

物理ベースのリアルタイムレンダリングに関する研究

Survey of Real-time Physically Based Rendering

1W173092-0 富岡 祐介 指導教員 坂井 滋和 教授
TOMIOKA Yusuke Prof. SAKAI Sigekazu

概要： 物理ベースのレンダリング (PBR) はビデオゲームなどのリアルタイムなパフォーマンスが要求される場でも必要とされ始めている。しかし、それに対処するための情報や最新の研究はまとまっておらず、日本語の情報源はほとんどない。そこで本研究は、リアルタイムに PBR を行う際の問題を明確にし、それに対処する最新の研究を俯瞰するための情報を論文に記述することで、この問題に対する研究や開発に取り組むための調査内容を提供する。

キーワード：物理ベースレンダリング, レイトレーシング, ハイブリッドレンダリング, グローバル照明

Keywords : physically based rendering, ray tracing, hybrid rendering, global illumination

1 はじめに

物理ベースレンダリング (PBR) [1] と呼ばれる手法は、グローバル照明を含むレンダリングを、数学的な裏付けのあるシンプルなアプローチで実現できる。ただし、計算コストが非常に高いため、リアルタイムに実行することは長い間難しいとされていた。

しかし、様々な研究がこの問題を扱い、リアルタイムな PBR を実現している例が生まれるようになって [2]。本研究では、この成果を利用したさらなる研究や開発を助長することを目的として、リアルタイムな PBR を可能にした研究とそのアプローチを調査・分類する。

2 PBR

PBR はオブジェクトの表面上の点 \mathbf{p} での光の相互作用を散乱方程式

$$L_o(\mathbf{p}, \omega_o) = \int_{S^2} f(\mathbf{p}, \omega_o, \omega_i) L_i(\mathbf{p}, \omega_i) |\cos\theta_i| d\sigma(\omega_i)$$
で扱う。ここで、 $L_o(\mathbf{p}, \omega_o)$ 及び $L_i(\mathbf{p}, \omega_i)$ はそれぞれ方向 ω_o と ω_i で出射及び入射する光の量である放射輝度を表す。また、 θ_i は点 \mathbf{p} での法線と方向 ω_i の間の角度、 $d\sigma(\omega_i)$ は方向 ω_i の微分立体角を表す。

オブジェクトの表面に入射する光は、他のオブジェクトで放出されたものか、散乱したものである。このことから、シーン内を飛び回る光の放射輝度の大きさは光輸送方程式

$$L(\mathbf{p}, \omega_o) = L_e(\mathbf{p}, \omega_o) + \int_{S^2} f(\mathbf{p}, \omega_o, \omega_i) L(t(\mathbf{p}, \omega_i), -\omega_i) |\cos\theta_i| d\sigma(\omega_i)$$

で求められる。ここで、 $t(\mathbf{p}, \omega_i)$ は、方向 ω_i にレイと呼ばれる半直線を伸ばしていき、シーン内のオブジェクトにぶつかった場合、その点の位置を返す関

数である。レイを用いて入射した光の道筋を辿っていくこの操作は、レイトレーシングと呼ばれる。

この光輸送方程式の積分は、解析的に解くことが難しく、モンテカルロ法と呼ばれる数値積分を用いて

$$L(\mathbf{p}, \omega_o) = L_e(\mathbf{p}, \omega_o) + \frac{1}{N} \sum_{j=1}^N \frac{f(\mathbf{p}, \omega_o, \omega_r) L(t(\mathbf{p}, \omega_r), -\omega_r) |\mathbf{n} \cdot \omega_r|}{p(\omega_r)}$$

のように解くことができる。ここで、 ω_r は確率密度関数 p に基づいて生成された方向サンプルである。

このように、光輸送方程式、レイトレーシング、モンテカルロ法を用いたレンダリングが PBR を形作っている。

3 リアルタイムレンダリング

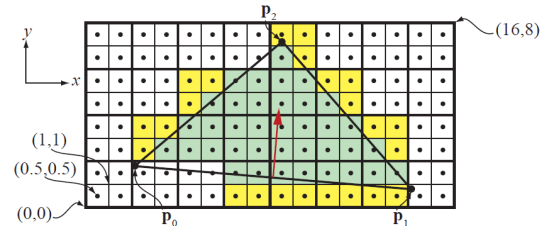


図1 この図はラスタライズの様子を示す [3].

リアルタイムレンダリングは、インタラクティブに変化するシーンをレンダリングするために、秒間 30 ~ 120 フレームの画像の内容を計算する必要がある。

この制約の中で、従来の手法は画像のどの部分にオブジェクトが映るのかを決定するラスタライズ (図 1) を中心に設計されている。ラスタライズされたオブジェクトのみが、画像のピクセルの色に影響すると考えるため、他オブジェクトの情報をピクセルの色の計算に利用することが難しく、グローバル照明が難しい要因となっている。

4 リアルタイム PBR の問題

リアルタイムな制約下で PBR を行うためには、主に以下のような問題が発生する。

サンプル不足によるノイズの発生. 計算コストの高さから、カメラに入射する放射輝度をモンテカルロ法で求めるためのサンプル数 N があまり多く出来ない。これによって、レンダリングされた画像にランダムな色のノイズが現れる。

ハードウェアに最適化された実装. リアルタイムレンダリングに用いられる GPU はラスタライズをベースとしたレンダリングに最適化されており、PBR を効率よく行うためには、プログラムの慎重なマッピングが必要。

移動/アニメーションをするシーンへの対処. リアルタイムにレンダリングを行わなければならないケースでは、シーン内のオブジェクトが頻繁に移動・変形する動的なシーンのレンダリングが必要になる事が多い。これによって、オフラインの PBR とは異なる最適化が必要になる。

5 リアルタイム PBR の戦略

リアルタイムに PBR を行うための最新の研究は、主に以下のような戦略を用いていることが調査によって明らかになった。

事前計算. シーンの中のカメラや光源の位置、オブジェクトの形状などの要素の固定することで光輸送を事前に計算することができる。

スクリーン空間の利用. ラスタライズをベースとしたレンダリングでは、最終的な画像の中でのオブジェクトの位置のみがピクセルの色の計算に用いられる。この性質を利用して、最終画像に映るオブジェクトのみでレイトレーシングを行うことで、計算コストを抑えつつ、従来のレンダリングに PBR の効果を加えることが可能となる。

仮想光源の利用. エリアライトや間接照明を考慮したレンダリングでは、モンテカルロ法を素早く収束させることが難しいため、それらをいくつかの点光源で置き換えることで、収束の早さを簡単に調整できるようになる。

時間フィルタリング. フレームごとに独立してモンテカルロ法を解くのではなく、それまでのフレームの結果を時間フィルタリングを行い再利用することによって、フレームごとのサンプル数を抑えつつノイズのない結果が得られる。

ノイズパターンの変更. モンテカルロ法によって現れるホワイトノイズに近いランダムなノイズを、ブルーノイズと呼ばれる低周波を取り除いたノイズに置き換えることで、人間の知覚にとって不快でない画像が生成できる。

コヒーレンスの活用. ソフトウェアにとっても、ハードウェアにとっても、処理すべき対象の一貫性は処理速度を引き上げる重要な手段である。PBR の処理に対して適切に遅延、並び替えを行うことで、効率を引き上げることができる。

高速化構造の改善. レイを素早くトレースするために用いられる、シーンの情報を格納する高速化構造を改善することで、PBR 全体のパフォーマンスを引き上げることができる。

専用のハードウェアの利用. GPU がラスタライズに特化しているように、レイトレーシングに特化したハードウェアを用いることで、速度の向上や、プログラミングモデルの改善が実現できる。

機械学習の利用. 機械学習は画像の後処理方法としても高い性能を発揮するが、光輸送方程式の解く操作や、事前計算した内容をコンパクトに格納するための手段としても利用が検討されている。

ハイブリッドレンダリング. 近年のハードウェアや API の方向性として、ラスタライズとレイトレーシングを組み合わせたハイブリッドレンダリングが用いられる。それぞれの処理を専用のハードウェアユニットで行うことで、かつてないパフォーマンスを実現することが可能になっている。

6 まとめ

本研究では、リアルタイムに PBR を行う際の問題と、リアルタイムに PBR を行うために研究されてきた様々な手法を分類し、調査の結果をまとめた。

参考文献

- [1] Matt Pharr, Wenzel Jakob, and Greg Humphreys. *Physically based rendering: From theory to implementation*. Morgan Kaufmann, 2016.
- [2] Jan Schmid and Johannes Deligiannis. It just works: Ray-traced reflections in 'battlefield v', 2019. GDC 2019.
- [3] Tomas Akenine-Möller, Eric Haines, and Naty Hoffman. *Real-time rendering*. Crc Press, 2019.