

多光源レンダリング法のための 画像生成の効率を考慮した可視関数の確率的評価

名畑 豪祐^{1,a)} 岩崎 慶¹

受付日 2018年9月5日, 採録日 2019年2月5日

概要: 本研究では, 多光源レンダリング法における可視関数を確率的に評価することで, 写実的な画像を効率的に生成する手法を提案する. 多光源レンダリング法では, 輝度の計算点に入射する直接光, 間接光を仮想的な点光源 (Virtual Point Light, VPL) からの寄与で近似する. VPL から輝度の計算点への寄与を計算するには, VPL と計算点間の可視関数を評価する必要があるが, 可視関数の評価は非常に高負荷であるため, 可視関数の評価が大きなボトルネックとなる. 提案法では, ある確率で可視関数の評価を省略することで, 高速に写実的な画像を生成する. ただし, 可視関数を確率的に評価することにより, 可視関数が新たな分散の要因となり, 可視関数の省略確率を大きくするほど分散が増加する. そこで, 増加する分散と画像生成時間の関係を考慮し, 画像生成の効率を最大にする確率を導出する. 提案法により, 正確に可視関数を評価した場合と比較して, 同時間レンダリングにおいて, より分散の小さい画像を生成することができる.

キーワード: 大域照明, 多光源レンダリング法, 可視関数

Efficient Probabilistic Visibility Evaluation for Many-light Rendering

KOSUKE NABATA^{1,a)} KEI IWASAKI¹

Received: September 5, 2018, Accepted: February 5, 2019

Abstract: In this paper, we propose an efficient probabilistic visibility evaluation method for many-light rendering. In many-light rendering, direct and indirect illumination is approximated by using many virtual point lights (VPLs). The computation of the reflected radiance using a VPL requires a computationally expensive visibility evaluation, which is the bottleneck for efficient rendering. Our method evaluates the visibility function with a certain probability, and skips the evaluation otherwise. Our probabilistic approach can reduce the number of computationally expensive visibility tests, but this yields additional variance due to the probabilistic approach. To solve this problem, we calculate the probability that is optimized for the trade-off between the computational efficiency and the additional variance. In equal-time rendering, our method can render less noise images than the previous methods with exact visibility evaluations.

Keywords: Global illumination, many-light rendering, visibility function

1. はじめに

CG の分野において, 写実的な画像の効率的な生成は重要な研究課題の 1 つである. 写実的な画像は, 映画, 工業デザイン, 建築設計, ゲームなど様々な分野で使用されて

いる. 写実的な画像を生成するには直接光だけでなく, 間接光も考慮する必要があるため, 計算コストが高く, 画像生成に多くの時間を要する.

近年, 効率的に写実的な画像を生成する手法の 1 つとして, 多光源レンダリング法が研究されている [1], [2]. 多光源レンダリング法は, あらかじめシーン中に大量の仮想的な点光源 (Virtual Point Light, VPL) を設置しておき, 出

¹ 和歌山大学
Wakayama University, Wakayama 640-8510, Japan
^{a)} s141044@wakayama-u.ac.jp

射輝度の計算点（以降シェーディング点と呼ぶ）では、入射する直接光、間接光を VPL からの寄与で近似する。このとき、設置する VPL の数が多いほど近似精度が向上するが、計算時間は VPL 数に比例して増加する。そのため、少量の VPL をサンプリングし、すべての VPL を使用したときの出射輝度を推定する手法が一般的である [3], [4], [5]。推定結果には誤差が含まれるが、寄与の大きい VPL を重点的にサンプリングするか、サンプル数を増やすことにより誤差を低減することができる。

VPL の寄与を計算するには、シェーディング点と VPL 間の可視判定が必要となる。可視判定を行うには、シェーディング点から VPL に向かうレイとシーン中のすべての物体で交差判定を行う必要があり、最先端の高速化データ構造を使用しても非常に計算コストが高い処理である。そのため、VPL の寄与計算において可視判定の計算時間が支配的となっており、同時間でより多く VPL をサンプリングするには、可視判定の高速化が重要である。

本研究では、VPL の寄与計算において可視判定を確率的に行うことで、可視判定回数を削減し効率的に写実的な画像を生成する手法を提案する。提案法では、ある確率で可視判定を省略し、それ以外では実際に可視判定を行う。以降、可視判定を省略する確率を省略確率と呼称する。省略確率を大きくするほど、可視判定回数を削減でき、同時間において、より多くのサンプルを計算できる。しかしながら、可視判定を確率的に行う場合、省略確率を高くするほど、可視判定結果に対する分散が増加し、生成される画像にノイズが生じるという問題がある。そこで、確率的評価による分散の増加量とサンプル数の関係を定式化し、与えられた時間内で分散を最小とする省略確率の導出を行う。本研究により、正確に可視関数を評価した場合と比較して、同時間レンダリングにおいて、よりノイズの少ない画像を生成することができる。また我々の実験では、同画質の画像を最大 1.55 倍高速に生成することができ、可視判定を確率的に行う先行研究 [6] と比較しても最大 1.3 倍の高速化を達成した。

2. 研究背景

2.1 先行研究

多光源レンダリング法による輝度計算において、少量の VPL をサンプリングし、輝度を推定する手法が多く提案されている。VPL の寄与は、VPL の放射強度、シェーディング点における BRDF、幾何項、可視関数の積で計算される。これら 4 つの項の積に比例して VPL のサンプリングを行うことで、分散の小さい高精度な推定を行うことができる。Wang らは、BRDF の重点的サンプリングを利用することで、放射強度・BRDF・幾何項の積に比例したサンプリング方法を提案した [3]。Georgiev らは、VPL の寄与を記録するキャッシュ点を使用することで、単純なシーン

において、VPL の寄与に比例したサンプリングを行う方法を提案した [4]。Wu らは、Wang らの手法に加えて、VPL とシェーディング点をクラスタリングすることで、複雑なシーンにおいても、VPL の寄与に比例したサンプリング方法を提案した [5]。これらの手法により、少ないサンプルでも高精度な輝度推定が可能となったが、可視判定の計算コストが高いシーンでは、依然として画像生成に時間がかかってしまう。特に Georgiev ら、Wu らの手法は、可視関数を考慮した VPL のサンプリングを行っているため、サンプリングされた VPL とシェーディング点間は互いに可視である確率が高く、つねに可視関数を評価するのは非効率である。

VPL の寄与計算自体を効率化する手法も多く提案されている。VPL の寄与計算では、可視判定の計算コストが特に高く、可視判定の効率化が重要である。Popov らは、経路空間を量子化し、可視判定結果を再利用することで、可視判定回数を削減する手法を提案した [7]。しかしながら、生成される画像には、経路空間の量子化による誤差が含まれる。Billen らは、可視判定の交差判定対象を確率的に選択することで、交差判定回数を削減する手法を提案した [8], [9]。しかしながら、同時間レンダリングにおいて分散が増加する結果となっており、実用的ではない。Veach は、画像生成の効率を考慮したロシアンルーレット法により、可視判定回数を削減する手法を提案した [6]。しかしながら、双方向パストレーシングにおいて導入された手法であるため、保守的な仮定を立てており大幅な高速化は期待できない。Tokuyoshi らは、寄与の小さい VPL を確率的にカリングすることで、VPL の寄与計算回数自体を削減する手法を提案した [10]。しかしながら、カリングを行う確率に画像生成の効率を考慮せず、ユーザが指定する許容誤差パラメータを基に計算しているため、高い効率で画像生成を行うにはパラメータ調整が必要である。

提案法は、Wu らの手法を基に、可視関数を確率的に評価することで、さらなる効率化を行うものである。可視関数の確率的評価は、Veach の提案した画像生成の効率を考慮したロシアンルーレット法を基にしており、可視関数の値に対する事前情報を用いることで、さらなる効率化を行う。Veach の手法では、寄与が小さい VPL に対して可視判定を省略する確率が高くなるのに対し、提案法では、寄与が小さい VPL だけでなく、高確率で可視または不可視の VPL に対しても可視判定を省略する確率を高く設定することができる。

2.2 多光源レンダリング法による輝度計算

多光源レンダリング法では、入射する直接光、間接光を、VPL の寄与で近似した以下の式で、シェーディング点 x から視点 x_v 方向に出射する放射輝度 L を計算する [2]。

$$L(x, x_v) = \sum_{i=1}^{N_{vpl}} I(y_i) f_r(x_v, x, y_i) G(x, y_i) V(x, y_i) \quad (1)$$

ここで、 N_{vpl} は VPL の数、 y_i は i 番目の VPL、 I は放射強度、 f_r は BRDF、 G は幾何項、 V は可視関数である。十分な近似精度を得るには、大量の VPL が必要であるが、計算時間は VPL の数に比例して増加する。そこで、少量の VPL をサンプリングし、式 (1) の値を推定する。

$$\begin{aligned} L(x, x_v) &\approx \hat{L}(x, x_v) \\ &= \frac{1}{N} \sum_{s=1}^N \frac{I(y_s) f_r(x_v, x, y_s) G(x, y_s) V(x, y_s)}{p(y_s | x)} \end{aligned} \quad (2)$$

ここで、 N はサンプリング回数、 y_s は s 回目の試行における VPL サンプル、 p は確率質量関数である。各試行はそれぞれ独立に行われ、 y_s のとりうる値は N_{vpl} 個の VPL のいずれかであり、 $y_s = y_i$ である確率は $p(y_i | x)$ である。寄与がある VPL に対して $p(y_i | x) > 0$ であれば、使用する確率質量関数は任意に選択でき、式 (2) の期待値は式 (1) と一致する。ただし、推定値の分散は確率質量関数に大きく依存し、以下の式で表される（導出の詳細は付録を参照）。

$$\begin{aligned} \text{var} \left[\hat{L}(x, x_v) \right] &= \frac{1}{N} \left\{ \sum_{i=1}^{N_{vpl}} \frac{I(y_i)^2 f_r(x_v, x, y_i)^2 G(x, y_i)^2 V(x, y_i)^2}{p(y_i | x)} \right. \\ &\quad \left. - L(x, x_v)^2 \right\} \end{aligned} \quad (3)$$

上式より、確率質量関数 p が $I \cdot f_r \cdot G \cdot V / L$ と類似するほど、つまり、VPL の寄与に比例するほど、分散を低減することができる。また、サンプル数 N を増やすことによっても分散を低減できることが分かる。そのため、サンプリングを効率的に行うことができれば、同時間のレンダリングにおいて、より多くのサンプルをとることができ、高精度な推定が可能となる。

本研究では、 $I(y_i) f_r(x_v, x, y_i) G(x, y_i)$ を VPL y_i の暫定的な寄与 t_i と定義する。また、表記簡略化のため $V(x, y_i)$ 、 $p(y_i | x)$ を、それぞれ v_i 、 p_i と表記する。

3. 提案法

提案法の処理の流れを図 1 に示す。基本的な処理の流れは、Wu らの手法 [5] と同様であり、最初に VPL とシェーディング点の生成を行い、それぞれをクラスタリングする (図 1(a))。次に、各シェーディング点クラスタで各 VPL クラスタに対して可視関数の平均値を推定する (図 1(b))。平均値の推定は、各クラスタの要素を数個サンプリングし可視関数を評価することで行う。各シェーディング点で出

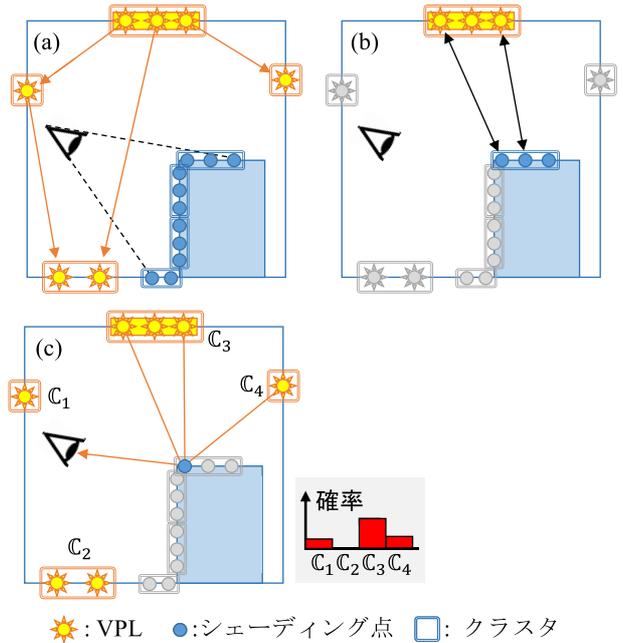


図 1 提案法の処理の流れ
Fig. 1 Overview of our method.

射輝度を計算する際は、推定した可視関数の平均値を用いて VPL クラスタに対する確率分布を構築する (図 1(c))。VPL のサンプリングは、VPL クラスタをサンプリングし、そのクラスタから VPL を一様にサンプリングすることで行う。このサンプリング方法では、可視の VPL を多く含む VPL クラスタがサンプリングされやすくなるため、そのクラスタからサンプリングされる VPL は可視である確率が高い。そして、提案法では、そのような VPL に対して可視関数の評価を高確率で省略するため、同時間でより多くのサンプルを使用することができ、分散の小さい推定が可能である。クラスタリングおよび確率分布の構築の詳細は文献 [5] を参照されたい。

3.1 可視関数の確率的評価

可視関数は 2 点間の可視性を表す関数で、シェーディング点 x と VPL y_i 間の可視関数 v_i は、 x と y_i が互いに可視の場合 1 を返し、それ以外は 0 を返す。可視関数の評価は計算コストが高く、VPL の寄与計算において可視関数評価の計算時間が支配的である。そこで提案法では、以下の式を用いて可視関数の値を推定する。

$$\tilde{V}(x, y_i) = \begin{cases} \frac{\alpha_i}{q_i} & \text{確率 } q_i \text{ で選択} \\ \frac{v_i - \alpha_i}{1 - q_i} & \text{それ以外} \end{cases} \quad (4)$$

ここで、 q_i はシェーディング点 x における VPL y_i の省略確率、 α_i はパラメータで任意の実数をとることができる。以下の式に示すように、提案法の可視関数の推定値の期待値 $\mathbb{E}[\tilde{V}(x, y_i)]$ は真値 v_i と一致する。

$$\mathbb{E}[\tilde{V}(x, y_i)] = \frac{\alpha_i}{q_i} q_i + \frac{v_i - \alpha_i}{1 - q_i} (1 - q_i) = v_i \quad (5)$$

提案法の可視関数の推定式では、省略確率 q_i で可視関数の評価を省略し、 α_i/q_i を推定値とする。そして、確率 $(1 - q_i)$ で実際に可視関数を評価し、 v_i に 0 か 1 の結果を代入した値を推定値とする。そのため、省略確率 q_i を大きくするほど、計算コストが高い可視関数の評価を省略でき、同時間においてより多くのサンプルを計算することができる。

提案法では、式 (2) の可視関数 V を推定式 \tilde{V} で置き換え、 s 回目の試行における VPL サンプル y_s の暫定的な寄与 t_s と確率質量関数の値 p_s を用いて以下の式で輝度計算を行う。

$$\tilde{L}(x, x_v) = \frac{1}{N} \sum_{s=1}^N \frac{t_s \tilde{V}(x, y_s)}{p_s} \quad (6)$$

可視関数の推定値の期待値が真値と一致するため、輝度の推定値の期待値も真値と一致する。

3.1.1 パラメータ α_i の計算

提案法では、可視関数を確率的に評価するため、可視関数の推定値には誤差が含まれる。そこで、パラメータ α_i には誤差が最小となる値を使用する。提案法の可視関数の推定値の分散は以下の式で表される。

$$\begin{aligned} \text{var}[\tilde{V}(x, y_i)] &= \left\{ \frac{\alpha_i^2}{q_i^2} q_i + \frac{(v_i - \alpha_i)^2}{(1 - q_i)^2} (1 - q_i) \right\} - v_i^2 \\ &= \frac{(v_i q_i - \alpha_i)^2}{q_i (1 - q_i)} \end{aligned} \quad (7)$$

上式より、 $\alpha_i = v_i q_i$ のとき、確率的評価による分散は 0 となる。しかしながら、可視関数の値 v_i は未知であるため可視関数の近似値が必要となる。提案法では、シェーディング点 x が所属するシェーディング点クラスタと、VPL y_i が所属する VPL クラスタ間の可視関数の平均値を v_i の近似値 r_i として用い、 $\alpha_i = r_i q_i$ とする。式 (4)、式 (7) に $\alpha_i = r_i q_i$ を代入した式は以下のとおりである。

$$\tilde{V}(x, y_i) = \begin{cases} r_i & \text{確率 } q_i \text{ で選択} \\ \frac{v_i - r_i q_i}{1 - q_i} & \text{それ以外} \end{cases} \quad (8)$$

$$\text{var}[\tilde{V}(x, y_i)] = \frac{q_i (v_i - r_i)^2}{1 - q_i} \quad (9)$$

3.1.2 省略確率 q_i の計算

可視関数を確率的に評価した場合、確率的評価による分散の増加量と、1 サンプルあたりの計算時間はトレードオフの関係にある。そこで推定値の分散はサンプル数 N に反比例して減少するという関係から、与えられた時間内で分散を最小とする省略確率 q_i を導出する。

1 サンプルの場合において、提案法の可視関数の推定式 (8) を用いたときの輝度の推定値の分散は以下の式で計

算される（導出の詳細は付録を参照）。

$$\begin{aligned} \text{var}[\tilde{L}^{(1)}(x, x_v)] \\ = \text{var}[\hat{L}^{(1)}(x, x_v)] + \sum_{j=1}^{N_{\text{vpl}}} \frac{t_j^2}{p_j^2} \frac{q_j (v_j - r_j)^2}{1 - q_j} p_j \end{aligned} \quad (10)$$

ここで、 $\text{var}[\hat{L}^{(1)}(x, x_v)]$ はサンプル数が 1 の場合の式 (3) である。また、第 2 項は可視関数の確率的評価による分散の増加量を表しており、可視関数を確率的に評価する場合、1 サンプルの場合では必ず分散が増加する。

次に、時間 T が与えられた下で計算できるサンプル数 N について考える。まず、シェーディング点の生成にかかる時間を t_x とすると、VPL サンプリングに割ける時間は $T - t_x$ となる。そして、VPL y_j の寄与計算にかかる時間は、可視関数を評価した場合と省略した場合の VPL の寄与計算時間をそれぞれ t_{skip} と t_{eval} とすると、平均で $t_{\text{skip}} q_j + t_{\text{eval}} (1 - q_j)$ となる。また、VPL によって可視関数評価の省略確率は異なるため、最終的な VPL の寄与計算時間は、各 VPL の平均寄与計算時間をさらに平均したものとなる。よって、時間 T が与えられた下で計算できるサンプル数 N は以下の式で表される。

$$\begin{aligned} N &= \frac{T - t_x}{\sum_{j=1}^{N_{\text{vpl}}} \{t_{\text{skip}} q_j + t_{\text{eval}} (1 - q_j)\} p_j} \\ &= \frac{T - t_x}{t_{\text{eval}} - t_{\text{diff}} \sum_{j=1}^{N_{\text{vpl}}} q_j p_j} \end{aligned} \quad (11)$$

ここで、 $t_{\text{diff}} = t_{\text{eval}} - t_{\text{skip}}$ とおき、 $\sum_{j=1}^{N_{\text{vpl}}} p_j = 1$ を用いた。分散はサンプル数 N に反比例して減少することから、式 (10)、式 (11) より、 N サンプルの場合において、輝度の推定値の分散は以下の式で表される。

$$\begin{aligned} &\left\{ \text{var}[\hat{L}^{(1)}(x, x_v)] + \sum_{j=1}^{N_{\text{vpl}}} \frac{t_j^2}{p_j^2} \frac{q_j (v_j - r_j)^2}{1 - q_j} p_j \right\} \\ &\times \left\{ \frac{t_{\text{eval}} - t_{\text{diff}} \sum_{j=1}^{N_{\text{vpl}}} q_j p_j}{T - t_x} \right\} \end{aligned} \quad (12)$$

なお、分散が最小となる省略確率 q_i の計算において定数は影響しないため、以降定数 $(T - t_x)^{-1}$ は省略する。VPL y_i に対する可視関数評価の省略確率を計算するために、式 (12) を q_i について微分し、結果を 0 とおくと以下の式が得られる。

$$\begin{aligned} &\frac{t_i^2}{p_i^2} \frac{(v_i - r_i)^2}{(1 - q_i)^2} \overbrace{\left\{ \frac{t_{\text{eval}}}{t_{\text{diff}}} - \sum_{j=1}^{N_{\text{vpl}}} q_j p_j \right\}}^{\approx \hat{t}} \\ &- \underbrace{\left\{ \text{var}[\hat{L}^{(1)}(x, x_v)] + \sum_{j=1}^{N_{\text{vpl}}} \frac{t_j^2}{p_j^2} \frac{q_j (v_j - r_j)^2}{1 - q_j} p_j \right\}}_{\approx \hat{\sigma}^2} = 0 \end{aligned} \quad (13)$$

上式を解けば、画像生成の効率を最大化する可視関数の省略確率 q_i が得られるが、2つの中括弧内の値の計算には他の VPL に対する省略確率が含まれるため、この式を解くのは困難である。そこで提案法では、現在の計算対象であるシェーディング点より以前に計算した近傍のシェーディング点の輝度計算情報を使用し、各中括弧内の値をそれぞれ定数 \hat{t} 、 $\hat{\sigma}^2$ で近似する（近似方法は 4.1 節で述べる）。

$$\frac{t_i^2 (v_i - r_i)^2}{p_i^2 (1 - q_i)^2} \hat{t} - \hat{\sigma}^2 = 0 \quad (14)$$

上式を q_i について解くことで、与えられた時間内で推定値の分散を最小とする可視関数評価の省略確率が得られる。

$$q_i = 1 - \sqrt{\frac{t_i^2 (v_i - r_i)^2}{p_i^2} \frac{\hat{t}}{\hat{\sigma}^2}} \quad (15)$$

上式では、パラメータ α_i の計算時と同様、未知の可視関数 v_i が含まれるため何らかの近似が必要となる。提案法では、 $(v_i - r_i)^2$ を平均値で近似する。提案法では可視関数の予測値 r_i としてクラスタ間の可視関数の平均値の近似値を用いているため、 $(v_i - r_i)^2$ の平均値はクラスタ間の可視関数の分散で近似できる。分散の定義より $\text{var}[v_i] = \mathbb{E}[v_i^2] - \mathbb{E}[v_i]^2$ であり、可視関数のとりうる値は 0 か 1 であるため $v_i^2 = v_i$ となること利用すると、 $\text{var}[v_i] = \mathbb{E}[v_i] - \mathbb{E}[v_i]^2$ となる。よって $\text{var}[v_i] \approx r_i - r_i^2$ であり、提案法では、式 (15) において $(v_i - r_i)^2$ を $r_i(1 - r_i)$ で置き換える。この近似では、 r_i が 1 か 0 に近いとき、 $r_i(1 - r_i)$ が小さくなるため省略確率が大きくなり、 r_i が 0.5 に近いとき、 $r_i(1 - r_i)$ が大きくなるため省略確率が小さくなる。つまりこの近似方法では、可視か不可視である確率が高いとき (r_i が 1 か 0 に近い)、高確率で可視判定を省略すべきであり、可視か不可視かが曖昧なとき (r_i が 0.5 に近い)、可視判定を正確に行うべきという直感どおりの特性を持つ。5 章で示すとおり、そのほかの近似方法との比較実験を行ったが、この方法が最も良好な結果が得られた。なお、式 (15) で可視関数評価の省略確率 q_i を計算すると、 q_i が負になる場合があるが、その場合は $q_i = 0$ とすればよい。

4. 実装の詳細

4.1 パラメータ \hat{t} 、 $\hat{\sigma}^2$ の計算

提案法の可視関数の確率的評価法において、VPL y_i に対する可視関数の省略確率の計算の際、他の VPL に依存する項 \hat{t} 、 $\hat{\sigma}^2$ の計算が必要となる。提案法では、現在の計算対象であるシェーディング点より以前に計算した近傍のシェーディング点における輝度計算情報を用いてモンテカルロ法により推定する。

$$\hat{t} = \frac{t_{eval}}{t_{diff}} - \frac{1}{N} \sum_{k=1}^N q_k \quad (16)$$

$$\hat{\sigma}^2 = \frac{1}{N} \sum_{k=1}^N \left(\frac{t_k \tilde{V}(x, y_k)}{p_k} \right)^2 - \left(\frac{1}{N} \sum_{k=1}^N \frac{t_k \tilde{V}(x, y_k)}{p_k} \right)^2 \quad (17)$$

ここで、 y_k は以前に計算した近傍のシェーディング点における k 番目の VPL サンプルであり、 q_k 、 t_k 、 p_k はそれぞれ VPL サンプル y_k に対する可視関数評価の省略確率、暫定的な寄与、確率質量関数の値である。 $\hat{\sigma}^2$ は、可視関数の確率的評価を用いた際の推定値の分散（式 (10)）を表すことから、式 (17) で表される標本分散で近似している。なお、最初に計算するシェーディング点は、可視関数評価の省略確率を 0 とした。

提案法では \hat{t} 、 $\hat{\sigma}^2$ を以前に計算した近傍のシェーディング点の情報を用いて近似するため、良い近似を行うには以前に計算した近傍のシェーディング点と現在のシェーディング点で VPL からの寄与が相関している必要があり、シェーディング点の計算順序が重要となる。シェーディング点クラスタ内のシェーディング点は、VPL からの寄与が相関していることが期待されるため、本研究では、シェーディング点クラスタを 1 つのレンダリング単位として、クラスタ内の全シェーディング点で輝度計算が終了してから、次のクラスタを処理する方法を採用した。

4.2 バイアスの回避

提案法では、可視関数評価の省略確率の計算式（式 (15)）中の $(v_i - r_i)^2$ の項を $r_i(1 - r_i)$ で近似する。そのため、 r_i が 0 か 1 のとき、可視関数評価の省略確率は 1 となり、実際の可視関数の値 v_i と近似値 r_i が異なる場合、推定値は真値に収束しない。そのため提案法では、 $(v_i - r_i)^2$ を $\max(\epsilon, r_i(1 - r_i))$ で置き換える。これにより、バイアスを回避できるだけでなく、高確率で可視、不可視の VPL でも、寄与が非常に大きい場合は可視関数評価を省略する確率が小さくなりロバスト性を向上させることができる。様々な値で実験した結果、下限値 ϵ は 0.1 程度に設定すれば良好な結果が得られた。

4.3 負値の可視関数

通常の可視関数のとりうる値は 0 か 1 だけであるため、輝度の推定値は必ず 0 以上となる。一方、提案法では可視関数の値を推定し、可視関数の推定値は 0 か 1 だけではなく、あらゆる値をとりうるため、負の値となることもある。VPL サンプル数が十分であれば、最終的な輝度の推定値は真値に収束するが、サンプル数が十分でない場合、輝度の推定値が負となることがある。そのため、ディスプレイに結果を出力する際は、負のピクセル値を 0 でクランプするといった、ピクセル値が負となる可能性を考慮したトーンマッピングが必要である。

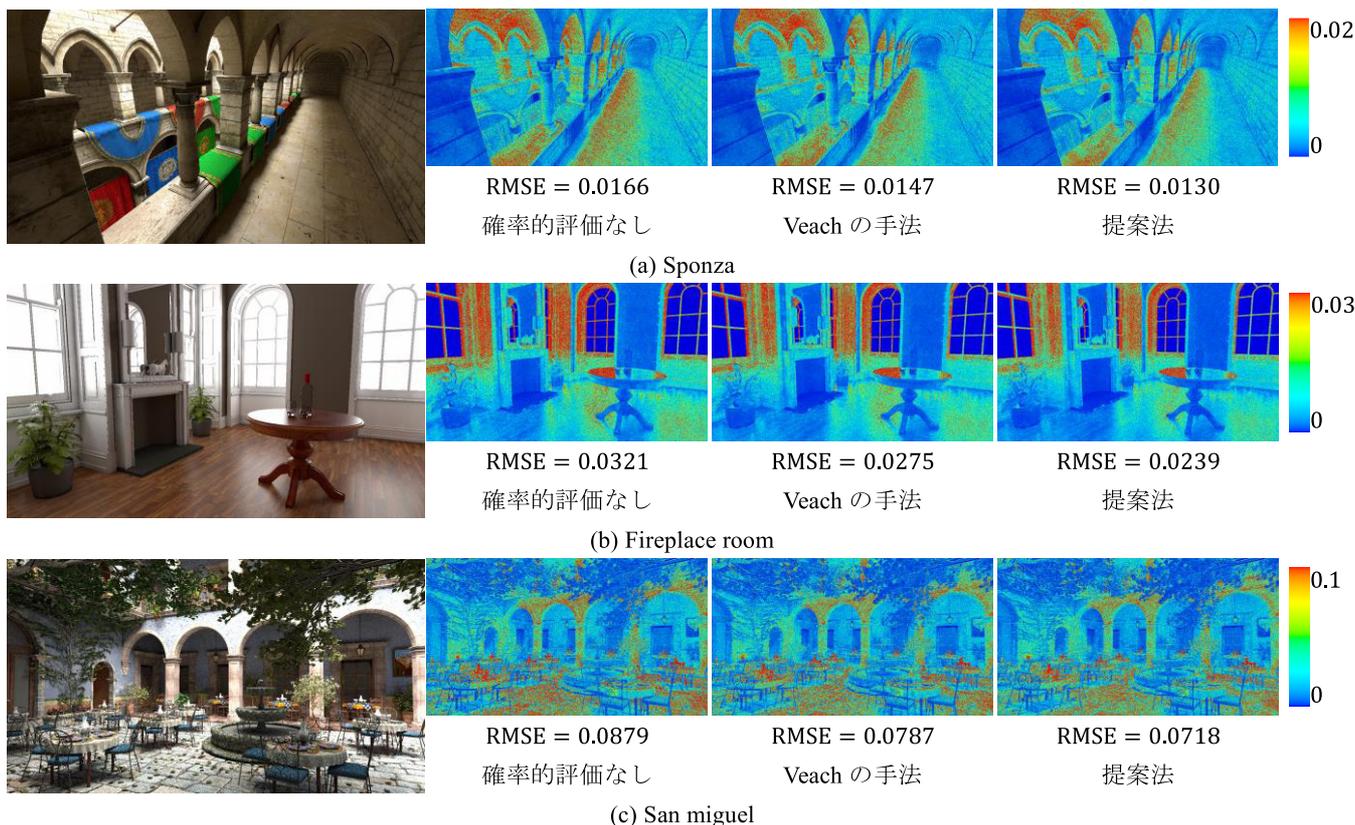


図 2 同時間 (60 秒) での誤差比較
 Fig. 2 Equal time (60s) comparison.

5. 実験結果

本章では、可視関数の確率的評価の有無による比較、先行研究である Veach の手法 [6] との比較により提案法の有効性を検証する。なお、実験は Intel Core i9-7980XE CPU, 32GB メモリを搭載した PC を使用し、マルチスレッドによる並列化を行っている。レイとシーンの交差判定に使用する高速化データ構造には、SIMD 命令に最適化された 4 分木の BVH [11] を使用している。提案法の可視関数の推定式のパラメータである t_{eval} と t_{skip} は、可視関数評価の省略確率を 0 とした場合、1 とした場合で低解像度の画像を生成し、その計算時間から設定した。また、他手法との比較の際に用いる誤差の指標には RMSE (Root Mean Square Error) を使用する。提案法と Veach の手法の主な違いは、Veach の手法ではつねに可視関数の近似値 r_i を 0 とし、式 (15) における $(v_i - r_i)^2$ を最大値の 1 で近似している点である。本実験におけるレンダリング画像の解像度は $1,280 \times 720$ である。

同時間レンダリングでの誤差の比較結果を図 2 に示す。各シーンの情報は表 1 に示すとおりである。なお、表 1 の Visibility Cluster の欄は、各手法で共通の処理である VPL・シェーディング点の生成とクラスタリング、確率分布の構築にかかった時間であり、サンプル数に依存しない計算時間である。実験結果より、提案法が最も誤差を低減

表 1 シーン情報

Table 1 Scene information.

	VPL 数	三角形数	Visibility Cluster	$\frac{t_{skip}}{t_{eval}}$
(a)	100K	262K	10 秒	0.15
(b)	200K	143K	12 秒	0.32
(c)	200K	10M	22 秒	0.05

できていることが分かる。特に、図 2(b) は光沢材質を含むシーン、図 2(c) はジオメトリが複雑であり可視関数の相関が小さいシーンであるが、このようなシーンでも提案法が最も誤差を低減できている。

次に図 2 の各シーンにおいて、提案法と Veach の手法で可視関数の確率的評価を行わない場合と同じ誤差の画像を生成するのに必要な時間を計測した結果を表 2 に示す。なお各行の上段と下段は、それぞれ Visibility Cluster の時間を含めた場合と含めない場合である。ノイズを視認できないほど小さくするには、より計算時間が必要となるが、Visibility Cluster の時間は表 1 に示す時間で固定であり、無視できるほど小さい割合となるため、下段に示す高速化率が実質のものである。結果より、Veach の手法では最大でも 1.26 倍の高速化率であるのに対して、提案法では 1.41 倍から 1.55 倍の高速化率が得られており、提案法の優位性が分かる。図 2(c) のシーンは可視判定の計算コストが高

表 2 同誤差画像の生成時間の比較 (括弧内は高速化率)

Table 2 Equal quality comparison (the number in each parenthesis indicates the acceleration ratio).

	確率的評価なし	Veach の手法	提案法
(a)	60秒	53 秒 (× 1.13)	43 秒 (× 1.40)
	50 秒	43 秒 (× 1.16)	33 秒 (× 1.52)
(b)	60秒	50 秒 (× 1.20)	43 秒 (× 1.40)
	48 秒	38 秒 (× 1.26)	31 秒 (× 1.55)
(c)	60秒	53 秒 (× 1.13)	49 秒 (× 1.22)
	38 秒	31 秒 (× 1.23)	27 秒 (× 1.41)

表 3 近似方法による比較

Table 3 Comparisons between ours and Veach’s conservative method.

	保守的な近似		提案法の近似	
	RMSE	平均省略率	RMSE	平均省略率
(a)	0.0145	39.6%	0.0130	60.7%
(b)	0.0263	45.6%	0.0239	61.6%
(c)	0.0770	40.5%	0.0718	55.3%

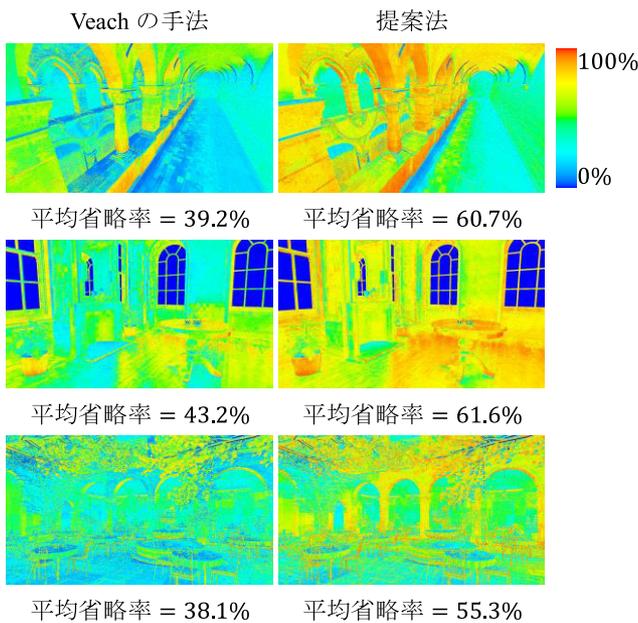


図 3 可視関数評価の省略率の可視化

Fig. 3 Visualization of skipping ratio of visibility evaluation.

いにもかかわらず、高速化率が小さいのは、ジオメトリが複雑であり正確な可視関数の近似ができていないためと思われる。

提案法と Veach の手法で、可視関数評価を省略した割合を可視化した画像を図 3 に示す。結果より、提案法が Veach の手法より高い省略確率が得られていることが分かる。特に遮蔽物の少ない領域であっても、Veach の手法では省略率が低いままであるのに対し、提案法では可視関数の事前情報を用いることで高い省略率が得られている。

可視関数評価の省略確率の計算式 (式 (15)) において行った近似について検証する。ここでの比較対象は Veach の手法と同様、保守的な近似を行った場合とする。可視関数の値 v_i は 0 か 1 であることから、 $(v_i - r_i)^2$ を $\max((1 - r_i)^2, (0 - r_i)^2)$ で置き換えたものが保守的な近似である。図 2 の各シーンで 60 秒かけてレンダリングしたときの RMSE と可視関数の平均省略率を表 3 に示す。結果より、保守的な近似より提案法の近似の方が、より多くの可視判定を省略しており、RMSE を低減できている。

図 2, 図 3 の Veach の手法の結果と保守的な近似の結果を比較した場合、両者は同程度の RMSE, 平均省略率であり、保守的な近似では、可視関数の事前情報使用の恩恵が受けられていないことが分かる。

最後に、可視判定の計算コストが高いシーンほど可視関数の確率的評価が効率的であることを明確に示すために、図 2(a) のシーンにおいて、レイとシーンの交差判定の高速化データ構造に単純な 2 分木の BVH [12] を用いた場合との比較を行う。2 分木の BVH を用いた場合、 t_{skip}/t_{eval} は 0.07 である。先ほどの同誤差画像の生成時間の比較と同様に、確率的評価なしで 60 秒かけてレンダリングした結果と同じ誤差の画像を提案法で生成するのにかった時間を計測すると、提案法は 40.7 秒であり、高速化率は 1.4 倍から 1.47 倍に向上し、Visibility Cluster の時間を除けば、高速化率は 1.52 倍から 1.63 倍に向上した。このことから、提案法は可視判定の計算コストが高いシーンほど有用である。

6. 制限事項

提案法は、可視関数の事前情報を用いることで Veach の手法を改良した手法である。提案法の可視関数評価の省略確率は、事前情報の精度がある程度正確であることを仮定しているため、この仮定を満たさない場合は、同時間において誤差が増加する可能性がある。また、クラスタ内の各シェーディング点は VPL からの寄与が相関していることも仮定しているため (4.1 節参照)、この仮定を満たさない場合も、同時間において誤差が増加する要因になりうる。本研究では、Wu らの手法 [5] を用いてクラスタリングを行っており、この手法では可視関数の相関のみを考慮しているため、提案法に適したクラスタリング方法の考案が必要である。

7. まとめ

本研究では、確率的な手法を用いて可視判定回数を削減することで、写実的な画像を効率的に生成する手法を提案した。提案法では、可視関数を確率的に評価するため、可視関数が新たな分散の要因となるが、可視関数評価を省略することによる計算コストの減少量と、可視関数の確率的評価による分散の増加量の関係を考慮した可視関数評価の

省略確率を用いることで、与えられた時間内でより誤差の少ない画像を生成することができる。提案法の可視関数の確率的評価法を用いることで、従来の可視関数の確率的評価法と比較し、より効率的にレンダリングを行えることを示した。

今後の研究課題としては、双方向パストレーシングへの応用があげられる。特に、近年 Popov らが提案した手法 [13] は、可視関数を考慮して VPL の重点的サンプリングを行う Georgiev らの手法 [4] を双方向パストレーシングに応用した手法であるため、提案法に必要な可視関数の予測値を容易に得ることができ、提案法の応用は比較的容易であると思われる。

参考文献

- [1] Dachsbacher, C., Krivanek, J., Hasan, M., Arbree, A., Walter, B. and Novak, J.: Scalable Realistic Rendering with Many-light Methods, *Eurographics STAR* (2013).
- [2] Keller, A.: Instant Radiosity, *Proc. SIGGRAPH*, pp.49–56 (1997).
- [3] Wang, R. and Akerlund, O.: Bidirectional Importance Sampling for Unstructured Direct Illumination, *Computer Graphics Forum*, Vol.28, No.2, pp.269–278 (2009).
- [4] Georgiev, I., Krivanek, J., Popov, S. and Slusallek, P.: Importance Caching for Complex Illumination, *Computer Graphics Forum*, Vol.31, No.2, pp.701–710 (2012).
- [5] Wu, Y.T. and Chuang, Y.Y.: VisibilityCluster: Average Directional Visibility for Many-Light Rendering, *IEEE Trans. Visualization and Computer Graphics*, Vol.19, No.9, pp.1566–1578 (2013).
- [6] Veach, E.: *Robust Monte Carlo Methods for Light Transport Simulation*, PhD Thesis, Stanford University (1998).
- [7] Popov, S., Georgiev, I., Slusallek, P. and Dachsbacher, C.: Adaptive Quantization Visibility Caching, *Computer Graphics Forum*, Vol.32, No.2, pp.399–408 (2013).
- [8] Billen, N., Engelen, B., Lagae, A. and Dutre, P.: Probabilistic Visibility Evaluation for Direct Illumination, *Computer Graphics Forum*, Vol.32, No.4, pp.39–47 (2013).
- [9] Billen, N., Lagae, A. and Dutre, P.: Probabilistic Visibility Evaluation using Geometry Proxies, *Computer Graphics Forum*, Vol.33, No.4, pp.143–152 (2014).
- [10] Tokuyoshi, Y. and Harada, T.: Stochastic Light Culling, *Journal of Computer Graphics Techniques*, Vol.5, No.1, pp.35–60 (2016).
- [11] Dammertz, H., Hanika, J. and Keller, A.: Shallow Bounding Volume Hierarchies for Fast SIMD Ray Tracing of Incoherent Rays, *Computer Graphics Forum*, Vol.27, No.4, pp.1225–1233 (2008).
- [12] Wald, I.: On Fast Construction of SAH-based Bounding Volume Hierarchies, *Proc. Eurographics/IEEE Symposium on Interactive Ray Tracing*, pp.33–40 (2007).
- [13] Popov, S., Ramamoorthi, R., Durand, F. and Drettakis, G.: Probabilistic Connections for Bidirectional Path Tracing, *Computer Graphics Forum*, Vol.34, No.4, pp.75–86 (2015).

付 録

輝度推定値の分散 (式 (3)) の導出

輝度推定値 (式 (2)) の分散の導出方法について説明する。分散の定義より、独立同一分布に従う確率変数 X_1, \dots, X_N において、これらの平均 $Z = (X_1 + \dots + X_N)/N$ の分散 $\text{var}[Z]$ は以下の式で計算される。

$$\begin{aligned} \text{var}[Z] &= \text{var}\left[\frac{1}{N} \sum_{s=1}^N X_s\right] \\ &= \frac{1}{N^2} \text{var}\left[\sum_{s=1}^N X_s\right] \\ &= \frac{1}{N} \text{var}[X_1] \\ &= \frac{1}{N} (E[X_1^2] - E[X_1]^2) \end{aligned} \quad (\text{A.1})$$

提案法の場合、 X_s は $t_s v_s / p_s$ に相当し、 N_{vpl} 個の VPL ごとの値をとる離散的確率変数であることに注意すると、期待値 $\mathbb{E}[t_s^2 v_s^2 / p_s^2]$, $\mathbb{E}[t_s v_s / p_s]$ は以下の式で計算される。

$$\begin{aligned} \mathbb{E}\left[\frac{t_s^2 v_s^2}{p_s^2}\right] &= \sum_{i=1}^{N_{\text{vpl}}} \frac{t_i^2 v_i^2}{p_i^2} p_i = \sum_{i=1}^{N_{\text{vpl}}} \frac{t_i^2 v_i^2}{p_i} \\ \mathbb{E}\left[\frac{t_s v_s}{p_s}\right] &= \sum_{i=1}^{N_{\text{vpl}}} \frac{t_i v_i}{p_i} p_i = \sum_{i=1}^{N_{\text{vpl}}} t_i v_i = L(x, x_v) \end{aligned} \quad (\text{A.2})$$

式 (A.1)、式 (A.2) より、式 (3) の輝度推定値の分散が得られる。

輝度推定値の分散 (式 (10)) の導出

可視関数の確率的評価を行う場合の輝度推定値の分散の導出方法について説明する。分散の定義より、確率変数 X , Y において、 Y が X に依存するとき、 XY の分散 $\text{var}[XY]$ は以下の式で計算される。

$$\begin{aligned} \text{var}[XY] &= \mathbb{E}[X^2 Y^2] - \mathbb{E}[XY]^2 \\ &= \mathbb{E}[\mathbb{E}[Y^2 | X] X^2] - \mathbb{E}[\mathbb{E}[Y | X] | X]^2 \\ &= \mathbb{E}[\mathbb{E}[Y^2 | X] X^2] - \mathbb{E}[\mathbb{E}[Y | X] X]^2 \\ &\quad + \mathbb{E}[\mathbb{E}[Y | X]^2 X^2] - \mathbb{E}[\mathbb{E}[Y | X]^2 X^2] \\ &= (\mathbb{E}[\mathbb{E}[Y^2 | X] X^2] - \mathbb{E}[\mathbb{E}[Y | X]^2 X^2]) \\ &\quad + (\mathbb{E}[\mathbb{E}[Y | X]^2 X^2] - \mathbb{E}[\mathbb{E}[Y | X] X]^2) \\ &= \mathbb{E}[\text{var}[Y | X] X^2] + \text{var}[\mathbb{E}[Y | X] X] \end{aligned} \quad (\text{A.3})$$

提案法の場合、 X は t_s / p_s に相当し、 Y は \tilde{V} に相当する。このとき $\mathbb{E}[Y | X]$ は可視関数の推定値の期待値であり式 (5) より $\mathbb{E}[Y | X] = v_s$ である。よって、 $\text{var}[\mathbb{E}[Y | X] X] = \text{var}[t_s v_s / p_s] = \text{var}[\hat{L}^{(1)}(x, x_v)]$ であり、式 (10) の第 1 項である。一方、 $\text{var}[Y | X]$ は可視関数の推定値の分散であり式 (9) で表される。これを $\mathbb{E}[\text{var}[Y | X] X^2]$

に代入すれば式 (10) の第 2 項が得られる.



名畑 豪祐 (正会員)

2013 年和歌山大学システム工学部情報通信システム学科卒業. 2015 年同大学大学院システム工学研究科システム工学専攻博士前期課程修了. 2019 年同大学院システム工学研究科システム工学専攻博士後期課程修了.



岩崎 慶 (正会員)

1999 年東京大学理学部情報科学科卒業. 2001 年同大学大学院新領域創成科学研究科複雑理工学専攻博士前期課程修了. 2004 年同大学院博士後期課程修了. 同年和歌山大学システム工学部情報通信システム学科助手. 2007 年同講師. 2009 年同准教授. 主としてコンピュータグラフィックスに関する研究に従事. 科学博士.