

# VPL サンプリングのための効率的な可視関数の確率的評価

## Efficient Probabilistic Visibility Evaluation for VPL Sampling

名畑 豪祐\* 岩崎 慶\* 土橋 宜典† 西田 友是‡

Kosuke NABATA\* Kei IWASAKI\* Yoshinori DOBASHI† and Tomoyuki NISHITA‡

\*和歌山大学

\*Wakayama University

†北海道大学/JST CREST

†Hokkaido University/JST CREST

‡広島修道大学/UEI リサーチ

‡Hiroshima Shudo University/UEI Research

### 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一つである。写実的な画像は、映画、産業デザイン、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の寄与だけでなく、間接光の寄与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として、多光源レンダリング法がよく研究されている[1,2]。この手法では、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、輝度を計算するシェーディング点では、入射する直接光、間接光の寄与を VPL 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、VPL 全体の寄与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、可視判定のため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。そのため、可視判定を省略する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ため、可視判定を省略する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。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考慮し、与えられた時間内で分散を最小とする確率の導出を行う。また、VPL の寄与の位置・方向に関するコヒーレンスを利用し、寄与の大きい VPL を重点的にサンプリングする手法を提案する。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、確率的に可視判定を行う先行研究[9]と比較して、2.5 倍の可視判定削減率の向上を達成した。

### 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、放射強度、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た 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 に比例した入射方向の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、シェーディング点と VPL のクラスタ間で可視判定結果の平均値を使用することで、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]。しかしながら、これらの手法は、VPL とシェーディング点の可視判定の高速化を行っていない。特に[4][5]は、可視関数を考慮した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、確率的評価によって可視判定回数の削減が期待できる。

計算コストが高い可視判定の効率化を行う手法も多く提案されている。Popovらは、類似する経路間で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[6]。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[7][8]。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[9]。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、低効率である。名畑らは、可視判定結果を記録するキャッシュ点を使用することで、可視判定回数を削減する手法を提案した[10]。しかしながら、すべてのVPLに対して寄与計算を行っているため、計算コストが高いという問題がある。

提案法は、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、多光源レンダリング法に応用し、VPLの寄与の一部をキャッシングすることで、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、近傍で可視判定結果が類似している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、間接光の寄与を、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 $x_v$ 方向に反射する放射輝度 $L_r$ を計算する[2]。

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

ここで、 $N_{vpl}$ はVPLの数、 $y_i$ は $i$ 番目のVPL、 $I$ は放射強度<sup>1</sup>、 $f_r$ 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。

$$L_r(x, x_v) \approx \hat{L}_r(x, x_v) = \frac{1}{N} \sum_{s=1}^N \frac{I(y_s, x) f_r(y_s, x, x_v) G(x, y_s) V(x, y_s)}{p(y_s)} \quad (2)$$

ここで、 $N$ はVPLのサンプル数、 $y_s$ は $s$ 回目の試行におけるVPLサンプルであり、確率関数 $p$ に従いサンプリングされる。使用する確率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関数に大きく依存し、効率的な推定

のためには確率関数の選択が重要となる。

理想的な確率関数の選択のために、式(2)の分散を考える。式(2)の分散は以下の式で計算される。

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

式(3)より、確率関数 $p$ が $I \cdot f_r \cdot G \cdot V / L_r$ 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。

確率関数の選択の他に分散を低減する方法としては、多数のサンプルを用いて推定することが挙げられる。式(3)より、推定値の分散はサンプル数 $N$ に反比例して減少する。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、低分散の推定が可能となる。

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

## 3 提案法

本研究では、多光源レンダリング法において可視関数を確率的に評価することで効率的に写実的な画像を生成する手法を提案する。また、VPL・シェーディング点のクラスタを使用したVPLの重点的サンプリング法を提案する。提案法は、前処理とレンダリング処理から成る。前処理ではVPL・シェーディング点のクラスタを使用してVPLの寄与に比例した確率分布の構築を行う。レンダリング処理では、構築した確率分布を使用し、輝度計算を行う。このとき、計算時間の大部分を占める可視関数の評価を確率的に行うことで、同時間内でより多くのサンプルを使用することができる。

### 3.1 確率分布の構築

VPLサンプリングにより、反射する放射輝度を高精度で推定するには、VPLの寄与に比例した確率分布を用いる必要がある。VPLの寄与は、放射強度、BRDF、幾何項、可視関数の積で計算される。これらの項は、VPLとシェーディング点の両方に依存するが、放射強度については、VPLの光の放射特性を完全拡散で近似することで、VPLのみに依存する項として扱うことができる[1]。そのため、BRDF、幾何項、可視関数の積の高精度な近似値が得られれば、おおよそVPLの寄与に比例した確率分布を構築することができる。

位置と法線が類似するシェーディング点と、位置と光の放射方向が類似するVPL間では、BRDF、幾何項、可視関数の積も類似する。そこで提案法では、VPL、シェーディング点を、位置

<sup>1</sup> VPL生成点に入射する光子放射束とBRDFの積で計算される。

と法線（VPLの場合は光の放射方向）に基づきクラスタリング [11]する。そして、各VPLクラスタと各シェーディング点クラスタ間で、少量の要素（10個程度）をサンプリングし、BRDF、幾何項、可視関数の積の平均値を計算する。各シェーディング点クラスタでは、この平均値を使用してVPLとシェーディング点間の寄与を近似し、確率分布を構築する。

しかしながら、VPLとシェーディング点の位置関係などによって、BRDF、幾何項、可視関数の積のよい近似値が得られない場合がある。そこで提案法では、サンプリングした要素の分散とVPLクラスタの要素数の積を誤差の指標として、誤差の大きいVPLクラスタを分割する。そして、分割したVPLクラスタとシェーディング点クラスタで、再度、BRDF、幾何項、可視関数の積の平均値を計算する。以上の分割処理を、誤差の指標が閾値以下か、VPLクラスタ数が指定数に達するまで行う。

### 3.2 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。

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

また、可視関数の推定値の期待値 $E[\tilde{V}(x, y_i)]$ は以下の式で計算される。

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

ここで、パラメータ $\alpha_i$ は任意の実数であり、推定値の期待値は真値と一致する。提案法の可視関数の推定式では、確率 $q_i$ で可視関数の評価を省略し、 $\alpha_i/q_i$ 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、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、可視関数を推定式で置き換えた以下の式で輝度計算を行う。

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

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。

#### 3.2.1 パラメータ $\alpha_i$ 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 $\alpha_i$ 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計算される。

$$\text{var}[\tilde{V}(x, y_i)] = \frac{(v_i q_i - \alpha_i)^2}{q_i (1 - q_i)} \quad (7)$$

式(7)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、各シェーディング点クラスタと各VPLクラスタ間で可視関数値の平均を計算しておき、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、VPL $y_i$ が所属するVPLクラスタ間の可視関数値の平均を $v_i$ 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となる。

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

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

#### 3.2.2 確率 $q_i$ の計算

可視関数を確率的に評価した場合、可視関数の推定値の分散の増加量と、1サンプル当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。

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

ここで、 $\text{var}[\tilde{L}_r^{(1)}(x, x_v)]$ 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 $T$ が与えられたときに、計算できるサンプル数は以下の式で計算される。

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

ここで、 $t_x$ はシェーディング点の生成にかかる時間、 $t_{\text{skip}}$ と $t_{\text{eval}}$ はそれぞれ可視関数評価を省略した場合、評価した場合のVPLの寄与計算にかかる時間であり、 $t_{\text{diff}} = t_{\text{eval}} - t_{\text{skip}}$ とおい。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。

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

なお、分散が最小となる確率 $q_i$ 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 $q_i$ について偏微分

し、結果を0とおくと以下の式が得られる。

$$\frac{t_i^2 (v_i - r_i)^2}{p_i^2 (1 - q_i)^2} \left\{ \frac{t_{eval}}{t_{diff}} - \sum_{j=1}^{N_{vpl}} q_j p_j \right\} - \left\{ \text{var}[\hat{r}_r^{(3)}(x, x_v)] + \sum_{j=1}^{N_{vpl}} \frac{t_j^2 q_j (v_j - r_j)^2}{p_j^2 (1 - q_j)} p_j \right\} = 0 \quad (13)$$

式(13)において、2つの中括弧内の値の計算には、他のVPLに対する可視関数評価の省略確率が含まれるため、VPL $y_i$ に対する可視関数評価の省略確率の計算は困難である。そのため提案法では、シェーディング点において輝度計算を開始する直前のVPLサンプルの情報を使用し、各中括弧内の値をそれぞれ $\hat{t}, \hat{\sigma}^2$ で近似する。(近似方法は4.1節で述べる。)

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

式(14)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。

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

式(15)では、パラメータ $a_i$ の計算時と同様、未知の可視関数 $v_i$ が含まれるため $v_i$ の近似が必要となる。平方根内に含まれる $(v_i - r_i)^2$ について考えると、 $r_i$ はシェーディング点クラスタとVPLクラスタ間の可視関数の平均であることから、この項は平均値 $r_i$ と $v_i$ の差の2乗である。そのためこの項は、シェーディング点クラスタとVPLクラスタ間の可視関数の分散によってよく近似できる。なお、可視関数の値は0か1であることから、シェーディング点クラスタとVPLクラスタ間の可視関数の分散は $r_i - r_i^2$ で計算できる。また、式(15)において、可視関数評価の省略確率 $q_i$ が負になる場合があるが、その場合は $q_i = 0$ として $\hat{V}(x, y_i)$ を評価すればよい。

## 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、シェーディング点で輝度計算を開始する直前の $M$ 個のVPLサンプルの情報を用いて推定する。

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

$$\begin{aligned} \hat{\sigma}^2 &= \frac{1}{M} \sum_{k=1}^M \left( \frac{t_k \hat{V}(x, y_k)}{p_k} \right)^2 - \left( \frac{1}{M} \sum_{k=1}^M \frac{t_k \hat{V}(x, y_k)}{p_k} \right)^2 \\ &= \frac{1}{M} \sum_{k=1}^M c_k^2 - \left( \frac{1}{M} \sum_{k=1}^M c_k \right)^2 \end{aligned} \quad (17)$$

ここで、 $y_k$ は直前の $k$ 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、サンプリング確率である。式(16)、式(17)では、直

前の $M$ 個のサンプルを使用した $q_k, c_k, c_k^2$ の平均値の計算が必要となるが、移動平均アルゴリズムを用いることで効率的に計算できる。なお本研究では、使用する直前のサンプル数 $M$ はVPLサンプル数とした。また、直前にVPLサンプルが存在しない場合は、可視関数評価の省略確率を0とした。

近似値 $\hat{t}, \hat{\sigma}^2$ を直前のVPLサンプルから計算するため、良い近似を行うには直前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

### 4.2.1 VPLサンプリング

提案法では、VPLクラスタとシェーディング点クラスタ間で、BRDF、幾何項、可視関数の積の平均値をサンプリングにより推定している。そのため、実際には0より大きい値を持つVPLとシェーディング点の組があるにも関わらず、平均値が0と推定された場合、その組のサンプリング確率は0となるため、推定値は真値に収束しない。

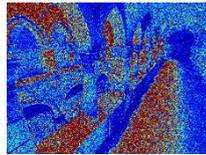
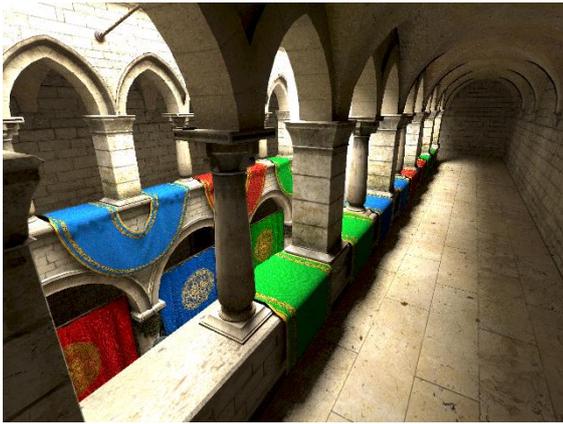
そのため提案法では、平均値を使用した確率分布と、すべてのVPLを一様にサンプリングする一様分布を用いて輝度計算を行う。なお、実際に2つの確率分布を用いて輝度計算を行う必要はなく、平均値を使用した確率分布と一様分布を平均した1つの確率分布を構築し、輝度計算を行えばよい。

### 4.2.2 可視関数評価の省略確率

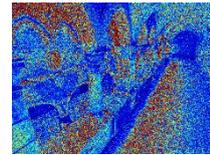
提案法では、可視関数評価の省略確率の計算式(式(15))中の $(v_i - r_i)^2$ の項を、シェーディング点クラスタとVPLクラスタ間の可視関数の分散で近似する。分散が0の場合、可視関数評価の省略確率は1となるため、実際の可視関数の値と、シェーディング点クラスタとVPLクラスタ間の可視関数の平均値が異なる場合、推定値は真値に収束しない。そのため、シェーディング点クラスタとVPLクラスタ間の可視関数の分散を、適当な値で下限をクランプした値で $(v_i - r_i)^2$ の項を近似する。下限値は0.1程度に設定すれば良好な結果が得られている。

## 4.3 負値の可視関数

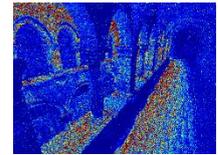
提案法による可視関数の推定値は0か1ではなく、負の値も取りうるため、放射輝度の推定値が負となることがある。現在の実装では、各ピクセルに対応する放射輝度値は負の値のまま保持しておき、最終的に画像として保存する際に、0でクランプし



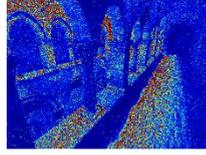
RMSE = 0.100021  
(a) 一様サンプリング



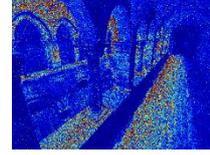
RMSE = 0.068055  
(b) パワーサンプリング



RMSE = 0.051957  
(c) 提案法(可視関数評価)



RMSE = 0.047964  
(d) Veach の手法



RMSE = 0.041961  
(e) 提案法(確率的評価)

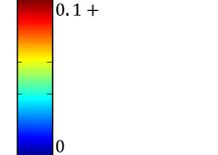
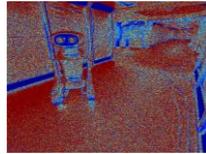
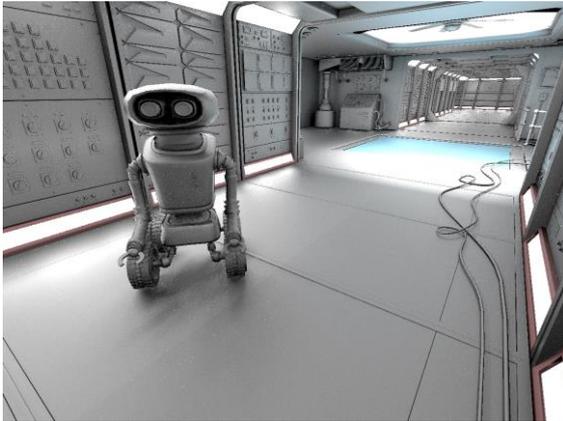
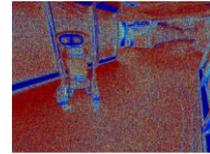


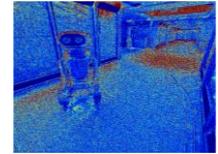
図1 同時間 (150秒) レンダリング結果の誤差比較



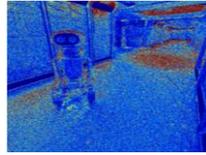
RMSE = 0.397117  
(a) 一様サンプリング



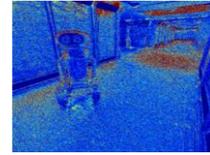
RMSE = 0.356253  
(b) パワーサンプリング



RMSE = 0.228430  
(c) 提案法(可視関数評価)



RMSE = 0.201967  
(d) Veach の手法



RMSE = 0.190752  
(e) 提案法(確率的評価)

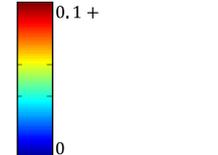
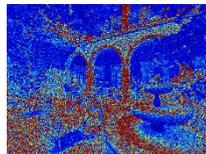
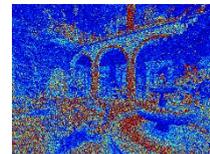


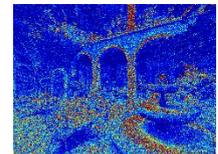
図2 同時間 (160秒) レンダリング結果の誤差比較



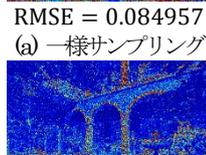
RMSE = 0.084957  
(a) 一様サンプリング



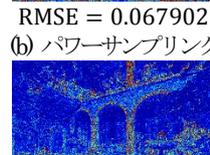
RMSE = 0.067902  
(b) パワーサンプリング



RMSE = 0.048753  
(c) 提案法(可視関数評価)



RMSE = 0.046487  
(d) Veach の手法



RMSE = 0.044169  
(e) 提案法(確率的評価)

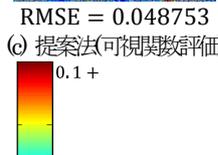


図3 同時間 (400秒) レンダリング結果の誤差比較

表1 シーン情報

	解像度	三角形数	VPL 数	シェーディング点 クラスタ数	VPL クラスタ数	$\frac{t_{skip}}{t_{eval}}$
図1	1,280×960	279K	100K	3,338	1,417	0.25
図2	1,280×960	787K	250K	5,513	2,603	0.34
図3	1,280×960	10M	500K	6,210	2,318	0.15

ている。そのため、0以下の値がある場合、生成された画像は統計的に偏りのあるものとなる。しかしながら、可視関数の推定値自体は不偏推定量であり、サンプル数を増やすことで真値へ収束するため、問題はないと思われる。

## 5 実験結果

実験には、Intel Core i7-4790K 4.00GHz CPU, 16.0GB 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、

表 2 同誤差画像の生成時間の比較 (括弧内は高速化率)

	提案法 (可視関数評価)	Veach の手法	提案法
図 1(c)	150 秒	134 秒(×1.12)	107 秒(×1.40)
図 2(c)	160 秒	147 秒(×1.09)	133 秒(×1.20)
図 3(c)	400 秒	371 秒(×1.08)	345 秒(×1.16)

SIMD 命令に最適化された 4 分木の BVH[12]を使用している。なお、可視関数の推定式のパラメータである  $t_{eval}$  と  $t_{skip}$  は、可視関数の省略確率を 0 とした場合、1 とした場合で時間を事前に計測し、その平均値を用いた。また、他手法との比較の際に用いる推定値の誤差の指標には、RMSE (Root Mean Square Error)を使用する。比較対象は、確率関数が全 VPL で一定の一樣サンプリング、VPL の放射強度に比例した確率分布のパワーサンプリング、先行研究である Veach の手法とする。また、可視関数の確率的評価法の有効性を示すために、提案法の確率分布を用い、可視関数は正確に評価する場合との比較も行う。なお、Veach の手法で用いる確率分布は提案法のものとし、可視関数の確率的評価法のみが異なるようにしている。

同時間レンダリングでの推定値の誤差の比較結果を図 1 から図 3 に示す。なお、図 2 のシーンは適度な光沢反射材質を適用している。各シーンの情報は表 1 に示す通りである。図(a)(b)(c)より、提案法の確率分布を用いたサンプリング法が、一樣サンプリング、パワーサンプリングより RMSE を減少させていることが分かる。また、図(c)(d)(e)より、Veach の提案した可視関数の確率的評価法では、RMSE の減少量がわずかであるのに対し、提案法の可視関数の確率的評価法では、RMSE をより減少させることができている。

提案法と Veach の手法で、図 1 から図 3 の(c)と同誤差の画像生成に必要な計算時間を計測した結果を表 2 に示す。結果より、Veach の手法では、最大でも 1.12 倍の高速化率であるのに対し、提案法では 1.16 倍から 1.40 倍の高速化率が得られており、提案法の優位性が分かる。

提案法と Veach の手法で、可視関数の確率的評価法による、可視関数の評価回数の削減率の比較を行う。図 4 は、図 1 から図 3 をレンダリングした際の可視関数の評価回数の削減率を可視化した画像である。結果より、提案法が Veach の手法より高い削減率が得られていることが分かる。特に、Veach の手法では、分散の小さい平らな領域で削減率が 10%程度であるのに対し、提案法では、そのような領域でも可視関数の相関を利用することで 40%程度の削減率が得られている。図 3 のシーンで削減率が低くなっているが、このシーンはジオメトリが非常に複雑であり、可視関数の相関が小さいためと思われる。

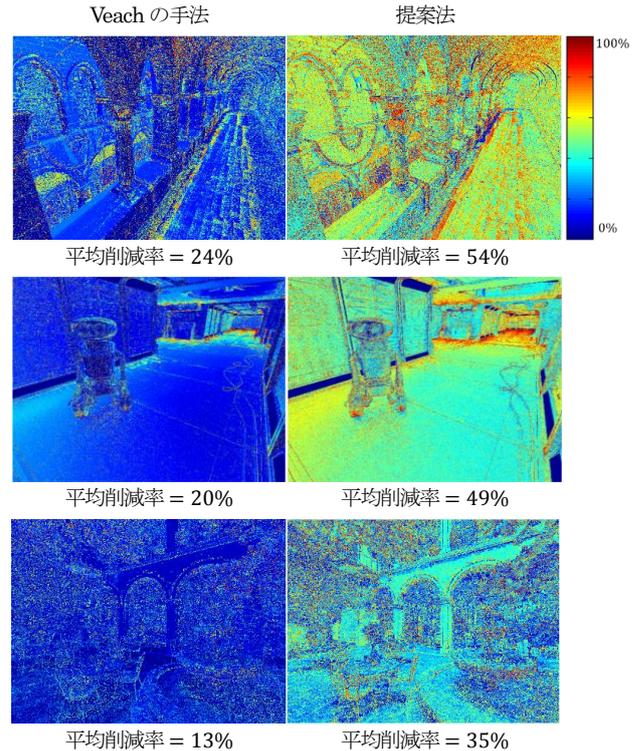
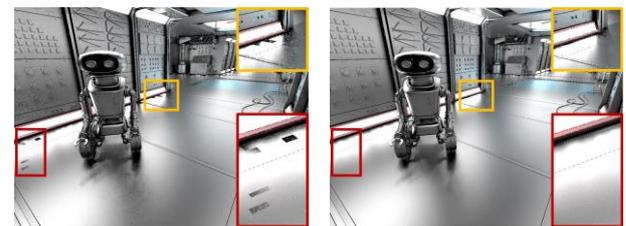


図 4 可視関数の評価回数の削減率の可視化



(a): サンプル数 10 (b): サンプル数 50

図 5 サンプル数の不足によるアーティファクト

## 6 制限事項

提案法では、確率分布の構築の際、VPL クラスタとシェーディング点クラスタ間の要素をサンプリングし、BRDF、幾何項、可視関数の積の平均値を推定している。そのため、指向性の強い材質などでは正確な推定値が得られず、生成される画像にブロック状のアーティファクトが発生する (図 5(a))。サンプル数を増加させることで、アーティファクトの発生を抑えることができるが (図 5(b))、確率分布構築の計算時間が増加してしまう。

## 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し、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ノイズの少ない画像を生成することができる。また、VPLとシェーディング点のクラスタを使用し、おおよそVPLの寄与に比例した確率分布を構築する手法を提案した。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、GPUに適したアルゴリズムの開発などが挙げられる。

## 参考文献

- [1] C. Dachsbacher, J. Krivanek, M. Hasan, A. Arbree, B. Walter, and J. Novak, “Scalable Realistic Rendering with Many-light Methods”, Eurographics STAR, 2013.
- [2] A. Keller, “Instant Radiosity”, Proc. of SIGGRAPH ’97, pp. 49-56, 1997.
- [3] R. Wang, and O. Akerlund, “Bidirectional Importance Sampling for Unstructured Direct Illumination”, Computer Graphics Forum, Vol. 28, No. 2, pp. 269-278, 2009.
- [4] I. Geogiev, J. Krivanek, S. Popov, and P. Slusallek, “Importance Caching for Complex Illumination”, Computer Graphics Forum, Vol. 31, No. 2, pp. 701-710, 2012.
- [5] Y. T. Wu, and Y. Y. Chuang, “VisibilityCluster: Average Directional Visibility for Many-Light Rendering”, IEEE Transaction on Visualization and Computer Graphics, Vol. 19, No. 9, pp. 1566-1578, 2013.
- [6] S. Popov, I. Georgiev, P. Slusallek, and C. Dachsbacher, “Adaptive Quantization Visibility Caching”, Computer Graphics Forum, Vol. 32, No. 2, pp. 399-408, 2013.
- [7] N. Billen, B. Engelen, A. Lagae, and P. Dutre, “Probabilistic Visibility Evaluation for Direct Illumination”, Computer Graphics Forum, Vol. 32, No. 4, pp. 39-47, 2013.
- [8] N. Billen, A. Lagae, and P. Dutre, “Probabilistic Visibility Evaluation using Geometry Proxies”, Computer Graphics Forum, Vol. 33, No. 4, pp. 143-152, 2014.
- [9] E. Veach, “Robust Monte Carlo Methods for Light Transport Simulation”, PhD thesis, Stanford University, 1998.
- [10] 名畑豪祐, 岩崎慶, 土橋宜典, 西田友是, “多光源レンダリング法のための効率的な可視関数の確率的評価”, VC/GCAD 合同シンポジウム 2014, pp. 13:1-13:6, 2014.
- [11] J. Ou, and F. Pellacini, “LightSlice: Matrix Slice Sampling for the Many-Lights Problem”, ACM Transactions on Graphics, Vol. 30, No. 6, pp. 179:1-179:8, 2011.
- [12] H. Dammertz, J. Hanika, and A. Keller, “Shallow Bounding Volume Hierarchies for Fast SIMD Ray Tracing of Incoherent Rays”, Computer Graphics Forum, Vol. 27, No. 4, pp. 1225-1233, 2008.