

レイトレーシングにおける半影表現の新しいアプローチ

吉村ももこ^{†1} 笠原信一^{†1} 安藤大地^{†1}

3次元CGにおいて半影は特に室内のシーンのリアリティを出すために重要な要素であるが、半影生成のためには本来ならば面光源を扱わなければならない、高速に半影を表現することは容易でない。3次元CGの代表的なレンダリング手法であるレイトレーシングは、リアルな画像を生成出来るが計算コストが高く、特に高速化が必要とされている。そこで、レイトレーシングの半影計算を高速化する新しい方法として、面光源よりも計算コストの低い点光源を用いて半影を表現する新しい手法を提案する。点光源によって半影を簡易的に表現する手法は、Variance Shadow Mapsなどといったものが挙げられる。本報告は、これら既往の研究の特徴と問題点を検証した上で、その解決策として本提案が有効であることを提示する。

New Approach to generate penumbra for Ray Tracing

MOMOKO YOSHIMURA^{†1} SHINICHI KASAHARA^{†1}
DAICHI ANDO^{†1}

Penumbra is important factor for three dimensional Computer Graphics in order to represent reality especially in interior view. However, it is not easy to generate penumbra by high speed calculation because we must handle area lights to generate penumbra. Ray tracing, that is the typical rendering algorithm for three dimensional Computer Graphics, can generate realistic images, but calculation cost is expensive, and ray tracing needs high speed algorithm. Therefore, we propose a new algorithm to generate penumbra using point lights instead of area lights in order to generate penumbra by high speed calculation using ray tracing. There are some algorithms to generate penumbra in a simple manner using point lights, for example Variance Shadow Maps. This paper verifies the characteristics and limitation of these algorithms and shows the new approach we propose in this paper is effective as the solution of the limits.

1. はじめに

3次元CGにおいて、影の表現はシーン（特に室内のシーン）のリアリティを出すために重要な要素であり、これまでも様々な研究がなされている。その影表現の中でも最も重要な役割を果たすのが、半影である。

影は、光源からの光が計算点に到達するまでに、途中にある物体によって遮られることによって生じる。光源は点光源と面光源に大別される。点光源の影は、影である部分とそうでない部分が明確に区別される。一方、面光源は、面積があるために、途中の障害物によって面の一部だけが遮られる場合がある。そのために、光源からの光が全て到達する所と、完全に遮られる所（本影）との間に、面光源の光の一部だけしか到達しない部分（半影）が生じる。図1より、点光源による画像と、面光源による画像を比較してみると、半影が加味された面光源計算の画像の方がよりリアルなのは明らかである。しかしながら、面光源を計算するには膨大な計算時間が必要となる。それは、点光源が計算点について影かそうでないかという0か1かの判定をするのに対し、面光源は計算点の値を求めるには光源の見える部分領域を計算し、0と1の中間の値を計算する必要があるからである。したがって面光源の影計算は高速化が必要とされているのである。

3次元CGの代表的なレンダリング手法であるレイトレーシング[1]は、質の高い表現が出来るものの、本来それ自身には面光源を計算する機能はない。レイトレーシングにおいて面光源を表現する方法として、レイトレーシングの拡張版であるパストレーシング[2]がある。しかしこの方法は膨大な計算時間がかかるという欠点がある。

そこで本研究では、計算高速化のために、面光源をそのまま扱うのではなく、面光源を点光源で近似して計算するアルゴリズムを提案する。本アルゴリズムはレイトレーシングをベースにした親和性の高いアルゴリズムで、レイトレーシングから容易に拡張することができるものである。

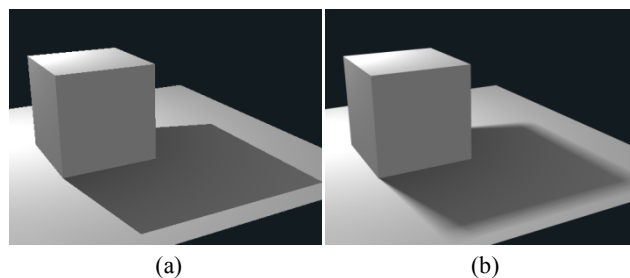


図1 (a)点光源による影 (b)面光源による影

2. 影計算の既往アルゴリズム

2.1 点光源を扱うアルゴリズム

本研究で提案する手法は、先にも述べたように、レイトレーシングに組み込むことで強みを発揮できるものであるの

^{†1} 首都大学東京 システムデザイン学部
Department of System Design
Tokyo Metropolitan University

で、レイトレーシング法における点光源の影の計算を説明する。

現実世界では、光源から発せられた無数の光線の中で、視点に届くものだけが見えているものとして認識されるが、3次元CGでそれをシミュレートしようとする、膨大な計算時間になってしまう。そこで、光源から出ている光全てを考慮するのではなく、視点に届く光だけを最初から考慮する。そのためには、本来なら光源→視点という方向に向けられている光線を、視点→光源という逆順路で辿ってしまえば良い。この方法をレイトレーシングと呼ぶ。レイトレーシングでは屈折・透過を考慮した高品質な画像を生成できるが、一つのピクセルに対して一つのレイを辿るため計算に時間がかかる。

レイトレーシングで影付けをする場合、視点からスクリーン上の計算点に向かってレイを飛ばして、光源からの遮蔽物の有無を調べる。図2において点Aが影かそうでないかを求めたいとき、まず視点から点Aまでレイを伸ばす。さらに点Aと光源を結ぶ。その光源と点Aを結んで出来た線分の中には遮蔽物との交点が存在するので、点Aは影と判定される。

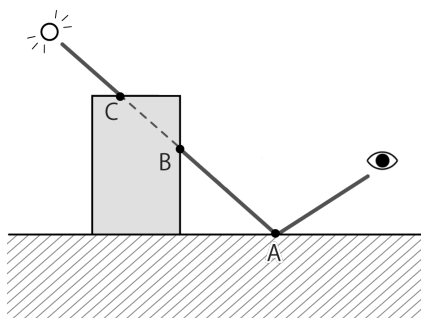


図2 レイトレーシング法の影計算

2.2 面光源を扱うアルゴリズム

面光源の強さを計算するアルゴリズムは、障害物によって面の一部が隠れる場合の形態係数をどのように求めるかということに帰着する。この問題を解決するために、図形処理的に光をボリュームとして扱うアプローチと、光を線の集合だと考慮してモンテカルロ法によって確率的に解くアプローチの二種類が挙げられる。前者はラジオシティ法[3][4]と呼ばれ普及している手法であり、後者はパストレーシング法と呼ばれている。

2.2.1 ラジオシティ法

ラジオシティ法は一つの面の中の明るさは一様な拡散光源として間接光の成分の計算を行うため、一つの面の面積が大きく、明るさが場所によって異なってしまう場合は面を細かいパッチに分割する。パッチ内では明るさは一定のものとして計算する。このとき面を分割するための図形処理をする必要がある。

さらに、見えている部分のみの値を取りだすため、隠面

消去を行う。

図形処理による解決方法は考え方が直感的で分かりやすいものだが、パッチを分割するときに複雑な形状だと扱いが難しく、物体の形状が制約されてしまう。また分割することで計算量が膨大になり、複雑な形状のモデルの影計算をするのが困難である。

2.2.2 パストレーシング法

一方でパストレーシング法では、光をボリュームではなく線の集合として扱う。面内の明るさを一定と仮定して制限付きの計算するラジオシティ法とは違い、光線一本一本を独立に扱うので、面を細かく分割するなど煩わしい図形処理を必要としない。

光線を追跡するという点では従来のレンダリング手法であるレイトレーシング法の拡張版であると言える。レイトレーシング法と同様に材質特性（鏡面反射、屈折、反射指向性）に対して汎用性を持つが、光線を一本ごとに扱うので、面をまとめて扱う場合と比較すると計算量が膨大になる。光を光線で考慮するとその数は無限になってしまい、全てを計算することは物理的に不可能であるので、確率を用いて扱う。つまり、計算する光線を確率によってランダムに選び徐々に数を増やしていき、必要な精度が得られたと判断した時点で計算打ち切る手法を取る。サンプリングされた情報から全体を推測するため、サンプル数によっては推測に誤差が生じ、画像内にまだ模様として現れてしまう。これを解消するためにはサンプル数を増やせば良いが、そのためには膨大な数の光線をサンプルとして扱わなくてはならず、このことが大きな計算コストとなっている。

2.3 点光源で面光源を擬似的に扱うアルゴリズム

半影を高速に表現する方法として、面光源を扱う代わりに、シャドウマップ法をベースにして点光源の影を作成し、その影をぼかすことによって半影を表現する手法が研究されている。

シャドウマップ[5]は光源にスクリーン（視点）を立て、光源から物体までの深度値を格納したシャドウマップ（シャドウデプスマップ）を作成する。次に、視点からのシーンをレンダリングし、可視点から光源までの距離を求める。その値をシャドウマップ深度値と比較し、深度値よりも座標値の方が大きければ計算点は影と判定される。この方法では遮蔽物の複雑さに関係なく影が描画できるのが長所であるが、サンプリング数によって画像の質と処理速度に差が出てしまうのと、エイリアシングが生じてしまう問題がある。

シャドウマップ自体には半影を表現するアルゴリズムは組み込まれていないため、シャドウマップで半影を表現するにはそうした機能を追加しなくてはならない。シャドウマップを拡張して半影を表現する代表的な方法としてPCF法やVSM法が発表されている。

2.3.1 PCF(Percentage Closer Filtering)法

PCF法[6]は、シャドウマップ法によって、各計算点で影か日向かを判定した後、計算点の周囲について影か日向かをサンプリングし、その影と日向のサンプル数の割合によって、影の濃さを決定する。つまりこの方法は、単純に、影と日向の混ぜ合わせによって影のエッジをぼかすもので、面光源からの距離に関係なくすべての影が一律にぼけた影になり、面光源をシミュレーションしているとは言えない。

2.3.2 VSM(Variance Shadow Maps)法

VSM法[7]は、シャドウマップが保持している深度値を利用することで、面光源からの距離によって影のぼけ具合を変化させる手法である。

基本的なシャドウマップ法は、シャドウマップに記録されている深度（光源から遮蔽物までの距離） x と、計算点での光源までの距離 t を比較して、 $x < t$ の場合に影と判定する。VSM法ではこの判定の x の代わりに、周囲のシャドウマップの複数の深度の平均値 μ を求め、 $\mu < t$ の場合に影と判定する。影と判定されたところでも、周囲の状況によって部分的に日向（半影）になることがある想定し、式(1)によって、日向になる確率（ $x > t$ の確率、 $P(x \geq t)$ ）の上限($P_{\max}(t)$)を求め、この値を影の濃さとする。

$$P(x \geq t) \leq P_{\max}(t) = \frac{\sigma^2}{\sigma^2 + (t - \mu)^2} \quad (1)$$

ここで、 σ^2 は周囲のシャドウマップの複数の深度の分散である。

この方法は、PCF法の単純な影のぼかしではなく、計算点周囲のシャドウマップの深度を考慮して影の濃さを変えることができ、遮蔽物に近いところの影は濃くなり、遮蔽物から遠ざかるにしたがって影が薄くなり、面光源の影に近い影が表現できる。

しかしながら、日向になる確率の上限値が物理的な根拠がない。また、シャドウマップ法をベースにしているため、影のエッジの粗さ（エイリアシング）は残ったままで、高画質と言えるものではない。

3. 点光源で面光源を表現する新アルゴリズム

本研究では、面光源を高速な計算時間で表現する方法として、面光源を点光源で近似して表現するアルゴリズムを提案する。上記の既往の面光源を点光源で近似する方法は、シャドウマップ法をベースにしているため、影の品質が劣る。一方、本提案アルゴリズムは、レイトレーシング法をベースにして、面光源を点光源で近似することにより、高品質な半影の表現力を保っている。

現実世界の影は、遮蔽物からの距離が遠くなればなる程薄くぼんやりとしていく(図3)。本提案は、図4の線分

AB間の距離 d の値に応じて、点Aの周囲を平滑化し半影を表現する方法である。つまり、距離 d の値が大きければ大きい程、点Aの周囲の平滑化するピクセル数は大きくなる。

レイトレーシング法では、通常の計算過程で距離 d を求めるので、その距離 d を改めて計算する必要がない。そのことが、本提案手法をレイトレーシング法の拡張版とする理由である。

ここで、 d と平滑化するピクセル数をどのように関連付けるかが問題となってくる。そこで図5のような面光源の状況を考える。図5は面光源、遮蔽物、地面の全てが平面の場合を想定しており、それらは互いに平行である。 W_{light} は面光源の（大きさ）幅、 W_{penumbra} は平滑化するピクセル数（半影の幅）である。こうしたシンプルな状況では二つの三角形の相似比により、次式の関係が成り立つ。

$$W_{\text{penumbra}} = \frac{Z_{\text{receiver}} - Z_{\text{occluder}}}{Z_{\text{occluder}}} W_{\text{light}} \quad (2)$$

このとき Z_{occluder} は光源から遮蔽物までの距離、 Z_{receiver} は光源からレシーバーまでの距離である[8]。ここで $Z_{\text{receiver}} - Z_{\text{occluder}} = d$ であり、また W_{light} と Z_{occluder} は一定であるとすると、距離 d と平滑化ピクセル数は比例する。

通常のレイトレーシング法は、影だけでなくシェーディング、テクスチャ、反射などの混在したレンダリングの最終結果をバッファに格納しているが、本提案では影の部分だけをぼかす必要があるため、ピクセル毎に影の値だけを格納するバッファと、それ以外を格納するバッファの二つに分け、最後に合成するという方法を取る。それとは別に、距離 d を格納するバッファも作成する必要がある。

このアルゴリズムの特徴は、影のぼかしにおいて、各計算点における正確な光源から遮蔽物までの距離 d を使用するために、シャドウマップ法における影のエッジのエイリアシングが発生せず、高品質な影を生成できること、およびレイトレーシング法の情報を活用するので、レイトレーシング法と親和性が高く、レイトレーシングのプログラムに組み込むことが容易な点である。

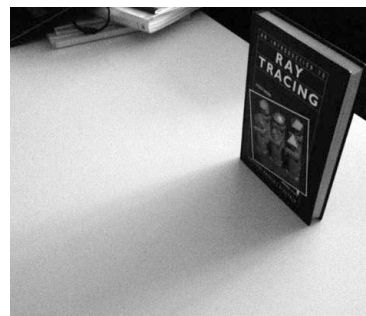


図3 現実世界の影

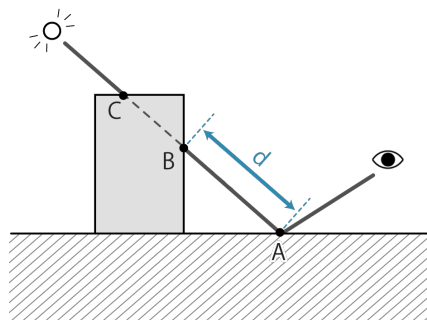


図 4 レイトレーシング法における距離 d の抽出

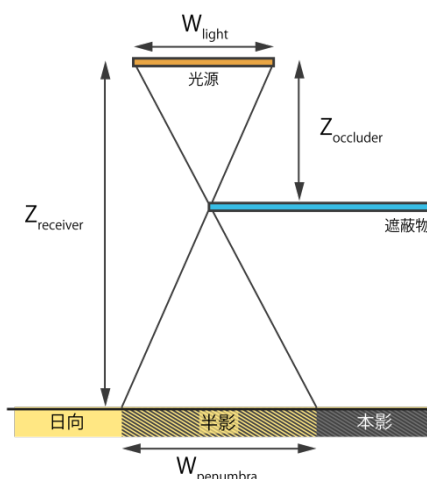


図 5 半影の幅のシミュレーション

4. 今後の課題

本論文では、面光源の影を効率的に表現する新しいアルゴリズムのアイデアを提案した。これからこのアルゴリズムを実装して、画像品質と計算時間の両面からこのアルゴリズムの評価を行う。実装の過程で、新たな問題が生じることが予想されるので、それらの問題についての解決策を考案することで、本アルゴリズムの詳細を完成させていく。

参考文献

- 1) Whitted, Turner. *An Improved Illumination Model for Shaded Display*. : Communications of the ACM, Vol.23, No.6, pp.343-349, 1980.
- 2) Kajiya, James T. *The Rendering Equation*. : Computer Graphics(SIGGRAPH '86 Proceedings), pp.143-150, 1986.
- 3) Goral, Cindy M. et al. *Modeling the Interaction of Light between Diffuse Surfaces*. : Computer Graphics(SIGGRAPH '84 Proceedings), pp.213-222, 1984.
- 4) Cohen, Michael F. and Greenberg, Donald P. *The Hemi-Cube A Radiosity Solution for Complex Environments*. : Computer Graphics(SIGGRAPH '85 Proceedings), pp.31-40, 1985.
- 5) Williams, Lance. *Casting Curved Shadows on Curved Surfaces*. : Computer Graphics(SIGGRAPH 78' Proceedings), pp.270-274, 1978.

- 6) Reeves, William T., Salesin, David H. and Cook, Robert L. *Rendering Antialiased Shadows with Depth Maps*. : Computer Graphics(SIGGRAPH '87 Proceedings), pp.283-291, 1987.
- 7) Donnelly, William and Lauritzen, Andrew. *Variance Shadow Maps*. : Proceedings of the 2006 Symposium On Interactive 3D Graphics and Games, pp.161-165, 2006.
- 8) Eisemann, Elmar and Schwarz, Michael. *Realtime Shadows*. : CRC Press, 338p, 2012.
- 9) 笠原信一. CGレンダリングにおける間接光計算の高速化アルゴリズム. デザイン学研究 Vol.55. NO.1.2008.
- 10) Glassner, Andrew S. *An Introduction to Ray Tracing*. : Academic Press, 1989.
- 11) Watt, Alan. *3D Computer Graphics*. 3rd ed. : Addison-Wesley Publishing, 570p, 2000.
- 12) Akenine-Möller, Tomas, Haines, Eric and Hoffman, Naty. *Real-Time Rendering 3rd ed*. : CRC Press, 1027p, 2008.