向サンプリング時に記録しておき、四分木の各層ごとに経路点における出射方向に対応するノードを求めたうえで寄与を重みに加算していく。最後に SD-Tree の学習が十分に進んだ段階でサンプリング用の方向四分木を学習用のものに置き換え、より複雑な分布形状を推定するため SD-Tree のノード分割を行

う。一連の処理を反復することで、強度分布の推定を進めていく。この段階では描画と学習が同時に行われるが、描画処理として考えると時間効率が低い。そこで十分学習が進んだ段階で分布推定を停止し、同一の SD-Tree を描画処理で繰り返し利用する。ここで学習段階のサンプル数を増やすほど、最終的な推定分布の精度が高く、描画段階にサンプル当たりの分散低減効果が高まる反面、学習にかかる時間分描画時間が増大するため、単位時間当たりの平均サンプル数が低下するトレードオフが存在する。つまり全サンプル数を学習段階に割り当てた場合、サンプル当たりの分散低減効果は最大となるが、単位時間当たりの平均サンプル数が最小となるため、時間基準では最適でない。ゆえに、提案手法では時間効率を考慮して全サンプルを学習段階に割り当てるのではなく、適切なサンプル数分描画段階に割り当てている。

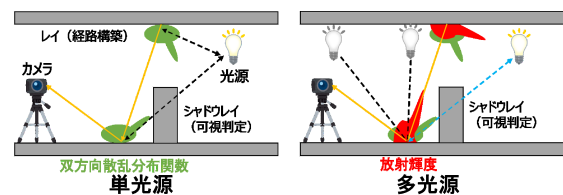


図 4: 単光源 NEE と提案手法による経路構築. 単光源 NEE では、光源からの寄与を求める経路と各媒質との交点で反射方向と反射率を求める経路からなる枝状の経路群を構築する。提案手法では光源からの寄与に近い分布関数に基づいて選択された光源に対して NEE を実行する。

### 3.2 NEE と光源選択

提案手法のアルゴリズムは、確率的に選択された光源上の点に対して明示的サンプリングを行う **NEE (Next Event Estimation)** に追加の処理を加えたものである。図 4 左に単光源 NEE における経路構築を示した。単光源の場合では、空間的同質性が高いため一様分布による選択であっても効果的なサンプリングが可能である。しかし多光源照明問題の場合、光源の位置・形状・強度が異なるため、一様分布による選択ではバイアスが生じる。バイアスの生じない最適な光源選択は、光源の発光放射輝度  $L$ 、双方向散乱分布関数  $B$ 、幾何項  $G$ 、そして可視関数  $V$  の積で表される大域照明計算の被積分関数  $f = LBGV$  を用いる場合である。ただし可視関数は遮蔽判定に追加の交差判定処理が必要なため、これを除いた寄与関数  $f = LBG$  に基づいて適応的に光源を選択し、図 4 右のように経路を構築する。この寄与関数は明示的にサンプリングが可能な形ではなく、分布そのものによる選択は困難である。そこでサンプリング可能な生成分布を用いた RIS を行い、寄与関数にできる限り近い標的関数により光源を選択する。得られた光源からの寄与を経路誘導の分布推定に利用し、推定済みの分布を用いて経路を構築する。

### 3.3 WRS と時空間的再利用

通常の RIS では、寄与関数  $p$  とは異なる生成分布  $q$  に基づいて複数の候補サンプル点を生成し、各サンプル点  $x_i$  に対して選択における重み  $w_i = p(x_i)/q(x_i)$  を割り当てる。重み  $w_i$  に対して定まる確率  $s_i = \frac{w_i}{\sum w_i}$  に応じてサンプル点を選択することで、元の寄与関数  $p$  に近い分布によるサンプル生成が実現できる。ただし通常の RIS をそのまま適用した場合、全候補サンプル点を保持する必要があるため、空間計算量が大きく効率的とはいえない。そこで本手法は、近年大域照明分野でよく用いられつつある **WRS (Weighted Reservoir Sampling)** を用いた Stream RIS を適用する。この手法では候補サンプル点全体の代わりに、現在の選択サンプル点  $y$  とその寄与  $p(y)$ 、重みの総和  $w_{sum}$ 、候補サンプル数  $M$  を保持する。WRS における更新およびサンプリングの疑似コードを **Algorithm 1** に示す。このとき選択サンプル点  $y$  は、 $x_1, x_2, \dots, x_{M-1}$  を候補サンプル点として RIS により生成されたサンプル点となり、 $y$  を用いたモンテカルロ推定は被積分関数  $f$  を用いて、 $\frac{f(y)}{p(y)} \cdot \frac{w_{sum}}{M}$  となる。候補サンプル数  $M$  について、 $M \rightarrow \infty$  において元の寄与関数によるサンプリングと一致することが知られているが、精度と時間計算量のトレードオフを考慮し、本手法の実装では  $M = 32$  としている。また生成分布  $q$  もモンテカルロ推定に影響を与えるが、寄与関数に比べて影響は小さい。そこで本研究では、RIS の生成分布として一様分布を用いている。具体的にはシーン上の光源モデル、光源モデル内

のポリゴン、ポリゴン上の一点を一様分布で選択し、選んだ点を候補サンプル点としている。

## 4. 実装

本研究では、NVIDIA OptiX [8] を用いた GPU レンダラを用いて C++ 言語で実装した。OptiX ではレイ生成および BVH の葉ノードとレイの交差状態に応じて実行される処理（交差、最近衝突、任意衝突）を記述するシェーダを組み合わせて記述する必要があるが、本研究では方向サンプリングと WRS による光源サンプリング処理を最近衝突シェーダ上で、画像の出力および SD-Tree の学習処理をレイ生成シェーダ上で実行されるように設計した。

## 5. 結果

実行環境として、CPU: AMD Ryzen Threadripper PRO 3995WX 64 コア 2.7 GHz, RAM: 128 GB, GPU: NVIDIA RTX A6000, パストレーシング, RIS, 経路誘導 (Müller), 提案手法の 4 手法を用いて、Bistro[4] シーンについて  $1,024 \times 1,024$  の画像をレンダリングし、リファレンス画像 (パストレーシング) 基準で **MAE (Mean Absolute Error)** 基準で比較した。ここで最大経路長を 4、経路誘導の分布学習に全サンプル数の 30 % を用いた。図 5 に 1,000 spp における出力画像を示した。

---

### Algorithm 1 Weighted Reservoir Sampling

---

```

1: struct Reservoir{
2:    $y \leftarrow 0$ 
3:    $M \leftarrow 0$ 
4:    $w_{sum} \leftarrow 0$ 
5:    $W \leftarrow 0$ 
6: }
7: function updateReservoir( $r, x_i, w_i$ )
8:    $r.w_{sum} \leftarrow r.w_{sum} + w_i$ 
9:    $r.M \leftarrow r.M + 1$ 
10:   if Random(0, 1) > ( $w_i/r.w_{sum}$ ) then  $r.y \leftarrow x_i$ 
11:   end if
12: end function
13: function RIS( $\mathbb{S}$ )
14:   Reservoir r
15:   for  $i = 1$  to  $M$  do
16:     generate  $x_i \sim q$ 
17:     update( $r, x_i, \frac{p(x_i)}{q(x_i)}$ )
18:   end for
19:    $r.W \leftarrow \frac{1}{p(r.y)} \cdot (\frac{1}{r.M} \cdot r.w_{sum})$ 
20:   return r
21: end function

```

---



(a) パストレーシング:  $11.2 \times 10^{-3}$

(b) 経路誘導:  $7.71 \times 10^{-3}$

(c) 提案手法:  $4.57 \times 10^{-3}$

図 5: 各手法の出力結果と MAE (1,000 spp)

### 5.1 サンプル基準での分散低減効果

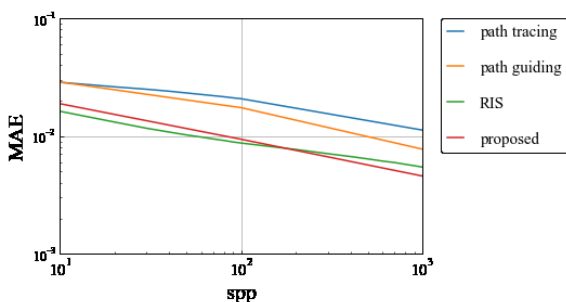


図 6: サンプル当たりの MAE (10,000,000 spp 基準)

図 6 に各手法のサンプル基準での比較を示した。提案手法は 1,000 spp において、パストレーシング、経路誘導に比べて大きく MAE を低減することが確認できた。また RIS の光源サンプリングの最適化により、サンプル数が少ない場合では、経路誘導・パストレーシングに比べて MAE が大きく低減できている。ただし学習分布の精度が高くないため、通常のリスキに比べて特性は悪化している。一方で、サンプル数が多い場合では、経路誘導に比べて低減効果が弱くなっている。これは WRS が外れ値の影響を強く受けるため、分布推定が進むにつれてバイアスになることや、SD-Tree の学習分布自体に光源への直接反射項が含まれているため、分布学習が進むにつれて SD-Tree 単体で高精度の光源サンプリングが行えるようになることが原因であると考えられる。

### 5.2 時間基準での分散低減効果

図 7 に各手法の時間基準での比較を示した。提案手法は、比較的サンプル数が少ない場合ではパストレーシングに比べて高い分散低減効果を示している。ただし、これは経路誘導によるものではなく、RIS による光源サンプリングの最適化によるものである。逆にサンプル数が増えるにつれて経路誘導による分散低減効果は高まるが、ノード数

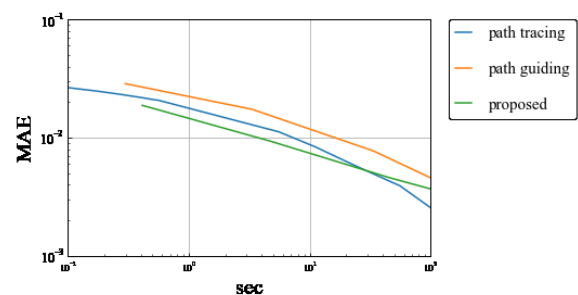


図 7: 時間当たりの MAE (10,000,000 spp 基準)

が増えるにつれて方向四分木のサンプリングに要する時間が增大するため、高サンプルでは分散低減効果が低いと考えられる。サンプリングに要する時間が增大する原因として GPU の並列処理に問題があると考えられる。本研究の実装で用いた Optix は GPPU 計算フレームワークである CUDA [6] を基盤としているが、その並列化モデルである SIMT (Single Instruction, Multiple Threads) は、並列単位で単一の命令しか実行できず、分岐処理が含まれていた場合両方の処理を実行したうえで、結果を捨てる形で処理されるため、計算性能が低下する。特に多数の分岐処理を伴う木構造の走査は分岐による低速化が起りやすく、本結果はその傾向が強く表れたものであると考えられる。また方向四分木の学習処理を並列実行する場合、複数のスレッドから同じ方向四分木の葉ノードに書き込みを行う可能性があるため、低速な不可分操作として実装している点も低速化の一因となっている。

## 6. まとめと今後の課題

本稿では、光源サンプリングの効率化により多光源問題における経路誘導の分布推定を改善する最適化手法を提案し、サンプル数当たりの分布収束効果を示した。評価実験からはサンプル数が少ない場合、分布収束効果は高い反面、サンプル数が多い場合では、分散低減効果が低くなる傾向

が得られた。また経路誘導のデータ構造として SD-Tree を利用したが、GPU 上では分岐処理による低速化の影響が確認できた。

### 6.1 WRS と時間的・空間的再利用

本研究では通常の WRS による Stream RIS を用いたが、関連研究で述べたようにワールド空間上で Reservoir の時間的・空間的再利用を行うことで大域照明における多光源照明計算を効率化する手法 [2] が提案されている。今後の研究として、提案手法と組み合わせることで分布収束効果を高めることを検討している。

### 6.2 経路誘導のデータ構造

提案手法を GPU 上で実装した場合、SD-Tree の走査に伴う分岐処理が課題となる。よって、経路誘導のデータ構造を現状の木構造から格子構造など GPU 上で高速に走査可能な構造へ変更することを検討している。

### 6.3 可視性の考慮

本研究では RIS の寄与関数として可視項を除いたものを用いている。しかし、可視性を無視したことで特定の光源以外遮蔽されているような状況では RIS の特性により分散が高まる可能性がある。そこで、各経路で得られた遮蔽情報を再利用し、寄与関数に可視性を部分的に考慮することで、この課題を解決できないかと考えている。

### 謝辞

本研究の一部は、科研費基盤研究 (A) 21H04916 の支援により実施された。

### 参考文献

- [1] Bitterli, B., Wyman, C., Pharr, M., Shirley, P., Lefohn, A. and Jarosz, W.: Spatiotemporal Reservoir Resampling for Real-Time Ray Tracing with Dynamic Direct Lighting, *ACM Transactions on Graphics*, Vol. 39, Article No. 4, DOI: 10.1145/3386569.3392481 (2020).
- [2] Boissé, G.: World-Space Spatiotemporal Reservoir Reuse for Ray-Traced Global Illumination, *SIGGRAPH Asia 2021 Technical Communications*, Article No. 4, DOI: 10.1145/3478512.3488613 (2021).
- [3] Kajiya, J. T.: The Rendering Equation, *SIGGRAPH Computer Graphics*, Vol. 20, No. 4, pp. 143–150, DOI: 10.1145/15886.15902 (1986).
- [4] Lumberyard, A.: Amazon Lumberyard Bistro, Open Research Content Archive (ORCA), <http://developer.nvidia.com/orca/amazon-lumberyard-bistro> (2017).
- [5] Müller, T., Gross, M. and Novák, J.: Practical Path Guiding for Efficient Light-Transport Simulation, *Computer Graphics Forum*, Vol. 36, No. 4, pp. 91–100, DOI: 10.1111/cgf.13227 (2017).
- [6] Nickolls, J., Buck, I., Garland, M. and Skadron, K.: Scalable Parallel Programming with CUDA: Is CUDA

- the Parallel Programming Model That Application Developers Have Been Waiting For?, *Queue*, Vol. 6, No. 2, pp. 40–53, DOI: 10.1145/1365490.1365500 (2008).
- [7] Ouyang, Y., Liu, S., Kettunen, M., Pharr, M. and Pantaleoni, J.: ReSTIR GI: Path Resampling for Real-Time Path Tracing, *Computer Graphics Forum*, Vol. 40, No. 8, pp. 17–29, DOI: <https://doi.org/10.1111/cgf.14378> (2021).
  - [8] Parker, S. G., Bigler, J., Dietrich, A., Friedrich, H., Hoberock, J., Luebke, D., McAllister, D., McGuire, M., Morley, K., Robison, A. and Stich, M.: OptiX: A General Purpose Ray Tracing Engine, *ACM Transactions on Graphics*, Vol. 29, Article No. 4, DOI: 10.1145/1778765.1778803 (2010).
  - [9] Talbot, J. F., Cline, D. and Egbert, P.: Importance Resampling for Global Illumination, *Proceedings of the Sixteenth Eurographics Conference on Rendering Techniques*, EGSR '05, Goslar, DEU, Eurographics Association, pp. 139–146, DOI: 10.2312/EGWR/EGSR05/139-146 (2005).
  - [10] Vorba, J., Hanika, J., Herholz, S., Müller, T., Křivánek, J. and Keller, A.: Path Guiding in Production, *ACM SIGGRAPH 2019 Course Notes*, DOI: 10.1145/3305366.3328091 (2019).