

# 自己遮蔽率マップによる アンビエントオクルージョンの効率化

## Efficient Ambient Occlusion using Self-Occlusion Maps

情報工学専攻 西村 龍哉  
Tatsuya NISHIMURA

### 概要

本研究は、アンビエントオクルージョンを計算する際に自己遮蔽率マップを導入することで効率的に計算結果を得る手法を提案した。自己遮蔽率は、遮蔽率を計算する点を中心とした単位半球面と周囲の面との交点からなる立体角から事前計算した。また、立体角は球面三角形の面積を計算することにより求めた。事前計算した自己遮蔽率は、遅延レンダリングを行うことからテクスチャとして保存した。遮蔽率の組み合わせは、二つの異なる遮蔽率を無相関である確率と仮定することで計算した。本研究で提案した手法により、いくつかのモデルをレンダリングし既存手法と比較した。

**キーワード:** アンビエントオクルージョン, 自己遮蔽率, 実時間レンダリング

### 1 はじめに

近年、様々な場面でコンピュータグラフィクス (CG) が用いられている。ゲームや映画のみならず、ライブなどのリアルイベントにも利用されている。ゲームの分野では、動的にシーンが変化するためリアルで高速なレンダリング手法が必要とされている。

アンビエントオクルージョンは、ゲームやその他のリアルタイムレンダリングを行うアプリケーションにおいて、グローバルイルミネーションの効果を近似するために広く用いられる手法である。しかし、任意のシーンに対するアンビエントオクルージョンの厳密解や、レイトレーシングなどを用いて数値解を求めようとする時間がかかってしまう。そのため、実際にアンビエントオクルージョンを求める場合、アンビエントオクルージョンを正確に求めるのではなく、ある程度現実のシーンと似ている結果が得られるアルゴリズムを用いて求めた近似解を利用する機会が多い。

本研究は、アンビエントオクルージョンを計算する際に自己遮蔽率を導入することで効率的に計算結果を得る手

法を提案する。提案手法のメリットは、自己遮蔽を事前計算により求めるため計算時間に影響をあたえないこと、遅延レンダリングを用いた手法に関して提案手法の導入が容易であることが挙げられる。

### 2 基礎知識

#### 2.1 環境光

光と物体との間の相互作用は複雑であり、レンダリングの際に求めるクオリティや環境によって大域照明 (global illumination) モデルと局所照明 (local illumination) モデルが使い分けられる。リアリティのある映像を作成する場合、間接光の影響まで正確に計算するため大域照明モデルが利用されるが、実時間レンダリングのように速度を求める場面では局所照明モデルが利用される。本研究では、局所照明モデルを利用した実時間レンダリングを扱う。また、局所照明モデルを扱うにあたり光を拡散反射光、鏡面反射光、環境光に分ける [1]。拡散反射光と鏡面反射光は光源から直接来る光である直接光を指し、環境光は光源からの光が他の物体によって反射や屈折を経て届くすべての間接光をまとめた光を指す。

環境光は、周囲すべての方向から入射する間接光を集めた光である。実際に求めるには、間接光を計算する必要があるが困難なため、入射する放射輝度  $L_i$  と反射率  $f$  を以下のように仮定する [1]。

- 空間を均一の放射輝度をもつ半球とする。すなわち、入射する放射輝度  $L_i$  はすべての方向に対して一定である。
- 環境光は視角に依存しない。すなわち、反射率  $f$  はすべての方向に対して一定である。

これらの仮定を利用することで、環境光は以下のように表せる。

$$L_a = f L_{ia} \int (\mathbf{n} \cdot \boldsymbol{\omega}) d\boldsymbol{\omega} \quad (1)$$

ここで、 $L_a$  は環境光、 $L_{ia}$  は周囲一定の入射する放射輝度、 $f$  は一定値とした反射率である。また、空間を半

球としていることから実際に (1) 式を計算すると、環境光  $L_a$  が一定値となることが分かる。

$$L_a = fL_{ia} \int (\mathbf{n} \cdot \boldsymbol{\omega}) d\boldsymbol{\omega} = \pi fL_{ia} \quad (2)$$

しかし、局所照明モデルを利用した実時間レンダリングのクオリティを上げようとした場合、現実の間接光が一定値をとらないように環境光が一定値であることは望ましくない。

## 2.2 アンビエントオクルージョン

アンビエントオクルージョンは周囲の遮蔽物による影響を考慮した環境光を求める手法である。遮蔽物による影響を考慮した環境光を求めるために、物体表面上の一点に届く環境光の強さを可視性の判定により調整することを考える。まず、(1) 式に注目する。入射する放射輝度  $L_{ia}$  と反射率  $f$  が一定値であることから、(1) 式の積部分が可視性に影響を及ぼすことが分かる。

$$O^*(\mathbf{x}, \mathbf{n}) = \frac{1}{\pi} \int (\mathbf{n} \cdot \boldsymbol{\omega}) d\boldsymbol{\omega} \quad (3)$$

しかし、(3) 式は、積分範囲を単位半球にしており頂点法線  $\mathbf{n}$  と方向  $\boldsymbol{\omega}$  によらず一定値である。これは、どの角度からでも任意の点  $\mathbf{x}$  に環境光が到達することを指し、任意の点  $\mathbf{x}$  が一定の光で照明されているかのように見えることを示している。そのため、ある点  $\mathbf{x}$  から方向  $\boldsymbol{\omega}$  に向かう光の可視性を判定する関数  $V(\mathbf{x}, \boldsymbol{\omega})$  を定義し (3) 式に導入する。関数  $V(\mathbf{x}, \boldsymbol{\omega})$  は、次のように定義する。

$$V(\mathbf{x}, \boldsymbol{\omega}) = \begin{cases} 0 & \text{ある点 } \mathbf{x} \text{ から方向 } \boldsymbol{\omega} \text{ に向かう光が遮られる} \\ 1 & \text{その他} \end{cases}$$

次に、(3) 式に関数  $V(\mathbf{x}, \boldsymbol{\omega})$  を導入する。

$$O^*(\mathbf{x}, \mathbf{n}) = \frac{1}{\pi} \int_{\Omega} V(\mathbf{x}, \boldsymbol{\omega}) (\mathbf{n} \cdot \boldsymbol{\omega}) d\boldsymbol{\omega} \quad (4)$$

ここで、積分区間  $\Omega$  は単位半球である。これにより、半球全体について可視性の判定を行うことができるようになり、(4) 式によりアンビエントオクルージョン  $O$  が求められる。アンビエントオクルージョンは、環境光に対する遮蔽率であるから (4) 式を用いて以下のように表せる。

$$O(\mathbf{x}, \mathbf{n}) = 1 - \frac{1}{\pi} \int_{\Omega} V(\mathbf{x}, \boldsymbol{\omega}) (\mathbf{n} \cdot \boldsymbol{\omega}) d\boldsymbol{\omega} \quad (5)$$

(5) 式は、環境光が完全に遮蔽されているとき 1、環境光が遮蔽されていないとき 0 を出力する。また、レイを飛ばすことを利用したモンテカルロ積分により、(5) 式を離散化することが可能である [4]。

$$O(\mathbf{x}, \mathbf{n}) \approx 1 - \frac{1}{N} \sum_{i=1}^N V(\mathbf{x}, \boldsymbol{\omega}_i) (\mathbf{n} \cdot \boldsymbol{\omega}_i) \quad (6)$$

ここで、 $N$  はサンプリングするレイの本数、 $\boldsymbol{\omega}_i$  はランダムに選ばれたレイの方向を表す。

## 3 自己遮蔽率

自己遮蔽率とは、自身の形状により環境光がどの程度遮蔽されているかを表す指標である。まず初めに、自己遮蔽率を求めるために自己遮蔽を定義する必要がある。自己遮蔽を定義するにあたり、アンビエントオクルージョンが物体表面上の一点に届く光の量に注目し、その点から半球状の空間において光源が見える割合を求めていることを利用する。ある点  $\mathbf{x}$  における自己遮蔽の定義は以下の通りである。

- ある点  $\mathbf{x}$  を中心とし、半径 1 の頂点法線  $\mathbf{n}$  の方向を正の軸とした単位半球において、ある点  $\mathbf{x}$  から光源が見えない単位半球の表面積。

自己遮蔽を  $A_{\text{self}}$  とすると、 $A_{\text{self}}$  は、単位半球の表面積のうち光源が遮蔽されている部分であり、これは光源が遮蔽されている部分の立体角 [6] を求めることと同義である。立体角とは、錐体の頂点を中心とした半径 1 の単位球の球面を切り取った面積で表される量である。単位半球の立体角は  $2\pi$  であるから、ある点  $\mathbf{x}$  での自己遮蔽率  $O_{\text{self}}(\mathbf{x})$  は以下のように求めることができる。

$$O_{\text{self}}(\mathbf{x}) = \frac{A_{\text{self}}}{2\pi} \quad (7)$$

ここで、 $A_{\text{self}}$  は光源が遮蔽されている単位半球の表面積である。

## 4 Efficient Ambient Occlusion

Efficient Ambient Occlusion は、SSAO[5, 2] がサンプル点の可視性を判定するときに自己遮蔽率マップを利用してアンビエントオクルージョンを求める。具体的には、選ばれたサンプル点の位置が現在アンビエントオクルージョンを求めている点と同じオブジェクト上の点なのか判定する。判定には、オブジェクトの番号が保存されたテクスチャを利用して行う。判定した結果、同じオブジェクト上の点の場合は自己遮蔽率としてすでに計算しているため遮蔽されている点として数えない。このようにして、自己遮蔽率としてすでに計算された位置のサンプル点以外を数えていき遮蔽率を計算する。求めた二種類の遮蔽率をそれぞれ  $O_{\text{self}}$  と  $O_{\text{run}}$  とすると、最終的なアンビエントオクルージョン  $O$  は遮蔽率の組み合わせより二つの異なる遮蔽率を無相関であると仮定した遮蔽物にレイが当たらない確率と捉え乗算により求める [3]。

$$O = 1 - (1 - O_{\text{self}})(1 - O_{\text{run}}) \quad (8)$$

## 5 実験結果

レンダリングした結果を比較するにあたり、面数の異なる二種類のモデルを利用した。それぞれのモデルの頂点数及び面数は、ウサギのモデルが頂点数 14290、面数 28576 のモデルであり、ウマサメのモデルが頂点数が 7272、面数が 14540 である。また、二種類のモデルを用いて、単体に対してレンダリング、簡略化したモデルに対してレンダリング、二つのモデルを配置してレンダリング、および、モデルの向きを変更してレンダリングして実験した。

ウサギのモデルについて比較する。提案手法を利用した図 1c は、正解データである図 1a と比較して細かい凹凸が目立ってしまったが、SSAO を利用した図 1b も含めて比較すると、顔の輪郭や耳の形状による陰影を再現することができている。ウサギのモデルを簡略化したモデルのレンダリング結果をそれぞれ比較すると、元のモデル、面数を  $\frac{1}{4}$  にしたモデル、面数を  $\frac{1}{10}$  にしたモデルの順に首の下の陰影が薄くなっていることが確認できる。また、面数を  $\frac{1}{10}$  にしたモデルが一番滑らかな陰影となっている。二種類のモデルを配置したレンダリングについて比較する。提案手法を利用した図 2c は、SSAO を利用した図 2b と比較してレンダリング結果のざらつきが軽減したことや SSAO では再現できていない顔の形状による陰影を再現することができている。正解データである図 2a と比較してウマサメのモデルの足や尾など陰影が薄い箇所が複数確認できる。二方向のレンダリングについて比較する。SSAO を利用した図は、足元の付け根の陰影が正面から見た場合と横から見た場合で濃淡に違和感がある。しかし、提案手法を利用した図や正解データである図は、足元の付け根の陰影が正面から見た場合と横から見た場合で濃淡に違和感がない。また、顔の輪郭による陰影が正面から見た場合と横から見た場合で一致している。

実験結果から、提案手法は SSAO に比べ細かい陰影を再現することには成功したが、入力メッシュによっては正解データと比較して提案手法は細かい陰影が多く現れてしまっている箇所が確認できた。また、提案手法は SSAO や正解データと比較して陰影が薄い箇所もいくつか見受けられた。そのほかに、SSAO で見られるざらつきについても提案手法により軽減することができた。ざらつきが軽減できることによりサンプル点をより少なくした場合のレンダリングにおいてもある程度のクオリティが期待できる。

陰影について濃い箇所や薄い箇所が現れてしまった原

因として、自己遮蔽率の計算における遮蔽として考慮する範囲を自己遮蔽率を計算する点からの一近傍にしたことが考えられる。SSAO での細かい陰影の表現が上手くいかない原因は、入力したオブジェクトの形状をデプスバッファを用いて近似していることやサンプル点の数を多くとる必要があることが考えられたため、自己遮蔽率を計算する点からの一近傍にした狭い範囲での自己遮蔽率の計算を行った。結果として、細かい陰影をの再現は達成したが入力メッシュ上の微小な凹凸が現れることや陰影が薄い箇所が出現してしまった。

## 6 むすび

本研究では、アンビエントオクルージョンを計算する際に SSAO に対して自己遮蔽マップを導入することで効率的に計算結果を得る手法を提案した。自己遮蔽率マップは、自己遮蔽率のある点から光源が見えない単位半球の表面積の割合として定義し、単位半球の表面積が立体角であることを利用して計算しテクスチャとして出力することで求めた。また、Efficient Ambient Occlusion を求める際に必要なオブジェクトの判定にはオブジェクト番号を保存したテクスチャを用い、異なる遮蔽率どうしの組み合わせには Kontkanen らが提案した手法を用いた [3]。今後の課題として、いくつか挙げられる。

自己遮蔽率を計算する最適な範囲をどのようにして求めるかである。提案手法では、重みを考慮したアンビエントオクルージョンを計算していないため、自己遮蔽率を計算する際に距離による重みをつけることでより広い範囲の遮蔽を考慮した場合の自己遮蔽率を計算が可能になることが考えられる。

動的に動くモデルに対してどのように自己遮蔽率を計算するのかである。自己遮蔽率をシェーダ内で実時間による計算ができる式を求めることで動的に動くモデルに対しても提案手法を取り入れることができると考えられる。

自己遮蔽率と他の遮蔽率の組み合わせに関してより優れた方法を確立することである。提案手法では、自己遮蔽率と SSAO の乗算によりアンビエントオクルージョンを計算したが、どちらかの遮蔽率が極端に小さいと望ましい結果が得られない。この問題に対しては、場合分けすることにより改善されることが考えられる。

## 参考文献

- [1] R. L. Cook and K. E. Torrance. A reflectance model for computer graphics. *ACM Transactions*

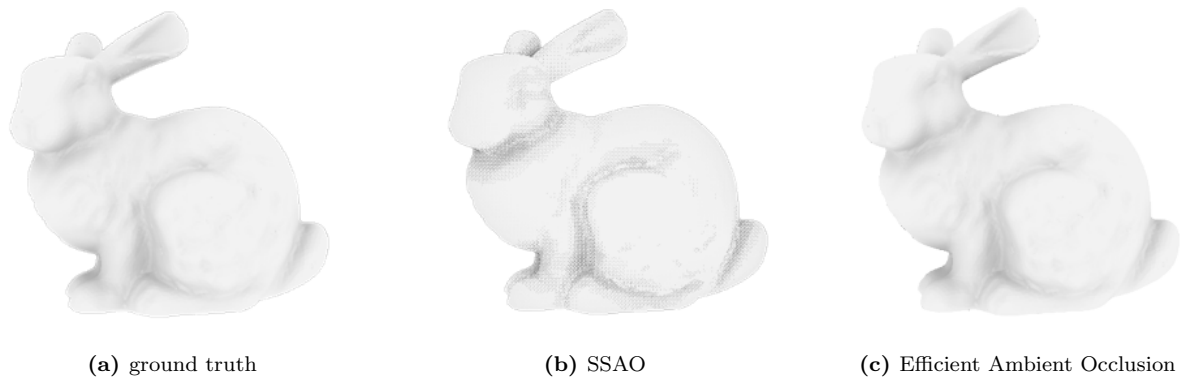


図1 単体モデルのアンビエントオクルージョンの値をグレースケール画像として出力した結果

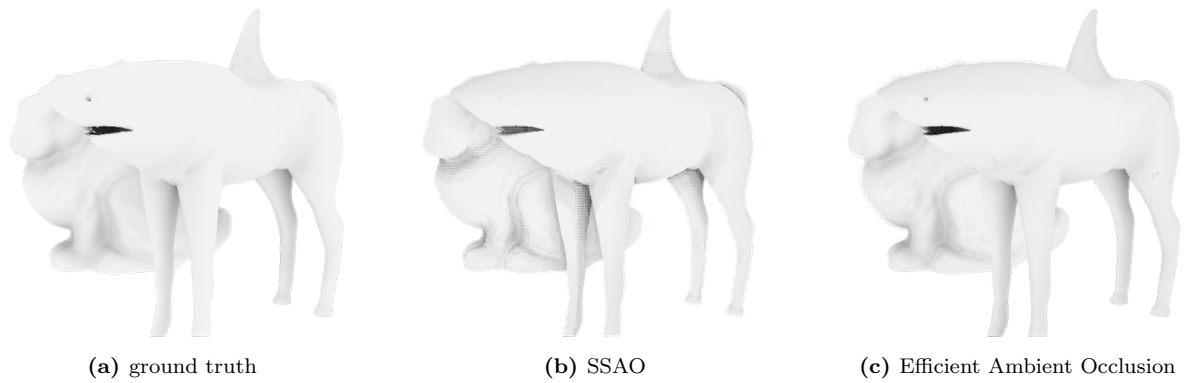
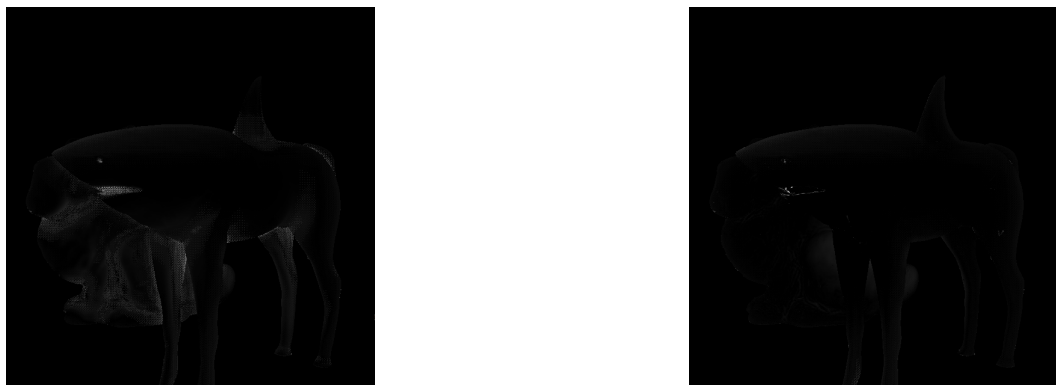


図2 複数モデルのアンビエントオクルージョンの値をグレースケール画像として出力した結果



(a) 図2a と 図2b の差分画像

(b) 図2a と 図2c の差分画像

図3 ground truth と各手法の差分画像

on *Graphics (ToG)*, 1(1):7–24, 1982.

[2] A. Kaplanyan. Light propagation volumes in cryengine 3. *ACM SIGGRAPH Courses*, 2009.

[3] J. Kontkanen and S. Laine. Ambient occlusion fields. In *Proceedings of the 2005 symposium on Interactive 3D graphics and games*, pp. 41–48, 2005.

[4] H. Landis. Production-ready global illumination. *ACM SIGGRAPH Courses*, 2002.

[5] M. Mittring. Finding next gen: Cryengine 2. *ACM SIGGRAPH Courses*, 2007.

[6] A. Van Oosterom and J. Strackee. The solid angle of a plane triangle. *IEEE Transactions on Biomedical Engineering*, BME-30(2):125–126, 1983.