

Title	フォトンマッピング法を用いた3DAPIによる大域照明手法
Author(s)	柿原, 利政
Citation	
Issue Date	2008-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/4288
Rights	
Description	Supervisor : 宮田一乗, 知識科学研究科, 修士

修 士 論 文

フォトンマッピング法を用いた
3DAPIによる大域照明手法

北陸先端科学技術大学院大学
知識科学研究科知識システム基礎学専攻

柿原 利政

2008年3月

修 士 論 文

フotonマッピング法を用いた
3 DAPIによる大域照明手法

指導教官 宮田 一乗 教授

北陸先端科学技術大学院大学
知識科学研究科知識システム基礎学専攻

650012 柿原 利政

審査委員： 宮田 一乗 教授（主査）
杉山 公造 教授
西本 一志 教授
由井蘭 隆也 教授

2008年2月

概要

3次元コンピュータグラフィックスにおける研究,開発の大きな流れの中に写実的なレンダリング手法の開発がある.この流れの成果として,近年,映画やTVといったメディアでは実写映像と合成しても遜色ない品質の写実的な映像をコンピュータで描くことが可能になってきた.これらの写実的な画像のレンダリングには,レンダリングされるモデル中での光のふるまいが重要となり,光のふるまいを計算したレンダリングに関する研究領域を大域照明と呼んでいる.

大域照明では,間接照明のようなモデルの表面から表面への光の伝搬をレンダリングすることが可能であるが,計算量が多く,ゲームのようなインタラクティブな表示が必要な分野では使うことができない.また,一般的にこうしたインタラクティブな分野では **DirectX** や **OpenGL** といった **3D API** を用いたラスタライズによってレンダリングを行う.一方,多くの大域照明手法はレイトレーシングを基としている.このことが,ラスタライズによるレンダリングに大域照明を組み込むことへの大きな障壁となり,大域照明をインタラクティブな分野に適用できない一因となっている.

本論文では,大域照明の手法の1つであるフォトンマッピング法を用いることで **3D API** による大域照明を実現する.従来のフォトンマッピング法は,レンダリングにレイトレーシングが用いられるが,前処理として照明情報を点群として格納したフォトンマップを構築している.本手法ではこのフォトンマップを **3D API** によって物体表面に適用することで大域照明を実現している.また,これを応用することでフォトンマッピング法が可能な集光模様や,媒質中の光の散乱のレンダリングを実現し,最適化を加えることで大域照明のリアルタイムに迫るレンダリングを可能とした.

Abstract

Realistic rendering is one of the tendencies of research and development in graphics. In media such as movies or TV, realistic rendering has been enabled by results of study in this field. In realistic rendering, simulation of the light is important. Global Illumination is a field studying the simulation of light for Rendering.

Global Illumination can perform rendering of indirect lighting by the propagation of light between object surfaces. However, Global Illumination is unsuitable when interactive contents (such as in games) are necessary, because computational complexity is too great. Therefore, we render with rasterization of 3D API such as DirectX or OpenGL in interactive contents. On the other hand, many techniques of Global Illumination regard ray tracing as the basis. We encounter a problem to introduce Global Illumination into interactive contents, because the rendering techniques are different.

I realized Global Illumination rendering by 3D API in this article by using Photon Mapping which is one of the techniques of Global Illumination. Conventional Photon Mapping uses ray tracing for rendering, but builds a Photon Map, which stores illumination information as preprocessing in point group. Global Illumination was rendered in this study by applying Photon Map to the object surface by 3D API. In addition, I realized rendering of Caustics and Participating Media made using Photon Mapping by applying rasterization technique, and enabled rendering usually in real time of Global Illumination by adding optimization.

目次

第 1 章	序論	1
1.1.	背景	1
1.2.	何が描けるか	2
1.3.	本論文の構成	2
第 2 章	はじめに	4
2.1.	大域照明	4
2.1.1.	光の記述	4
2.1.2.	光の性質	7
2.1.3.	レンダリング方程式	11
2.1.4.	光の伝達表記	11
2.2.	レイトレーシング	12
2.2.1.	パストレーシング	14
2.3.	3D-API(DirectX,OpenGL)	14
2.3.1	GPU について	17
2.3.2	3D-API を用いた大域照明	17
2.4.	3D-API の問題点	18
第 3 章	フォトンマッピング法	19
3.1.	フォトンマッピング法 の概念	19
3.2.	フォトン追跡法	19
3.3.	フォトンの放射	20
3.4.	フォトンの散乱	21
3.5.	鏡面反射	22
3.6.	散乱反射	22
3.7.	ロシアンルーレット	22
3.8.	フォトンの格納	23
3.9.	フォトンマップのデータ構造	25
3.10.	放射輝度の推定	25
3.11.	光線追跡法を用いたフォトンマップの可視化と問題点	27
第 4 章	3D-API を用いたフォトンマッピング法の実装	28
4.1.	フォトンマッピング法 の改良	28
4.2.	3D-API を用いた放射輝度の推定	29
4.3.	結果	31

4.4.	集光模様	31
第 5 章	関与媒質への応用	33
5.1.	関与媒質中での光の散乱	33
5.2.	体積フォトンマップ	35
5.3.	体積放射輝度	36
5.4.	3DAPI による体積放射輝度の推定	38
5.5.	媒質の描画	38
5.6.	均一な媒質のレンダリング	39
5.7.	不均一な媒質のレンダリング	41
第 6 章	最適化	46
6.1.	レベルオブディテール	46
第 7 章	まとめと課題	51
7.1.	本論文のまとめ	51
7.2.	今後の課題	52

目次

1.1	本手法でレンダリングできる照明効果	2
2.1	直接照明と大域照明のレンダリング結果	4
2.2	放射輝度の定義	6
2.3	BRDFでの光の入射と反射	8
2.4	拡散面における光の反射	9
2.5	鏡面反射	10
2.6	屈折と反射のベクトル	10
2.7	光伝達表記法による経路の区別	12
2.8	レイトレーシングの処理	13
2.9	DirectXのパイプライン構成	15
2.10	一般的なレンダリング時の座標変換	16
2.11	ラスタライズによる変換	16
3.1	フォトン追跡法	20
3.2	棄却を伴うサンプリング法を用いた拡散点光源からフォトン を放射する疑似プログラム	21
3.3	フォトンマップに格納されたフォトンとレンダリング結果	24
3.4	放射輝度の推定	26
3.5	簡単なレイトレーシングプログラムを用いたフォトンマッピング法による 大域照明のレンダリング	27
4.1	放射輝度推定に使用するポリゴン	29
4.2	ポリゴンの変換	30
4.3	大域照明のレンダリング結果	31
4.4	集光フォトンマップに格納されるフォトン	32
4.5	ガラス玉による集光模様	32
5.1	関与媒質中でのフォトンの散乱	33
5.2	体積放射輝度の計算	38
5.3	均一な媒質中の光	39
5.4	各フォトンからの放射輝度	40
5.5	均一な媒質シーンのレンダリング結果	41

5.6	領域設定による不均一な媒質	42
5.7	視線を分割した場合のフォトンからの放射輝度	43
5.8	不均一時の $\sum_{p=1}^{N_{\text{all}}} e^{-(\sigma(x_f))} L_p(\mathbf{x}_p, \bar{\mathbf{w}})$ 計算プロセス	44
5.9	不均一な媒質シーンのレンダリング結果	45
6.1	LODの有無による描画速度の比較	48
6.2	リアルタイムレンダリング出来るまで精度を落とした結果	48
6.3	均一な媒質中のステージのレンダリング結果	50
6.4	リアルタイムレンダリング出来るまで精度を落とした結果	50

表目次

1.1 実行環境	3
1.2 使用する主な記号とその意味	3

第1章 序論

1.1. 背景

コンピュータグラフィックスで重要となるのが光源,物体表面,カメラの視点の三つである.特に写実的なレンダリングを行う場合,光源から出た光がどのように物体表面を伝搬し,カメラの視点に届いたかを計算することが重要となる.こうした光源からカメラの視点に届く光のふるまいを物理法則に基づいてシミュレートしたレンダリングを大域照明と呼んでいる.

コンピュータグラフィックスは現在様々な分野で用いられているが,扱い方を大きく分けるとオフラインシステムとリアルタイムシステムの二つに分けられる.オフラインシステムは,レンダリングにかかる時間に制約はないが,主に,高品質な映像が求められる映画や TV といった映像制作の分野で使用されている.オフラインシステムではレイトレーシング (2.6 節で述べる) がよく用いられ,何度も反射・屈折する光を CPU で時間をかけ計算し,高画質なレンダリングを行っている.大域照明の多くの手法がこのレイトレーシングを基とし,より厳密に光のふるまいを計算することで写実的なレンダリングを可能にしている.一方,リアルタイムシステムは,主にゲームのようなインタラクティブ性の高いコンテンツやシステムで利用されるため,高速なレンダリングが求められる.そのため,レンダリングにはレイトレーシングよりも高速なラスタライズ (2.7 節で述べる) を使うことが一般的である.また,ラスタライズによるレンダリングには DirectX や OpenGL といった 3D API を用いることができ,3D API をサポートするグラフィックスハードウェア (GPU) を導入することで,CPU で行うレイトレーシングに比べ,遥かに高速なレンダリングを可能としている.しかし,ラスタライズは高速な反面,光の反射を一度しか考慮しないため,光を何度も反射・屈折させる大域照明手法を適用することが難しい.

1.2. 何が描けるのか

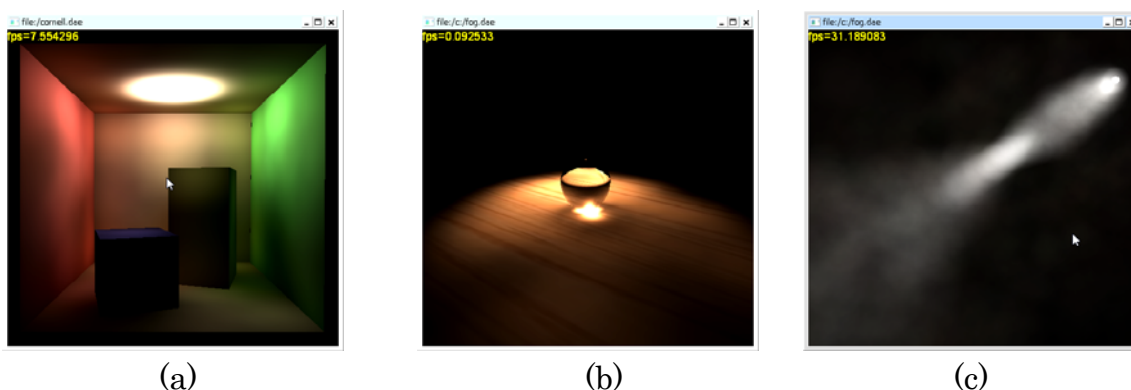


図 1.1 本手法でレンダリングできる照明効果

(a)間接照明.代表的な大域照明による照明効果.

(b)集光模様.反射・屈折した光が特定の場所に集中して起こる.

(c)媒質中の光の散乱.

本論文では,3DAPI を使った大域照明手法を提案する.本手法では大域照明の一つであるフォトンマッピング法を用いることで,図 1.1 に示す代表的な大域照明の効果である間接照明や,鏡面反射や屈折による集光模様,空気や霧などの関与媒質中で起こる光の散乱のレンダリングを 3DAPI によって可能にする.また,6 章で述べる最適化によって,条件次第ではこれらの効果のレンダリングをリアルタイムに行うことができる.

1.3. 本論文の構成

まず 2 章では,使用する 3DAPI や大域照明について述べるとともに,大域照明における光に関して述べる.3 章では,本論文の基となるフォトンマッピング法[1]のアルゴリズムについて述べる.つづいて 4 章では,3DAPI を用い,どのように大域照明を実現するかについて述べ,その結果を示す.5 章では,4 章で述べた手法を応用し,媒質中の光の散乱をレンダリングする手法について述べ,その結果を示す.6 章では 5 章までで述べてきた手法の最適化についてのべ,最適化した結果について述べる.最後に 7 章では,本論文のまとめと今後の課題について述べる.

本論文では実装したプログラムは表 1.1 に示す構成のマシンで実行している.また,本論文ではいくつかの記号を頻繁に用いて論を進めるため,これらを表 1.2 に整理して示す.

表 1.1 実行環境

OS	Windows Vista
CPU	Intel Core2 Quad 2.66GHz
メインメモリ	3069MB
グラフィックスカード	NVIDIA GeForce 8800 GTX
3DAPI	DirectX 10(August 2007)

表 1.2 使用する主な記号とその意味

記号	意味
\mathbf{x}	位置
\vec{n}	\mathbf{x} における法線 (常に正規化されている)
\vec{w}	方向 (表面から離れる方向)
$d\vec{w}$	立体角の微分
(θ, ϕ)	極座標系における方向
L	放射輝度
$L(\mathbf{x}, \vec{w})$	位置 \mathbf{x} における \vec{w} 方向の放射輝度
$L(\mathbf{x}, \vec{w}')$	位置 \mathbf{x} に \vec{w}' 方向から入射する放射輝度
L_e	放射された放射照度
L_r	反射された放射照度
L_i	入射する放射照度
Φ	放射束 (Flux)
E	放射照度
f_r	双方向反射率分布関数 (BRDF)
ρ	反射率
Ω	方向を表す半球
$\Omega_{4\pi}$	方向を表す球
σ_a	吸収係数
σ_s	散乱係数
σ_t	消滅係数
τ	光学的深度
ξ	0 から 1 の値をとる一様乱数

第2章 はじめに

2.1. 大域照明

大域照明は,物体の表面から表面へと反射もしくは屈折し,散乱する現実の光を,物理法則に基づいてシミュレーションし,その結果を画像としてレンダリングする手法である.直接照明によるレンダリング結果と,大域照明による間接照明のレンダリング結果を図 2.1 に示す.光と影がはっきり現れる直接照明に対し,大域照明では照らされた床や壁が2次的な光源となるため,間接照明の効果で全体に光が行き届き,写実的なレンダリングが行える.

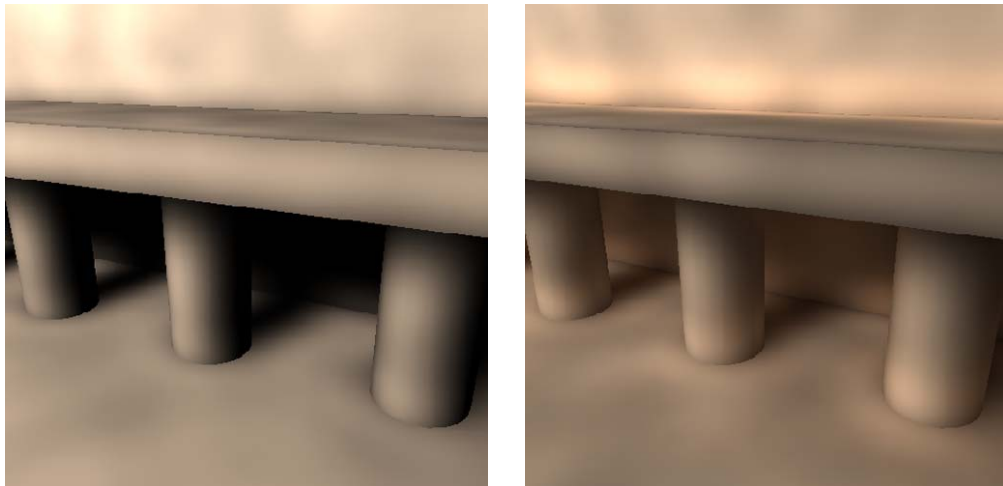


図 2.1 直接照明(左)と間接照明(右)のレンダリング結果

2.1.1. 光の記述

大域照明では光のシミュレーションが重要となるため,光を何らかの形で記述する必要がある.本項では光の記述に用いられる放射分析学[5]と観測学について述べる.

- 放射分析学

放射分析学での光の基本単位は光子(photon)である.波長 λ の光子のエネルギー e_λ は,

$$e_{\lambda} = \frac{hc}{\lambda} \quad (2.1)$$

である。ただし $h \approx 6.63 \cdot 10^{-34} \text{ J} \cdot \text{s}$ はプランク定数, c は光速である。(真空中では $c = c_0 = 299,792,458 \text{ m/s}$)

波長 λ の光子が n_{λ} 個あるとき, スペクトル放射エネルギー Q_{λ} は式(2.2)のようになる。

$$Q_{\lambda} = n_{\lambda} e_{\lambda} = n_{\lambda} \frac{hc}{\lambda} \quad (2.2)$$

放射エネルギー Q は光子の集団としてのエネルギーでありすべての波長にわたってスペクトルエネルギーを積分することで計算される。

$$Q = \int_0^{\infty} Q_{\lambda} d\lambda \quad (2.3)$$

放射束 Φ は放射エネルギーの時間変化である。

$$\Phi = \frac{dQ}{dt} \quad (2.4)$$

放射束面密度は放射束と物体表面における面積の微分 $d\Phi = dA$ と定義される。放射束面密度は物体表面から出ていく放射発散度 M と物体表面の位置 \mathbf{x} に入ってくる放射照度 E に分けられる。

$$E(\mathbf{x}) = \frac{d\Phi}{dA} \quad (2.5)$$

放射強度 I は単位立体角 $d\vec{\omega}$ あたりの放射束である。

$$I(\vec{\omega}) = \frac{d\Phi}{d\vec{\omega}} \quad (2.6)$$

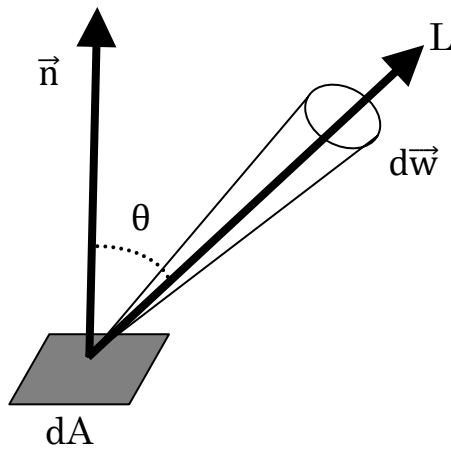


図 2.2 放射輝度の定義.

放射輝度 L は単位投影面積 dA あたりかつ単位立体角 $d\vec{\omega}$ あたりの放射輝度と定義されている.

図 2.2 に示す放射輝度 L は, 単位立体角あたりかつ単位面積当たりの放射束で, 式(2.7)のように定義される.

$$L(x, \vec{\omega}) = \frac{d^2\Phi}{\cos\theta dA d\vec{\omega}} \int_0^\infty \left\{ \frac{d^3n_\lambda}{d\vec{\omega} dAdt} \right\} \frac{1}{\lambda} d\lambda \quad (2.7)$$

放射輝度は物体の色をもっともよく表す概念であり, 大域照明において最も重要な物理量である. 放射輝度は特定の方向から小さな領域に単位時間に到達する光子の数と見なすことができ, 空間中の一点における一定方向の光の強さを記述することに利用できる. また, 真空中では光が進行しても放射輝度は変わらないという性質を, すべての光線追跡アルゴリズムが利用しているが, 5章で述べる関与媒質が存在する場合には適用できない.

式(2.7)により, 放射束から放射輝度を計算することができる. 物体表面における放射輝度の広がり分かれば, これをすべての方向 Ω と領域 A にわたって積分することにより放射束を計算することができる.

$$\Phi = \int_A \int_\Omega L(x, \vec{\omega}') (\vec{\omega}' \cdot \vec{n}) d\vec{\omega}' dx \quad (2.8)$$

ただし \vec{n} は物体表面の位置 x における法線である.

- 測光学

測光値には観察者の視覚応答やカメラのフィルムの感光が反映されておりこの点が放射分析学と測光学の違いである。

光束 Φ_v は放射束に対する視覚応答である。これは式(2.9)のように計算される。

$$\Phi_v = \int_{\Lambda} \Phi_{\lambda} V(\lambda) d\lambda \quad (2.9)$$

ただし $V(\lambda)$ は標準的な観察者の視覚応答であり, Λ は可視領域における波長である。

光束面密度 $d\Phi_v/dA$ は光が入射する場合には照度 E_v ,出射する場合は光発散度 M_v と呼ばれる。光度 I_v は単位立体角あたりの光束 $d\Phi_v/d\vec{\omega}$ であり,輝度 L_v は,

$$L_v(x, \vec{\omega}) = \frac{d^2\Phi_v}{\cos\theta dA d\vec{\omega}} \quad (2.10)$$

である。輝度は放射輝度と同等の測光学的概念であり,大域照明プログラムでしばしば利用される。

フォトンマッピング法[1]では光の物理量のみを扱うため放射分析学のみを利用するが,観察者による視覚応答は後処理として付加することができる。(色調写像法と呼ばれる)

2.1.2. 光の性質

光子の形態をとる光は光源で生成され,モデル中に放出され,物体表面にぶつかり,散乱もしくは吸収される。本節では, 3.2 節で後述するフォトン追跡法に関係する光の放射,光の散乱(拡散反射,鏡面反射),放射輝度の計算に関する双方向反射率分布関数(bidirectional reflectance distribution function.以下 BRDF)について述べる。

- 光の放射

光子の形態をとる光は,電球,太陽,化学反応のような光源で生成される。光源の強度(light intensity)は普通,光源の出力もしくはワット量(wattage)で与えられる。拡散点光源の場合はある物体表面における放射輝度 E を式(2.11)で計算することができる。

$$E(x) = \frac{\Phi_s \cos\theta}{4\pi r^2} \quad (2.11)$$

ただし, r は x から光源までの距離, θ は物体表面の法線と光源方向のなす角である. 光源から全方向に光子を放出する光源から離れるほど光子の密度は減少する. 減少する割合は光源からの距離 r における球の表面積 $4\pi r^2$ に比例する.

- 双方向反射率分布関数(BRDF)

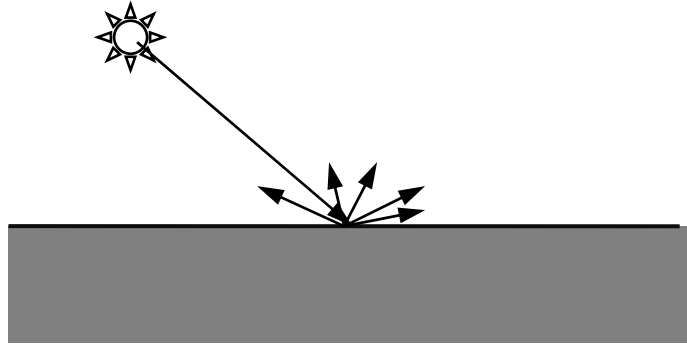


図 2.3 BRDF での光の入射と反射.

BRDF では光の入射と反射が同じ位置で起きると仮定している.

双方向反射率分布関数(BRDF)[6]は Nicodemus らによって導入された物体表面における光の反射を記述する手段で, BSSRDF(双方向散乱面反射率分布関数[6])の近似である. BRDF では物体表面のある位置でぶつかった光は同じ位置で反射すると仮定されている.

BRDF f_r は反射される放射輝度と放射照度の関係式(2.12)のように定義している.

$$f_r(x, \vec{w}', \vec{w}) = \frac{dL_r(x, \vec{w})}{dE_i(x, \vec{w}')} = \frac{dL_r(x, \vec{w})}{L_i(x, \vec{w}')(\vec{w}' \cdot \vec{n})d\vec{w}'} \quad (2.12)$$

ただし \vec{n} は位置 x における法線である.

BRDF は局所照明(local illumination)モデルを記述するもので, 物体表面上にある位置に入射する放射輝度が分かっているならば, それがあらゆる方向に反射していく放射輝度を計算することができる. これは入射する放射輝度 L_i を積分することによってなされる.

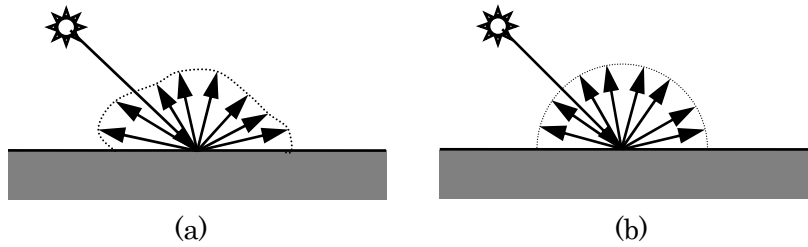
$$L_r(x, \vec{w}) = \int_{\Omega} f_r(x, \vec{w}', \vec{w}) dE(x, \vec{w}') = \int_{\Omega} f_r(x, \vec{w}', \vec{w}) L_i(x, \vec{w}') (\vec{w}' \cdot \vec{n}) d\vec{w}' \quad (2.13)$$

BRDF は光が進む方向に依存しない(ヘルムホルツ相反性の法則) という重要な性質がある.

$$f_r(\mathbf{x}, \vec{\omega}', \vec{\omega}) = f_r(\mathbf{x}, \vec{\omega}, \vec{\omega}') \quad (2.14)$$

光を両方向に追跡することを可能にする性質であるため,ほとんどの大域照明アルゴリズムで利用されている.

- 拡散反射



(a) (b)

図 2.4 拡散面における光の反射.

(a)一般的な拡散反射 (b)Lambert 反射

光が拡散反射する物体にぶつかった場合,物体表面で全方向に反射が起こる.この種の反射は主として図 2.4(a)のような光が無作為な方向に反射するあらい表面や微小面散乱において起こる.拡散反射の特別な場合として,図 2.4(b)に示すような光が入射する方向に関わらず,光が一様に拡散する Lambert 反射を考えることができる.このとき,反射される放射輝度は入射光の方向分布によらず,すべての方向に対して一定になる.これによって以下のように一定値の BRDF $f_{r,d}$ が得られる.

$$L_r(\mathbf{x}, \vec{\omega}) = f_{r,d}(\mathbf{x}) \int_{\Omega} dE_i(\mathbf{x}, \vec{\omega}') = f_{r,d}(\mathbf{x}) E_i(\mathbf{x}) \quad (2.15)$$

Lambert 反射する表面の場合光の反射方向は完全にランダムになる.二つの一様に分布した乱数を $\xi_1 \in [0,1], \xi_2 \in [0,1]$ とすると反射方向 $\vec{\omega}_d$ は

$$\vec{\omega}_d = (\theta, \phi) = (\cos^{-1}(\sqrt{\xi_1}), 2\pi\xi_2) \quad (2.16)$$

である.ただし,方向は球面座標 (θ, ϕ) を用いて表し, θ は物体表面の法線となす角度であり, ϕ は法線周りの回転角である.

- 鏡面反射

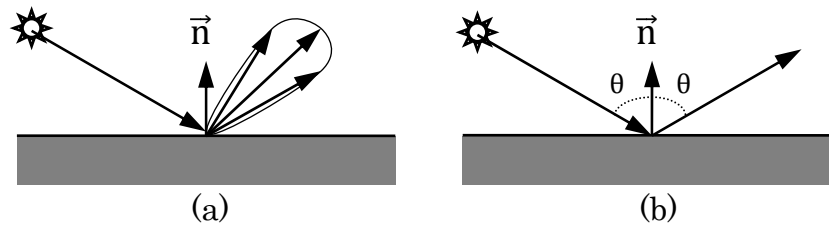


図 2.5 鏡面での鏡面反射.

(a)光沢反射(粗い鏡面) (b)完全な鏡面反射(鏡面)

滑らかな物体表面・金属やガラスや水などの誘電体などに光がぶつくと鏡面反射が起きる.ほとんどの物体表面は不完全であり,結果として鏡面反射方向を中心とする小さな円錐状に光の反射が起きる(図 2.5(a)参照).鏡面反射方向にしか光が反射しない完全に滑らかな物体表面では,完全な鏡面反射が起きる(図 2.5(b)).

鏡面反射によって反射された放射輝度は式(2.17)のようになる.

$$L_r(\mathbf{x}, \vec{\omega}_s) = \rho_s(\mathbf{x})L_i(\mathbf{x}, \vec{\omega}') \quad (2.17)$$

完全な鏡面反射における反射方向 $\vec{\omega}_s$ は式(2.18)のようになる.

$$\vec{\omega}_s = 2(\vec{\omega}' \cdot \vec{n})\vec{n} - \vec{\omega}' \quad (2.18)$$

ここで, $\vec{\omega}_s$ と $\vec{\omega}'$ は図 2.6 に示すとおり,物体表面から離れる方向を正とするベクトルである.

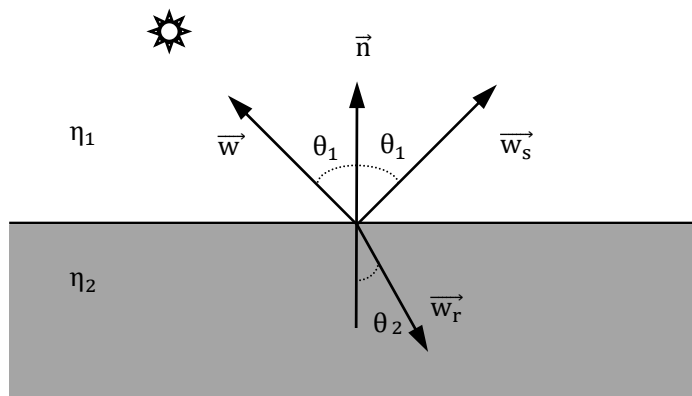


図 2.6 屈折と反射のベクトル

球面座標で方向をあらわした BRDF で完全な鏡面反射を喜寿すると式(2.19)のようになる.[6]

$$f_{r,s}(\mathbf{x}, \vec{\omega}', \vec{\omega}) = 2\rho_s \delta(\sin^2 \theta' - \sin^2 \theta) \delta(\phi' - \phi \pm \pi) \quad (2.19)$$

ただし Dirac のデルタ関数 $\delta(\mathbf{x})$ は BRDF が 0 にならない方向の範囲を制御するために用いている ($\delta(\mathbf{x})$ は $\mathbf{x}=0$ のときだけでない). $\vec{\omega} = (\theta, \phi)$ かつ $\vec{\omega}' = (\theta', \phi')$ である.

2.1.3. レンダリング方程式

レンダリング方程式はすべての大域照明アルゴリズムに対する数学的基礎を与えるものである. レンダリング方程式は関与媒質が存在しないモデル中における光伝達の平衡状態を提示している. レンダリング方程式はモデル中のあらゆる物体表面上において出射される放射輝度の計算に利用され, 出射される (outgoing) 放射輝度 L_0 は放射される (emitted) 放射輝度 L_e と, 反射される (reflected) 放射輝度 L_r の和である.

$$L_0(\mathbf{x}, \vec{\omega}) = L_e(\mathbf{x}, \vec{\omega}) + L_r(\mathbf{x}, \vec{\omega}) \quad (2.20)$$

式(2.13)を用いて反射される放射輝度を計算すると,

$$L_0(\mathbf{x}, \vec{\omega}) = L_e(\mathbf{x}, \vec{\omega}) + \int_{\Omega} f_r(\mathbf{x}, \vec{\omega}', \vec{\omega}) L_i(\mathbf{x}, \vec{\omega}') (\vec{\omega}' \cdot \vec{n}) d\vec{\omega}' \quad (2.21)$$

となり, このレンダリング方程式はしばしば多くの大域照明アルゴリズムで利用される.

2.1.4. 光の伝達表記

光が, 経路に沿って起こる異なる種類の反射を記述する方法に Heckbert の表記法がある [7]. Heckbert の表記法では光線経路の頂点が以下の状態となり得る.

- L 光源
- E 目
- S 鏡面反射
- D 拡散反射

経路の組み合わせを記述するために, 以下の正規表現を用いることが一般的である.

- (k)+ k が 1 回かそれ以上起きる
- (k)* k が 0 回かそれ以上起きる
- (k)? k が 0 回か 1 回起きる
- (k|k') k もしくは k' が起きる

例を図 2.7 に示す.

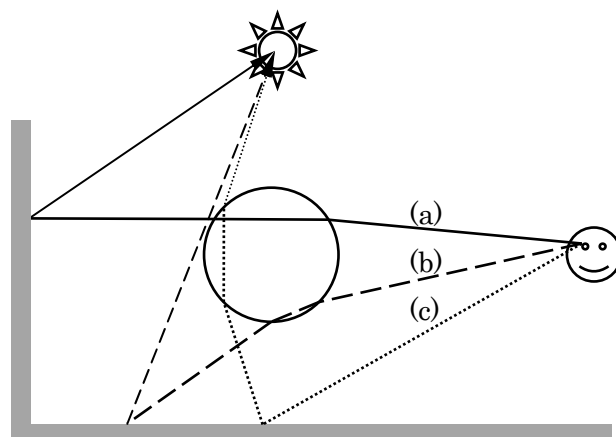


図 2.7 光伝達表記法による経路の区別.

ガラス球と 2 つの拡散壁からなるこのシーンで示されている経路は,
(a)LDSSE,(b)LDSE,(c)LSSDE である.

2.2. レイトレーシング

レイトレーシング(光線追跡法)はコンピュータグラフィックスの分野で,1980 年に導入された再帰的光線追跡法によって広く利用されるようになった[8].レイトレーシングは影や鏡面反射を的確に簡略化した簡潔なアルゴリズムである.単純なレイトレーシングでは光伝達の表記法では

LD?S*E

しか計算できないが(拡散面における間接照明や不完全な鏡面反射物質,ソフトシャドウ,カメラのフォーカス効果,モーションブラーは計算できない),完全な大域照明計算(LS|D)*Eが可能なパストレーシング [9]はレイトレーシングを拡張したものであり,フォトンマッピング法では放射輝度の推定にレイトレーシングを利用するなど,大域照明におけるアルゴリズムの基となる重要な手法である.

自然界では光源から放出された光子がシーン中を散乱し,その後一部の光子が目に届く.しかし,この過程をまじめに計算することは現実的ではない.いっぽう,光子は真空中を進行しても輝度は変化しないという性質と,BRDF では物体表面における光は対称性を持って散乱するという特質により,観察者から光源に向かって逆方向に光を追跡することが可能である.レイトレーシングはこの性質を利用し,レンダリングを行う.

シーンを画像としてレンダリングするには画像の各画素の色（平均の放射輝度）の計算が必要である。レイトレーシングでは各画素に対し、1本以上の光線（レイ）を追跡し得られた放射輝度の平均値を計算することで画像を各画素の色を計算する。観察者から各画素を通過していくレイは1次レイと呼ばれ、レイ r は式(2.22)のように記述される。

$$r(x, \vec{w}) = x + d \cdot \vec{w} \quad (2.22)$$

ただし、 x はレイの出発点(観察者)、 \vec{w} はレイの方向、 d はレイに沿った距離である。

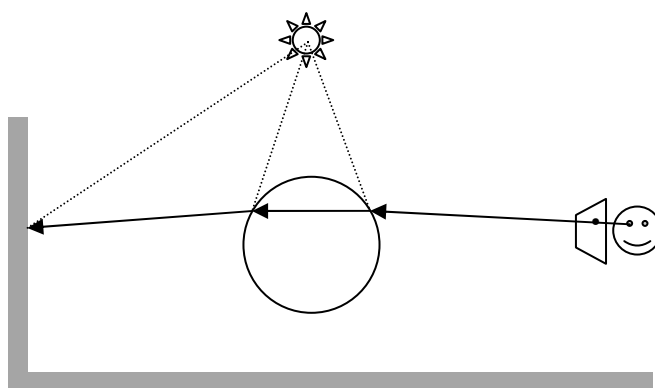


図 2.8 レイトレーシングの処理

観察者から光を逆方向に追跡し、物体表面との交差点で直接照明を計算する。
また、交差点が影か否かはシャドウレイ(点線)を使い検査する。

1次レイの放射輝度を計算するためには、レイと交差する最近傍の(d が最小となる)物体の表面を検索する必要がある。(図2.8の例では球面)

交点 x を求めた後、レイの方向に出ていく放射輝度を求める。これには BRDF 項 f_r の他、位置 x における法線 \vec{n} が用いられる。例えば位置 p に出力 Φ_1 の点光源があるときの反射される放射輝度 L_r は式(2.23)で計算される。

$$L_r(x, \vec{w}) = f_r(x, \vec{w}, \vec{w}') \frac{\vec{w}' \cdot \vec{n}}{\|\vec{p} - x\|^2} V(x, p) \frac{\Phi_1}{4\pi} \quad (2.23)$$

ただし $\vec{w}' = (\vec{p} - x) / \|\vec{p} - x\|$ は光源方向の単位ベクトル、 V は可視性を表す。可視性 V は位置 x から光源方向に向うシャドウレイと呼ばれるレイを新たに追跡することで、位置 x と光源の間の物体とシャドウレイが交差する場合は $V=0$ (位置 x は影の中)、そうでなければ $V=1$ となる。

レイトレーシングでは鏡面反射方向 \vec{w}_s (式(2.18)) のレイを追跡して鏡面を扱う。この反射・屈折されたレイ(2次レイ)を放射輝度の計算方法は1次レイの方法と同じである。

2.2.1. パストレーシング

パストレーシング(Path Tracing) [9]はレイトレーシングを拡張し,大域照明を計算できるようにした大域照明手法である.レイトレーシングでは放射輝度計算で入射する光を光源に限定するが,パストレーシングでは間接光など全方向からの入射光を使用する.全方向から入射する光はモンテカルロサンプリングを用い計算するため,モンテカルロレイトレーシングとも呼ばれる

モンテカルロサンプリングは,解析的に解けない問題を乱数を用い,十分な回数のシミュレーションを行うことで近似的な解を求める方法である.パストレーシングでは,計算する物体の表面から無作為に選ばれた方向に2次レイを複数発生させ,追跡した先の物体表面でも同様のサンプリングを行うことで全方向から入射する光を求め,放射輝度を計算することで大域照明をレンダリングする.

パストレーシングは完全な大域照明を計算できる一方,拡散面で複数のレイを発生させるため,レイトレーシングより遥かに高負荷である.また,モンテカルロサンプリングでは,十分なサンプリングを行わなければノイズが発生してしまうため散乱面で発生させるレイの数を抑えることが難しい.

2.3. 3DAPI (DirectX, OpenGL)

3Dグラフィックスの描画機能を有したAPI(Application Program Interface)を3DAPIと呼ぶ.3DAPIには様々なものが存在するが,DirectXやOpenGLが代表的である.本節では本論文の実装でも用いている DirectX の ver.10 を例に3DAPIでのレンダリングについて述べる.

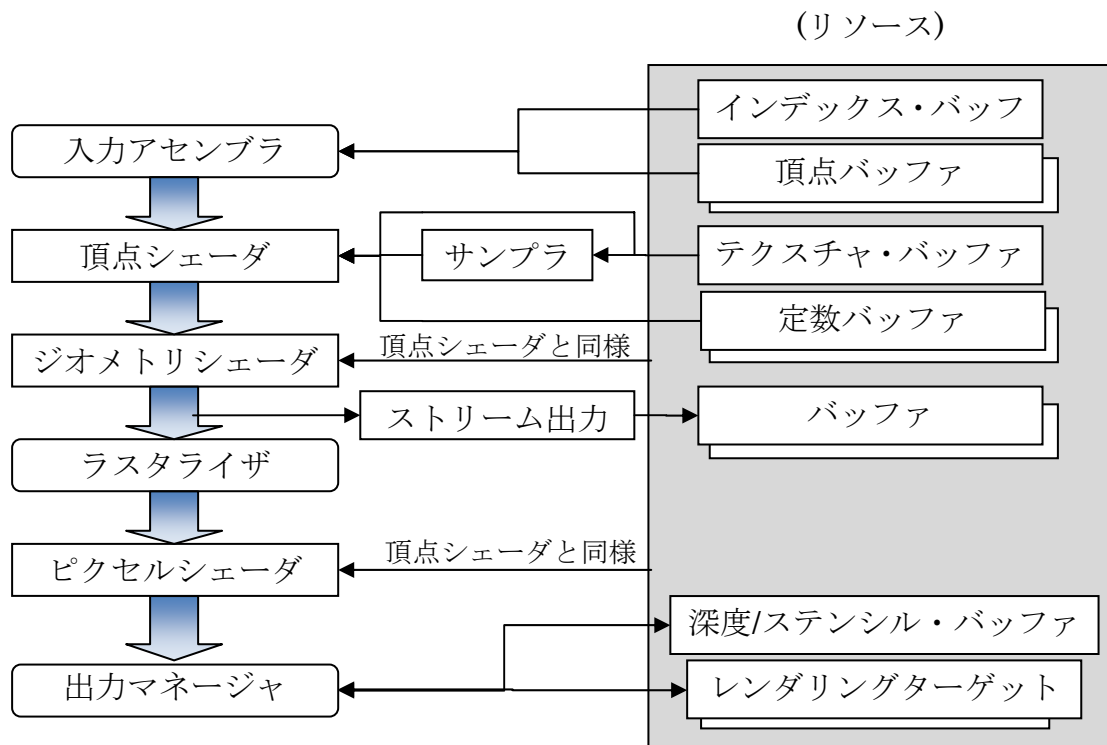


図 2.9 DirectX のパイプライン構成

DirectX ではレンダリングをパイプラインで処理する. パイプラインの構成は図 2.9 のように複数のステージで構成されている. アプリケーションはこれらのステージの動作の設定とリソースの準備を行い, レンダリングメソッドを実行することでデータが入力アセンブラから出力マネージャへと流れ, 最終的にレンダリングターゲットに画像が出力される. 詳細は参考文献[4]を参照されたい.

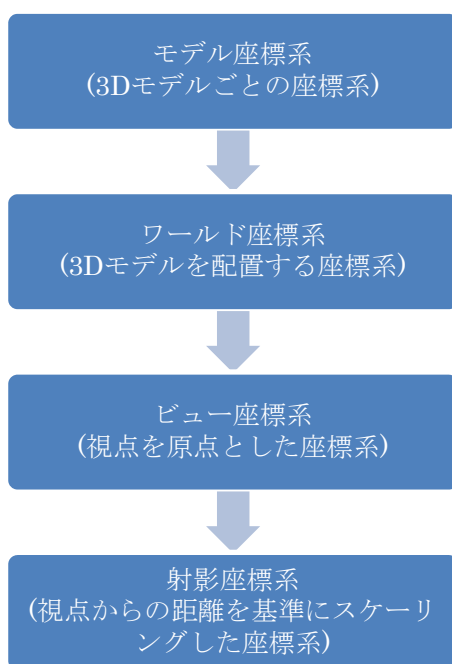


図 2.10 一般的なレンダリング時の座標変換

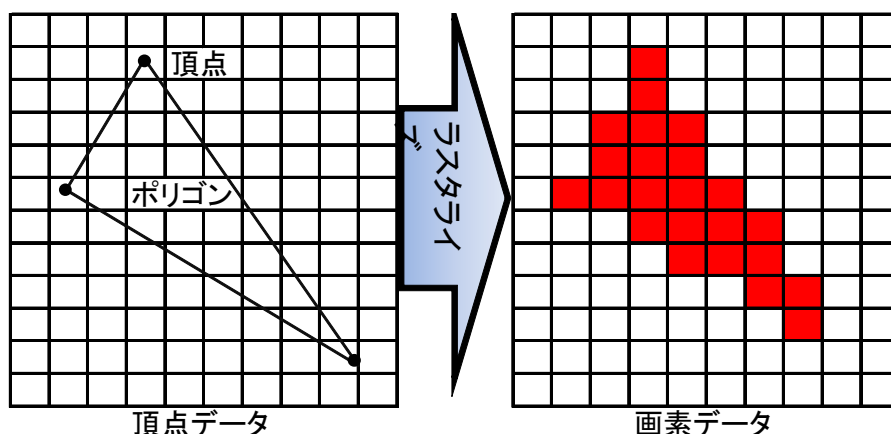


図 2.11 ラスタライズによる変換

レイトレーシングと 3D API のレンダリングの概念の大きな違いは、レイトレーシングが観察者の視線の先あるものがどのように見えるかを計算するのに対し、3D API はある 3D モデルが画面上のどこにあり、どんな色で表示されるかを計算することである。標準的なレンダリングでは、以下の過程で計算される。頂点シェーダ及びジオメトリシェーダで、頂点データを図 2.10 のように座標系を変換しラスタライザへ頂点データを出力、ラスタライザで頂点データをスクリーン座標系(ピクセル単位のスクリーン上の位置)に変換後、ラスタライズ (図 2.11 参照) によって画素データを出力し、ピクセルシェーダで画面上の色を計算、結果を出

コマネージャがレンダリング先の画像バッファ（レンダリングターゲット）に書き込み画像を出力する。DirectX の面(ポリゴン)の描画対象は三角形のみであるため、レンダリングするモデルの表面を三角形ポリゴンに分割する必要がある。ここで、複数のポリゴンが画面上の同じ位置で重なった場合、視点から見て手前のポリゴンが画面に表示されなければならない。この問題の解決には Z バッファ（深度バッファ）が用いられる。Z バッファにはレンダリングされたポリゴンの深度値が記録されている。新たにレンダリングされるポリゴンは、レンダリングターゲットに記録される前に Z バッファの深度値を評価して、レンダリング済みのポリゴンよりも手前にある時のみレンダリングターゲットと Z バッファを更新する。この処理によって常に手前にあるポリゴンを表示することが出来る。

3DAPI は以上のような処理でレンダリングを行い、レイトレーシングにおける 1 次レイのみのレンダリング結果に相当する画像を出力することが出来る。しかし、座標変換によるレンダリング手法であるため、反射、屈折によって 1 次レイとは別方向に向かう 2 次レイ以降の光線追跡が行えないという欠点がある

2.3.1. GPU について

3DAPI の特徴の一つにハードウェアの存在がある。特に DirectX や OpenGL をサポートするハードウェアは、現在ではほとんどの PC に組み込まれるほど普及し、一般的にこのようなハードウェアを GPU(Graphics Processing Unit)と呼んでいる。

GPU は 3DAPI によるレンダリングに特化したハードウェアであると共に、複数のパイプラインを持ち、並列処理が可能のため、CPU によるレンダリングよりも遥かに高速なレンダリングが可能である。近年では頂点シェーダ、ジオメトリシェーダ、ピクセルシェーダはプログラミング可能なユニット(Programmable Shader)になっており、ハードウェアの汎用性が増している。一方で、種類によるが、GPU は命令発行ユニットに比べ実行ユニットが多いという構造上、複数の実行ユニットに同じ命令が出来なくなってしまう動的な分岐処理が苦手である[3]。このため、データベースから情報を検索するような処理を苦手としている。

2.3.2. 3DAPI を用いた大域照明

3DAPI を用いてレンダリングが出来る大域照明に PRT[10]がある。PRT はレンダリング前に大域照明の照明効果を計算し、テクスチャとして持つておくことで間接照明のような大域照明をレンダリングすることができる。PRT で行う照明効果の事前計算は 3D モデルの各頂点で行うため 5 章で述べるような表面のない物体や 6 章で述べる集光模様への拡張が難しい..

2.4. 3DAPI の問題点

以上述べたように 3DAPI は, GPU のサポートもあり, CPU に比べ高速なレンダリングを行える. 一方で制約も多く, 汎用性では CPU に劣る. この制約によって, いくつかのコンピュータグラフィックスの手法は, 3DAPI を使ったレンダリングに適用することが困難である. 特に大域照明に関しては多くの大域照明手法がレイトレーシングを基としているため, レイトレーシングにおける 2 次レイ以降の追跡を行えない 3DAPI ではこうした手法を用いたレンダリングは難しい.

本手法では, 2 次レイ以降の効果を 3 章で述べるフォトンマッピング法を用いることで解決し, 4 章で述べる方法によって 3DAPI 上で大域照明の描画を行う.

第3章 フォトンマッピング法

本章ではフォトンマッピング法の概要と一般的な実現を述べる。

3.1. フォトンマッピング法の概要

フォトンマッピング法は光線を光源から飛ばされる粒子（フォトン）に置き換え、フォトンの経路を計算することで光線のシミュレーションを行う大域照明の手法である。フォトンマッピング法では 1.2 節で挙げたような間接照明や媒質、微小面散乱、コースティクス等を扱うことができる。

光の経路を計算するレンダリング手法にはフォトンマップの他にパストレーシング[9] などがあるが、視線と光路を同時に処理するパスレイトレーシングに対し、フォトンマッピングは光路(フォトン追跡)と視線の経路(レンダリング)の処理を別々に行う二段階手法をとり、光路とレンダリングを切り離すことで効率的な光路計算を可能にしている。

3.2. フォトン追跡法

フォトン追跡法は(photon tracing)は、光源からフォトンを放出し、それを追跡する過程であり、フォトンマップ（詳細は 3.3 節で述べる）を構築するために利用される手法である。概念を図 3.1 に示す。本節ではフォトン追跡法がどのように光源からフォトンを生成し、効率的に追跡するかについて述べる。

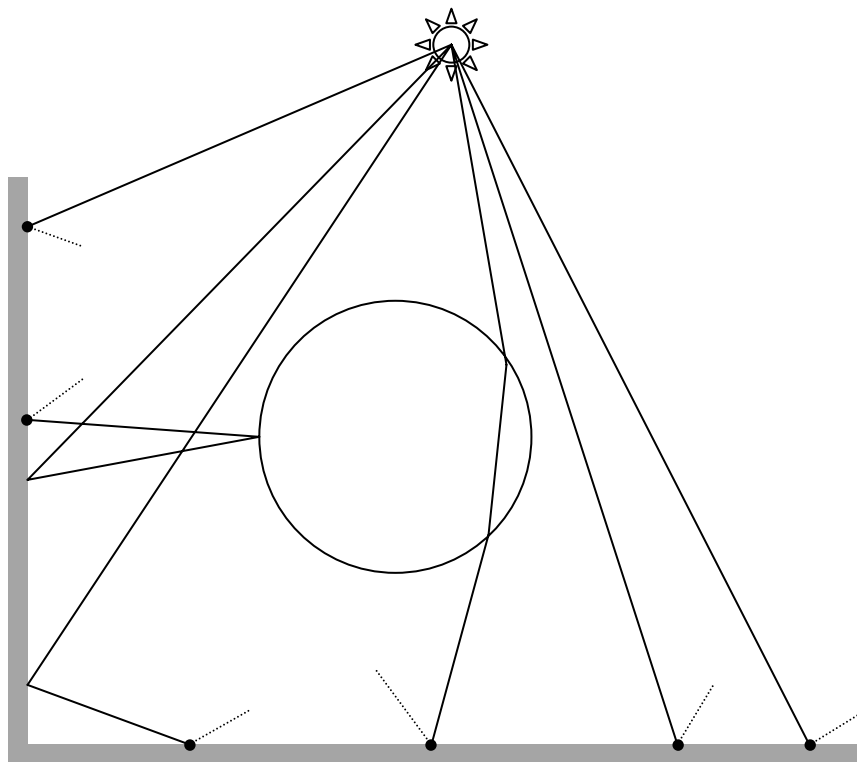


図 3.1 フォトン追跡法
フォトンは光源から放出され、散乱面と交差したとき
フォトンマップに格納される。

3.3. フォトンの放射

フォトンはシーン中の光源で生成される。光源には、コンピュータグラフィックスの分野で典型的な光源である点光源や指向性光源、面光源、および（任意形状や分布を実現する）物理モデルに基づく光源を用いることができる。

自然界ではそれぞれの光源から膨大な数のフォトンが放射され、光源の明るさ（単位はワット）は放出されたフォトンに割り振られ、それぞれのフォトンが光源の明るさの一部を伝達する。

次に本研究で使用している拡散点光源について述べる。拡散点光源は光源に設定されているある点から全方向に一樣にフォトンが放射される光源の中でもっとも単純なものの一つである。フォトンの生成にはモンテカルロサンプリング(2.2.1 節参照)を利用する。球面を一樣にサンプリングするための主要な手法としては明示的なサンプリングと棄却を伴うサンプリングの二種類がある。

明示的なサンプリング法では球面上の位置に乱数を対応させる。典型的なやり方とし

では、角度を無作為にサンプリングすることである[11].棄却を伴うサンプリング法では特定の性質が現れるまで乱数の評価を繰り返す.拡散点光源の場合,単位立方体内部に乱数を発生させ単位立方体内部の点が単位球の内部に入っている場合をフォトンの放射方向とする.棄却を伴うサンプリング法の擬似プログラムを図 3.2 に示す.

```

emit_photons_from_diffuse_point_light(){
  ne = 0 放射されるフォトンの数.
  while ( not enough photons )
    u { 棄却を伴うサンプリング法でフォトンの方向を決める.
      x = 2ξ1 - 1   ξ1 ∈ [0,1]は乱数
      y = 2ξ2 - 1   ξ2 ∈ [0,1]は乱数
      z = 2ξ3 - 1   ξ3 ∈ [0,1]は乱数
    } while ( x2 + y2 + z2 > 1 )
     $\vec{d} = \langle x, y, z \rangle$ 
     $\vec{p}$  = photon source position
    trace photon from  $\vec{p}$  in direction  $\vec{d}$ 
    ne = ne + 1
  }
  scale power of stored photons with 1/ne
}

```

図 3.2 棄却を伴うサンプリング法を用いた拡散点光源からフォトン放射する擬似プログラム

3.4. フォトンの散乱

フォトン放射後はフォトン追跡法を用いたフォトン追跡を行う.フォトン追跡法は光線の放射輝度を束ねた放射束をフォトンが伝搬している点を除いて光線追跡法とまったく同様に機能する.フォトンと物質の相互作用は光線とは異なるため,例えば屈折では相対的な屈折率に基づいて放射輝度に変化するが,これはフォトンでは起こらない.

フォトンが物体にぶつくと,反射(reflected),透過(transmitted),吸収(absorbed)のいずれかが起こる.反射・透過・吸収のどれが起こるかは物体表面の材質に基づいて確率的に決定される.相互作用の種類を決めるために利用される手法はロシアンルーレットとして知られている.以下では異なる種類の物質に対していかにフォトン反射させるか,ロシアンルーレットを用いていかに効率的にフォトン扱うかについて述べる.

3.5. 鏡面反射

フォトンが鏡面にぶつかると新しいフォトンが鏡面反射方向 (mirror direction) に反射される. 法線 \vec{n} と入射方向 \vec{w}' が与えられれば反射方向 \vec{w} は式(3.1)のようになる.

$$\vec{w} = 2(\vec{n} \cdot \vec{w}')\vec{n} - \vec{w}' \quad (3.1)$$

ここで, 入射方向は交差する点から離れる方向を正としている. 式(3.1)は光線追跡法において鏡面反射した光線の追跡でも用いられる. フォトンが鏡面反射する場合はフォトンマップには格納されず, ロシアンルーレットを適用しない場合は反射されたフォトンの出力を鏡面の反射率を用いて計算する.

3.6. 拡散反射

フォトンが拡散反射面にぶつかるとフォトンマップに格納され, 交差した地点の半球内において法線となす角度の余弦に比例する確率に従って無作為に拡散反射されたフォトンの方向が決められる ((2.16)式). ロシアンルーレットを適用しない場合, 反射されたフォトンの出力は入射フォトンの出力に拡散反射率を掛けることで求められる.

3.7. ロシアンルーレット

ロシアンルーレットは, 重要でないフォトンを取り除き, 重要なフォトンに処理を集中させるためにフォトン追跡法で利用される重要な手法である. 計算量を減らすために効率的なサンプリングを行っても正しい結果が得られるというのがロシアンルーレットの基本的な考え方である. 重要でない部分を取り除くため確率密度関数を用いる重点的サンプリング法の一つと考えることができる. 放射輝度推定値が L_n になる確率が p である時 L_n の値は式(3.2)のようになる.

$$L_n = \begin{cases} \frac{L}{p} & (\xi < p \text{ のとき}) \\ 0 & (\xi \geq p \text{ のとき}) \end{cases} \quad (3.2)$$

ここで L は通常は光線追跡法によって計算される. 推定量(estimator) L の期待値を式(3.3)のように計算することによって, これが正しく機能することを確認できる.

$$E\{L\} = (1 - p) * 0 + p * \frac{E\{L\}}{p} = E\{L\} \quad (3.3)$$

このように、ロシアンルーレット法によって、 L に関して統計的に偏りのない正しい推定を、無限回の反射を追跡せずに有限回の追跡で行うことができる。

次に前述の鏡面反射,拡散反射へのロシアンルーレットの適用について述べる.反射率 d の物質に出力 ϕ_d の光子が入射するとき,光子の反射と吸収のいずれかが起きるのかをロシアンルーレットを使って決めることができる.これは以下の手順で行われる.

```
p=d
ξ = random()
if(ξ < p)
  reflect photon with power φp
else
  photon is absorbed
```

このアルゴリズムの効果は,例えば反射率 0.5 の物体表面に 1000 個の光子を当てた場合,反射される光子が 1000 個(出力は半分になっている)に対し,ロシアンルーレットを適用することで反射される光子を 500 個(出力は変わらない)に削減し,光子追跡法の計算量を減らすことができる.これを鏡面反射と拡散反射の両方が起きる物体表面に適用した場合,拡散反射率を ρ_d ,鏡面反射率を ρ_s ,ただし $\rho_d + \rho_s \leq 1$ とした場合,一様散乱 $\xi \in [0,1]$ を用いると以下のようにして決められる.

```
ξ ∈ [0, ρd] → 拡散反射
ξ ∈ ]ρd, ρs + ρd] → 鏡面反射
ξ ∈ ]ρs + ρd, 1] → 吸収
```

この場合も,光子が反射されたときに光子の出力の修正は行わない。

以上のようにロシアンルーレット法を用いることで光子追跡法の計算量を軽減し,それぞれの光子の出力が揃うことで,光子マップの統計的性質を向上させることができる.一方でロシアンルーレットを適用することで解の分散を増加させてしまう.このため十分な数の光子を用いなければ正しい結果に収束しない。

3.8. フォトンの格納

光子マップ(photon map)はレンダリングするモデル中を散乱した光子を記録した分布図である.前述の光子追跡法によって光子が拡散面に到達した場合にのみ光子は光子マップに格納される.ここで,鏡面に到達した場合に光子を格納しないのは,鏡面反射方向から入射する光子

が得られる確率が小さいためである。鏡面反射を正確に描画したい場合は、標準的な光線追跡法を使い、鏡面反射方向の光線を追跡することが最良の手段になると考える。それ以外の物体表面とフォトンの相互作用に対しては大域的なデータ構造であるフォトンマップの中にデータを蓄積する。

図 3.3 にコーネルボックスのシーンでフォトンマップに格納されたフォトンを示す。

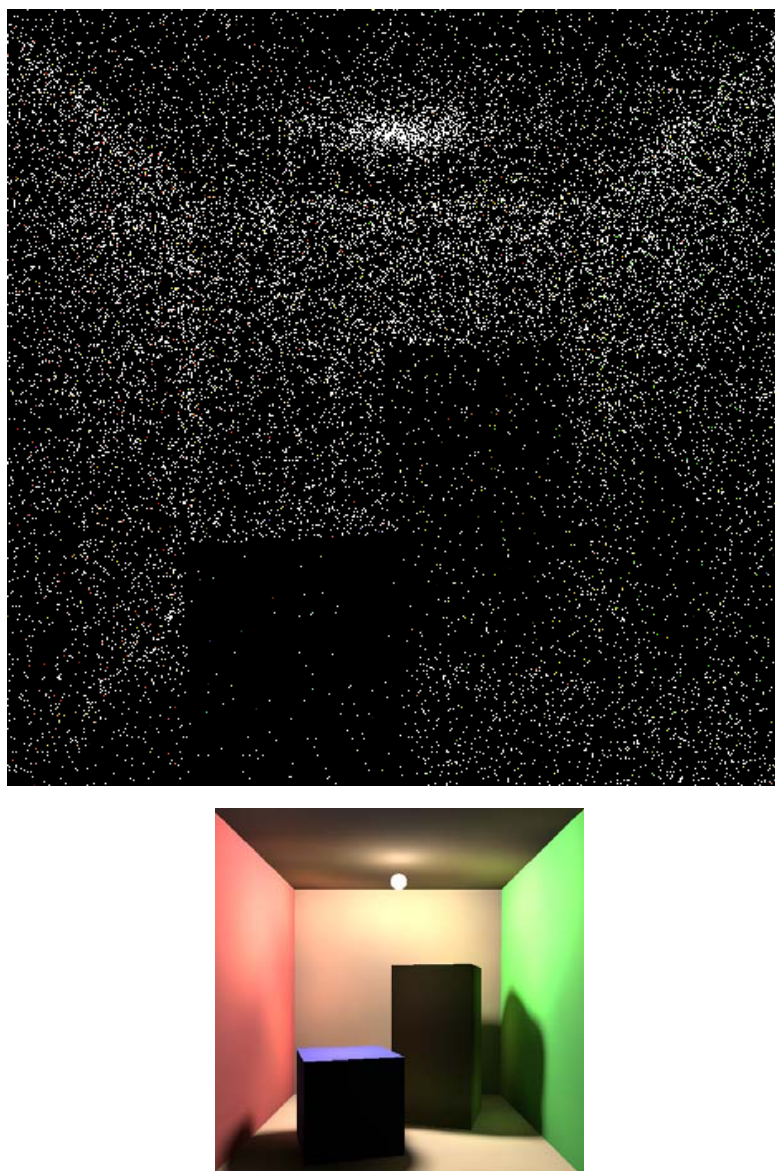


図 3.3 フォトンマップに格納されたフォトン(上)とレンダリング結果(下).

3.9. フォトンマップのデータ構造

フォトンマップは、フォトン追跡法によって格納されたフォトンの分布図であり、モデルの幾何学形状と切り離された別のデータ構造で保持されている。フォトンマップはモデル中の照明計算に利用され、注目点に最も近い位置にあるフォトン（最近傍フォトン）を用いて計算する。後述の光線追跡法を用いたレンダリングでは、この最近傍フォトンを探る処理を頻繁に行うため、フォトンマップはこの検索を高速に行えるデータ構造が望ましい。Jensen らはこのデータ構造に kd-木を用い、簡潔な記述法で高速な最近傍検索が可能なデータ構造を実現している[1]。

フォトンマップに記録されているそれぞれのフォトンは、フォトンと物体表面が交わった時の位置、入射フォトンの出力、入射方向の情報と kd-木で使われるフラグを持っている。

フォトンマップの kd-木の詳細や最近傍フォトンの検索の詳細については文献[1] を参照されたい。

3.10. 放射輝度の推定

物体表面上のある位置 \mathbf{x} で $\vec{\omega}$ 方向に反射される放射輝度 L_r は式(2.13)より

$$L_r(\mathbf{x}, \vec{\omega}) = \int_{\Omega_x} f_r(\mathbf{x}, \vec{\omega}', \vec{\omega}) L_i(\mathbf{x}, \vec{\omega}') (\vec{n}_x \cdot \vec{\omega}') d\vec{\omega}' \quad (3.4)$$

で求められる。ここで Ω_x は入射方向を表す半球で、 f_r は位置 \mathbf{x} における BRDF、 L_i は入射する放射輝度である。フォトンマップは入射する放射輝度に関する情報であるため放射輝度と光束の関係(式 2.10)から

$$L_i(\mathbf{x}, \vec{\omega}') = \frac{d^2\Phi_i(\mathbf{x}, \vec{\omega}')}{(\vec{n}_x \cdot \vec{\omega}') dA_i} \quad (3.5)$$

であるため、積分式は式(3.6)のようになる。

$$\begin{aligned} L_r(\mathbf{x}, \vec{\omega}) &= \int_{\Omega_x} f_r(\mathbf{x}, \vec{\omega}', \vec{\omega}) \frac{d^2\Phi_i(\mathbf{x}, \vec{\omega}')}{(\vec{n}_x \cdot \vec{\omega}') dA_i} (\vec{n}_x \cdot \vec{\omega}') d\vec{\omega}' \\ &= \int_{\Omega_x} f_r(\mathbf{x}, \vec{\omega}', \vec{\omega}) \frac{d^2\Phi_i(\mathbf{x}, \vec{\omega}')}{dA_i} \end{aligned} \quad (3.6)$$

入射光束 Φ_i を位置 \mathbf{x} に最も近い n 個のフォトンを探ることで位置 \mathbf{x} に入射する光の近似を行う。各フォトン \mathbf{p} は出力

$\Delta\Phi_p(\vec{w}_p)$ を有し、光子と物体表面が位置 \mathbf{x} で交差すると仮定すると式(3.7)が得られる。

$$L_r(\mathbf{x}, \vec{w}) \approx \sum_{p=1}^n f_r(\mathbf{x}, \vec{w}_p, \vec{w}) \frac{\Delta\Phi_p(\mathbf{x}, \vec{w}_p)}{\Delta A} \quad (3.7)$$

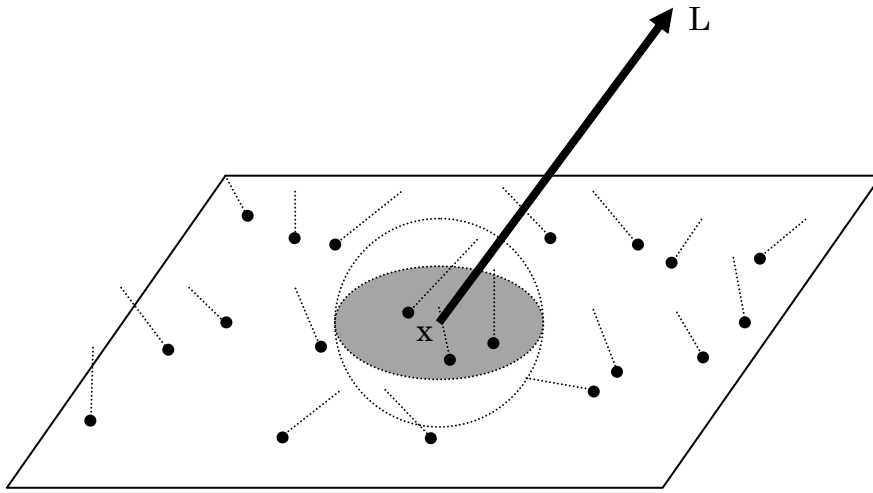


図 3.4 放射輝度の推定

位置 \mathbf{x} の最近傍光子を光子マップから見つけ出し、局所的な光子の密度を推定することで反射された放射輝度を計算する。

物体表面が位置 \mathbf{x} 付近で局所的に平らであると仮定すると、 ΔA は図 3.4 中の灰色の領域となり、式 (3.8) で表せる。

$$\Delta A = \pi r^2 \quad (3.8)$$

ただし、 r は位置 \mathbf{x} から各光子までの最大距離である。これらをまとめると物体表面のあらゆる位置において反射される放射輝度の推定値 (radiance estimate) を光子マップを用いて計算する式(3.9)が得られ、画像合成が可能になる。

$$L_r(\mathbf{x}, \vec{w}) \approx \frac{1}{\pi r^2} \sum_{p=1}^N f_r(\mathbf{x}, \vec{w}_p, \vec{w}) \Delta\Phi_p(\mathbf{x}, \vec{w}_p) \quad (3.9)$$

3.11. 光線追跡法を用いたフォトンマップの可視化と問題点

一般的にフォトンマップは光線追跡プログラムを用いて可視化される。光線追跡プログラムでは, 図 3.5 に示すように拡散反射する素材に対する反射計算にフォトンマッピング法による放射輝度推定を利用し, 鏡面反射する素材に対しては標準的な再帰的光線追跡を行う。(詳細については文献[1]参照)

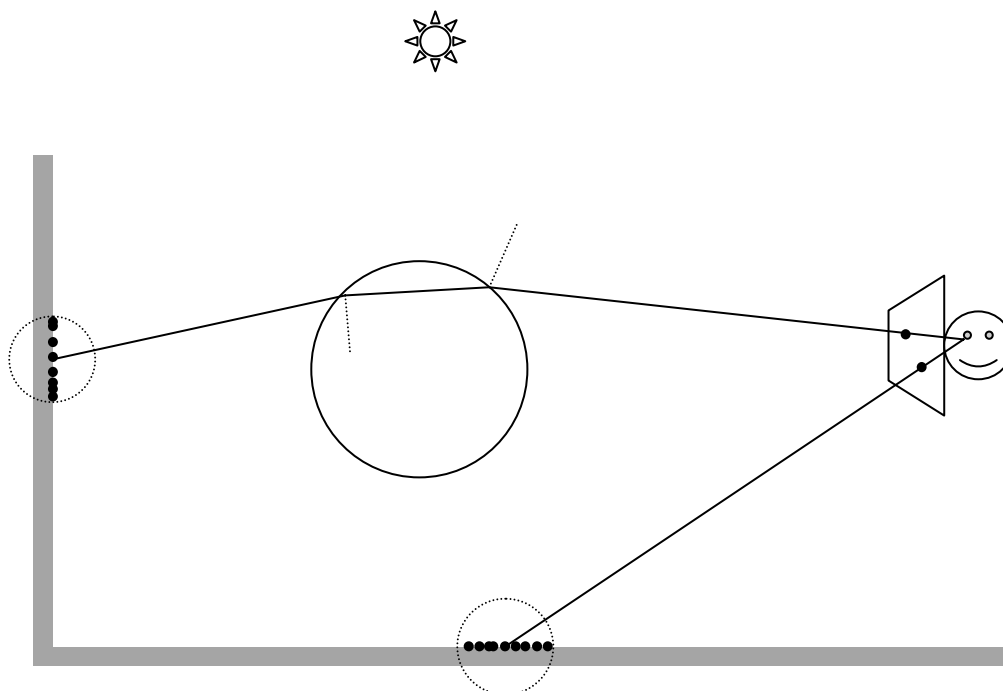


図 3.5 簡単なレイトレーシングプログラムを用いた,
フォトンマッピング法による大域照明のレンダリング.
鏡面には標準的なレイトレーシングを適用し,
拡散面に対してフォトンマップによって推定された放射輝度を適用する.

この方法を用いた場合, 光線追跡による反射屈折の描画とフォトンマッピング法による大域照明を描画でき, 非常に高画質な画像を出力することができる. 一方で光線追跡法の計算量が非常に多い. 次の章では 3D API と GPU を用いることで計算コストの低減と並列処理によるコストの分散を図った手法について述べていく.

第4章 3DAPI を用いた フォトンマッピング法の実装

本章では, 3章で述べたフォトンマッピング法の改良点と,実装方法,および結果を示す.

4.1. フォトンマッピング法の改良

2.3.1節で述べたように GPU には検索処理は向かないため,近傍フォトンの検索処理を簡略化するための改良を行った.3章の式(2.9)において,検索に関係しているのは,フォトンマップから見つけ出す最近傍フォトンの個数 N と位置 \mathbf{x} から最近傍フォトンまでの最大距離 r である.そこで,まず r を定数とし,ユーザーが設定するパラメータとした.次に N をフォトンマップに記録されているすべてのフォトンの数 N_{all} とした.これにより放射輝度の推定値は式(4.1)で求められる.

$$L_r(\mathbf{x}, \vec{\omega}) \approx \frac{1}{\pi r^2} \sum_{p=1}^{N_{\text{all}}} f_r(\mathbf{x}, \vec{\omega}_p, \vec{\omega}) \Delta\Phi_p(\mathbf{x}, \vec{\omega}_p) w_p(d_p) \quad (4.1)$$

ここで d_p は位置 \mathbf{x} とフォトン p の距離, w_p は d_p が最長距離 r を超えた場合評価されないようにするため,

$$w_p(d_p) = \begin{cases} 1 & (d_p \leq r \text{ の場合}) \\ 0 & (d_p > r \text{ の場合}) \end{cases} \quad (4.2)$$

とする.これによりフォトン p が放射輝度の推定値に与える影響 L_{rp} を

$$L_{rp}(\mathbf{x}, \vec{\omega}) = \frac{1}{\pi r^2} f_r(\mathbf{x}, \vec{\omega}_p, \vec{\omega}) \Delta\Phi_p(\mathbf{x}, \vec{\omega}_p) w_p(d_p) \quad (4.3)$$

とでき,放射輝度の推定値を

$$L_r(\mathbf{x}, \vec{\omega}) \approx \sum_{p=1}^{N_{\text{all}}} L_{rp}(\mathbf{x}, \vec{\omega}) \quad (4.4)$$

で求めることができる.

以上の改良によりフォトンマップに格納されたすべてのフォトン放射輝度の推定で使用するため,最近傍フォトンの検索処理を簡略化できる

4.2. 3DAPI を用いた放射輝度の推定

3DAPI の描画機能を使い放射輝度の推定を行う.放射輝度は式(4.4)より,各フォトンの放射輝度への影響 L_{rp} をフォトンマップに格納されたフォトン数だけ計算し,加算することで求める.

L_{rp} の計算は 3DAPI の三角形ポリゴン描画機能を使い,ピクセルシェーダ (図 2.9 参照) によって L_{rp} を計算する.はが L_{rp} が 0 以上となる範囲は,式(4.2)からフォトンの位置を中心とした半径 r の球内となる.したがって,本手法ではスクリーン上のこの球を覆うポリゴンを 3DAPI でレンダリングし,その過程で L_{rp} を計算することとした.この処理では図 4.1 に示すポリゴンを図 4.2 のように変換して作成する.

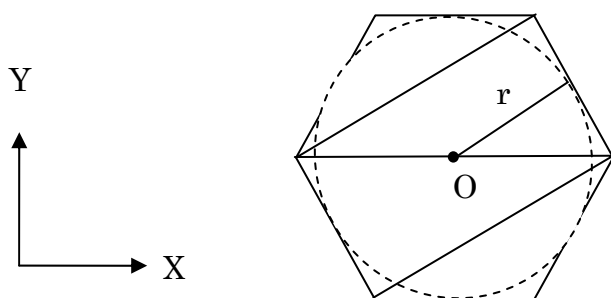


図 4.1 放射輝度推定に使用するポリゴン.
原点を中心とした正多角形で,半径 r の円を内接する.

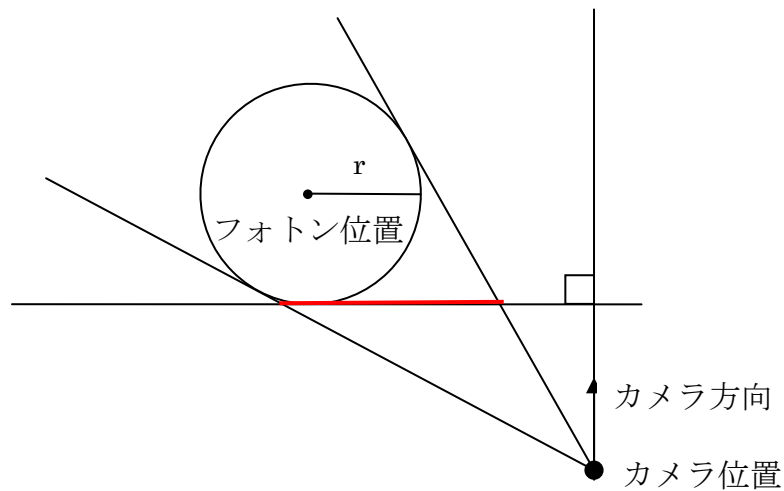


図 4.2 ポリゴンの変換.

ラスライザに入力されるポリゴンは赤い位置に変換されている。

L_{rp} 計算では図 2.9 の入力アセンブラに各フォトンのデータを頂点データとして入力する. 図 4.1 のポリゴンは, 定数バッファに頂点座標の配列で定義している, ジオメトリシェーダ(図 2.9)を用い, 図 4.2 に示す平面上(赤い部分)に図 4.1 のポリゴンを変換する. 変換されたポリゴンはラスライザでピクセルデータに変換されピクセルシェーダに送られる(図 2.9). ピクセルシェーダで計算する L_{rp} は, ラスライザで変換されたピクセルのスクリーン座標に表示されているモデル表面がフォトンから受ける影響である. モデル表面の位置 \mathbf{x} は, モデル表面の座標を記録したテクスチャから取得する. このほかにモデルの色と法線を記録したテクスチャを用いて

L_{rp} を計算する.

計算した L_{rp} は大域照明用のバッファに加算し, 以上の処理をフォトンマップに格納されたフォトンの数だけ行うことで大域照明による放射輝度を計算する.

4.3. 結果

4.2 で述べた手法によるレンダリング結果を図 4.3 に示す.

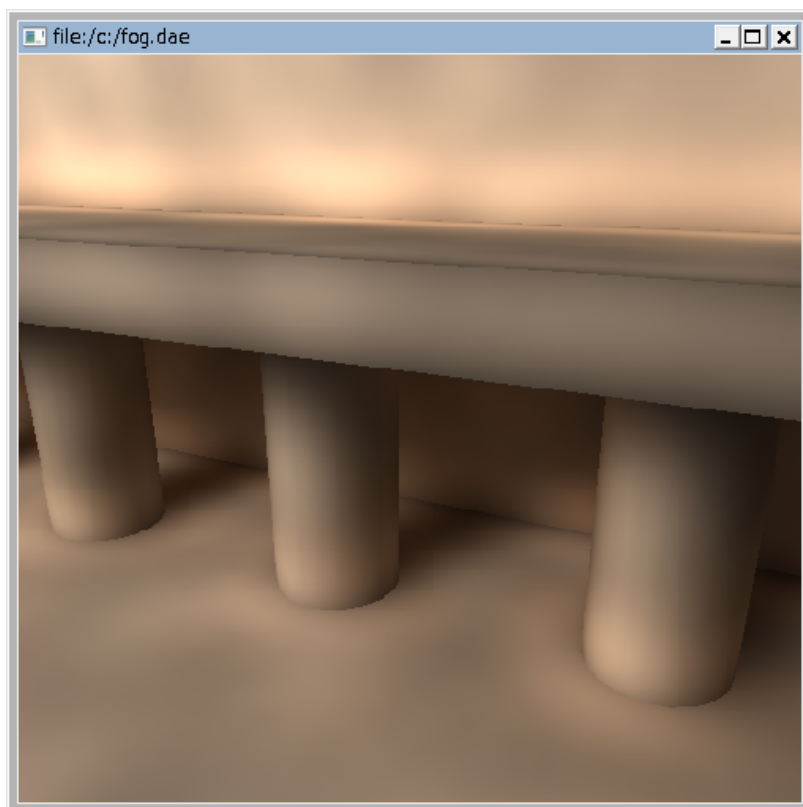


図 4.3 大域照明のレンダリング結果

4.4. 集光模様

フォトンマッピング法[1]では集光模様も扱えるため,4.2 節の手法を応用し,3DAPI を用いた集光模様のレンダリングを行った.

集光模様は,鏡面やガラスに反射した光が局所的に集中して模様になる現象である.フォトンマッピング法では図 4.4 に示すように,直接光が 1 回以上反射・屈折して拡散面に到達した際,集光模様をレンダリングするためのフォトンマップである,集光フォトンマップにフォトンを格納する.本件研究ではこの集光フォトンマップを用い 4.2 節で述べた手法によって集光模様をレンダリングする.

ガラス玉による屈折で描かれた集光模様のレンダリング結果を図 4.5 に示す.ここで,本手法による集光模様は,床部分のみで,ガラス玉による視線(レイ)の反射屈折はレイトレーシングを用いてレンダリングしている.

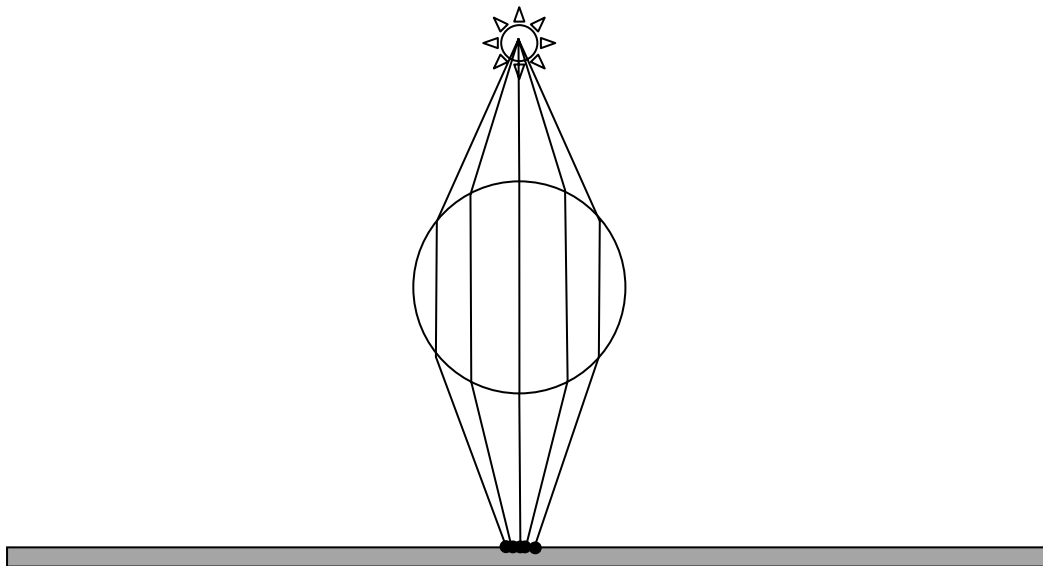


図 4.4 集光光子マップに格納される光子
反射屈折する前に,一度でも拡散面で反射された光子は格納されない.

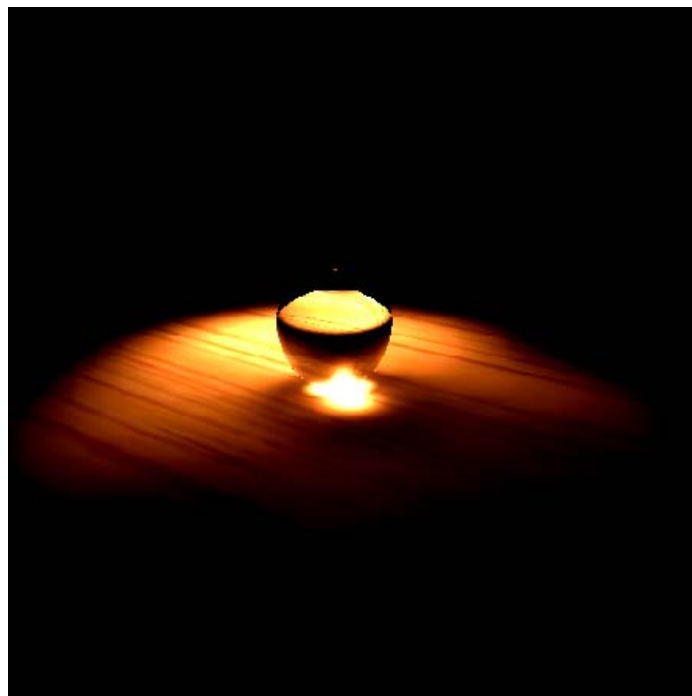


図 4.5 ガラス玉による集光模様.
ガラス玉部分のレンダリングにはレイトレーシングを用い,
床面に 3DAPI を用いてレンダリングしている.

第5章 関与媒質への応用

関与媒質とは空気や雲,濁った水,大理石,人肌などの半透明な存在であり,光を通すが,散乱や吸収といった形で光りに関与する.青空や夕焼け,朝焼けはこの作用による代表的な現象である.多くの場合,このような媒質を無視することが可能であるが,野外のシーンや,ほこりや微粒子の多い場面の表現には無視できない.フォトンマッピング法ではフォトンの関与媒質中での散乱,吸収を体積フォトンマップ(5.2 節で述べる)に記録することでレンダリングを可能にしている.本章では4章で述べた手法を応用した3DAPIによる関与媒質の描画手法について述べる.

5.1. 関与媒質中での光の散乱

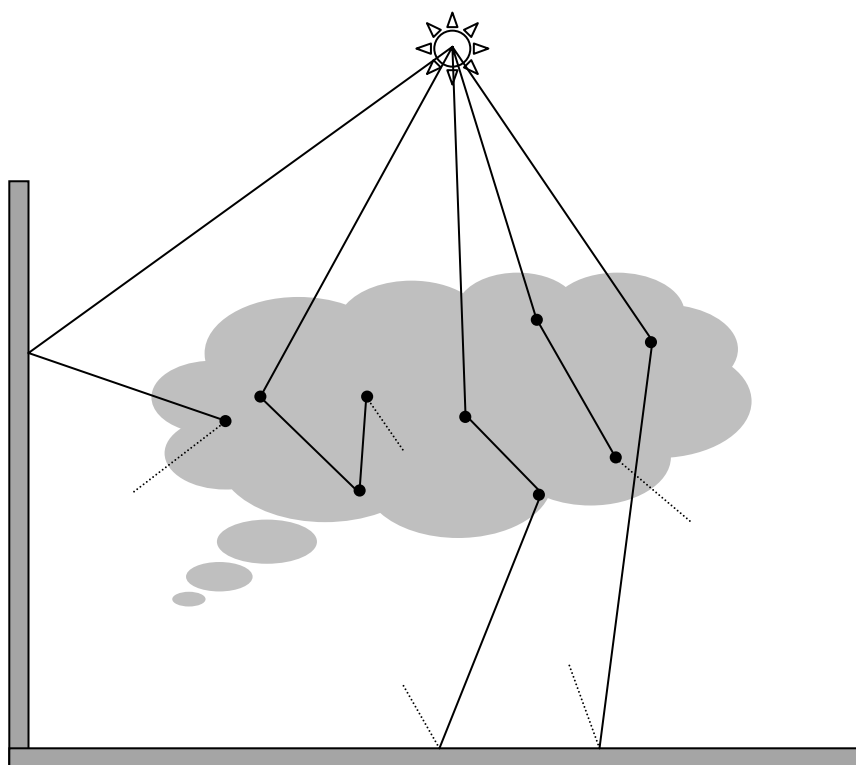


図 5.1 関与媒質中でのフォトンの散乱.
枠線のない灰色の部分に関与媒質である.
フォトンが関与媒質内で散乱もしくは吸収したとき
体積フォトンマップに蓄積される.

フォトンが関与媒質内に入射するとそのまま影響を受けず通過することもある

れば,ある場所で媒質と相互に作用することもある.フォトンと媒質の相互作用では吸収か散乱が起こり,その確率は散乱係数 σ_s ,吸収係数 σ_a によって与えられる.関与媒質中でのフォトンの散乱を図 5.1 に示す.

方向 \vec{w} における放射輝度 L の変化の中で,外への散乱による成分は式(5.1)で,吸収による成分は式(5.2)で与えられる.

$$(\vec{w} \cdot \nabla)L(x, \vec{w}) = -\sigma_s(x)L(x, \vec{w}) \quad (5.1)$$

$$(\vec{w} \cdot \nabla)L(x, \vec{w}) = -\sigma_a(x)L(x, \vec{w}) \quad (5.2)$$

式(5.1)および式(5.2)で与えられる散乱,吸収によって放射輝度 L の放射輝度が低くなる.両者を合わせ,放射輝度 L の放射輝度の損失は式(5.3)で与えられる.

$$(\vec{w} \cdot \nabla)L(x, \vec{w}) = -\sigma_t(x)L(x, \vec{w}) \quad (5.3)$$

ここで, $\sigma_t = \sigma_s + \sigma_a$ であり,消滅係数と呼ぶ.

媒質中をフォトンが進むにつれ,内部への散乱により放射輝度 L が増すこともある.内部への散乱による放射輝度の変化は式(5.4)で与えられる.

$$(\vec{w} \cdot \nabla)L(x, \vec{w}) = \sigma_s(x) \int_{\Omega_{4\pi}} p(x, \vec{w}', \vec{w}) L_i(x, \vec{w}') d\vec{w}' \quad (5.4)$$

ただし,入射輝度 L_i は球体 $\Omega_{4\pi}$ 上の全方向にわたり積分して求めている.媒質から放出される輝度の増加 L_e もあり得る.例えば炎のような媒質が該当する.この現象は,式(5.5)で与えられる.

$$(\vec{w} \cdot \nabla)L(x, \vec{w}) = \sigma_a(x)L_e(x, \vec{w}) \quad (5.5)$$

以上の式を組み合わせることで,単位距離当たりの放射輝度の変化は式(5.6)のようになる.

$$(\vec{w} \cdot \nabla)L(x, \vec{w}) = \sigma_a(x)L_e(x, \vec{w}) - \sigma_t(x)L(x, \vec{w}) + \sigma_s \int_{\Omega_{4\pi}} p(x, \vec{w}', \vec{w}) L_i(x, \vec{w}') d\vec{w}' \quad (5.6)$$

式(5.6)の両側を区間 s で積分し,別方向から媒質に入射する放射輝度の関与を付加することによって式(5.7)を得る.

$$\begin{aligned}
L(x, \vec{w}) = & \int_0^s e^{-\tau(x, x')} \sigma_a(x') L_e(x') dx' + \\
& \int_0^s e^{-\tau(x, x')} \sigma_s(x') p(x', \vec{w}', \vec{w}) L_i(x', \vec{w}') d\vec{w}' dx' \\
& + e^{-\tau(x, x+s\vec{w})} L(x + s\vec{w}, \vec{w})
\end{aligned} \tag{5.7}$$

ただし, 光学的深度 $\tau(x, x')$ は式(5.8)で与えられる.

$$\tau(x, x') = \int_x^{x'} \sigma_t(t) dt \tag{5.8}$$

式(5.7)がボリュームレンダリング方程式であり, この方程式を解くことで関与媒質を描画することができる.

5.2. 体積フォトンマップ

関与媒質のあるシーンの中でフォトンを追跡する間に, フォトンは媒質と相互作用し, 散乱もしくは吸収される. フォトンが関与媒質の中に入った場合, 媒質境界では散乱が起こらない代わりに媒質中で散乱か吸収が起きるまでフォトンは突き進む. この相互作用がの発生確率は, 消滅係数によって決まる. フォトンが次の相互作用まで媒質中を進む平均距離 d は, 式(5.9)で表せる.

$$d = \frac{1}{\sigma_t} \tag{5.9}$$

媒質の中を進む光線の強度は, $e^{-\sigma_t s}$ (s は媒質内を移動する距離) に従って減少する. 不均一な媒質の場合は, フォトンを追跡する過程で $\tau(0, d)$ の代わりに σ_t を用い, 次の相互作用までの距離 d を求め, 重点サンプリング法[1]を適用する. 距離 d は式(5.10)で求められる.

$$d = -\frac{\log \xi}{\sigma_t} \tag{5.10}$$

ここで $\xi \in [0, 1]$ は一様に分布した乱数である. 次の作用までの距離を式(5.10)で求めた場合, フォトンが媒質中を進むに従ってその出力を減らす必要はなくなる.

相互作用点においてフォトンは吸収か散乱のいずれかが起きる. 散乱率は散乱反射能 Λ で与えられる.

$$\Lambda = \frac{\sigma_s}{\sigma_t} \quad (5.11)$$

フォトンが媒質中で散乱されるか,吸収されるかはロシアンルーレットを用いる.すなわち,乱数 $\xi \in [0,1]$ と Λ を比較して,フォトンのふるまいを式(5.12)で決める.

$$\xi \in [0,1] \rightarrow \begin{cases} \xi \leq \Lambda & \text{フォトンが散乱される} \\ \xi > \Lambda & \text{フォトンが吸収される} \end{cases} \quad (5.12)$$

フォトンの散乱が起きても,フォトンは同じ出力を維持し続ける.散乱したフォトンの方向は,位相関数の重点サンプリング法によって計算する.位相関数には式(5.13)に示す Schlick の位相関数を用いる.

$$p(\theta) = \frac{1-k^2}{4\pi(1+k \cos\theta)^2} \quad (5.13)$$

ここで,新しい散乱方向を示す角度 θ は式(5.14)で与えられる.

$$\cos\theta = \frac{2\xi+k-1}{2k\xi-k+1} \quad (5.14)$$

ただし, $k \in [-1,1]$ は散乱方向を制御するために使われ ($k=0$ は等方散乱, $k>0$ は前方散乱, $k<0$ は後方散乱), ξ は0から1の間に一様分布した乱数である.回転角度 ϕ は一様に分布している.

関与媒質の場合は別途用意した体積フォトンマップに,フォトンが媒質と相互作用したとき格納される.Jensen らは,光線漸進法[1]が直接照明による寄与成分を計算できるため,少なくとも一度以上散乱したフォトンだけを格納して体積フォトンマップを構築した.本手法ではすべての散乱したフォトンに格納して体積フォトンマップを構築する.フォトンの関与媒質中での相互作用を体積フォトンマップに蓄積することで,体積フォトンマップから関与媒質中の放射輝度を求めることができる.

5.3. 体積放射輝度

媒質中のある点において外向きに散乱された放射輝度を,体積フォトンマップから推定する.外向きに散乱された放射輝度 L_o の計算には式(5.4)を用い,式(5.15)で行う.

$$(\vec{w} \cdot \nabla)L_o(x, \vec{w}) = \sigma_s(x) \int_{\Omega_{4\pi}} p(x, \vec{w}', \vec{w})L(x, \vec{w}')d\vec{w}' \quad (5.15)$$

格納された光子は入射する放射束を表しているため、この式を入射する放射束を積分する形に変形する.関与媒質内における放射束と放射輝度に関する式(5.16)の係数を用い,式(5.17)を得る.[2]

$$L(x, \vec{w}) = \frac{d^2\Phi(x, \vec{w})}{\sigma_s(x)d\vec{w}dV} \quad (5.16)$$

$$\begin{aligned} (\vec{w} \cdot \nabla)L_o(x, \vec{w}) &= \sigma_s(x) \int_{\Omega_{4\pi}} p(x, \vec{w}', \vec{w}) \frac{d^2\Phi(x, \vec{w}')}{\sigma_s(x)d\vec{w}'dV} d\vec{w}' \\ &= \int_{\Omega_{4\pi}} p(x, \vec{w}', \vec{w}) \frac{d^2\Phi(x, \vec{w}')}{dV} \end{aligned} \quad (5.17)$$

一方, Jensen らは, n 個の最近傍光子を検索し, 物体表面の放射輝度と同じ戦略で,最近傍光子を含む球(図 5.2 の灰色部分)の体積から,局所的な光子の密度を求め,外向きに放射される体積放射輝度を, 式(5.18)で求めている.

$$\begin{aligned} (\vec{w} \cdot \nabla)L_o(x, \vec{w}) &= \int_{\Omega_{4\pi}} p(x, \vec{w}', \vec{w}) \frac{d^2\Phi(x, \vec{w}')}{dV} \\ &\approx \sum_{p=1}^n f(x, \vec{w}'_p, \vec{w}) \frac{\Delta\Phi_p(x, \vec{w}'_p)}{(3/4)\pi r^2} \end{aligned} \quad (5.18)$$

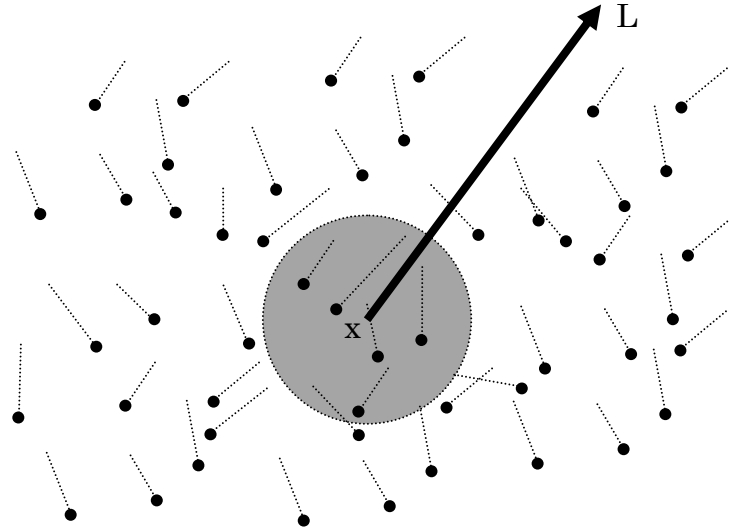


図 5.2 体積放射輝度の計算.

最近傍フォトン を体積フォトンマップから検索し,球の体積に基づいた
フォトンの密度から体積放射輝度を計算する.

5.4. 3DAPI による体積放射輝度の推定

4.2 節と同様に,体積フォトンマップに格納されたフォトン をすべて使うこと
で, 式(5.18)の計算に必要な最近傍フォトンの検索処理を簡略化する.

$$(\vec{w} \cdot \nabla)L_o(x, \vec{w}) \approx \sum_{p=1}^{N_{\text{all}}} f(x, \vec{w}'_p, \vec{w}) \frac{\Delta\Phi_p(x, \vec{w}'_p)}{(3/4)\pi r^2} w_p(d_p) \quad (5.19)$$

$$w_p(d_p) = \begin{cases} 1 & (d_p \leq r \text{ の場合}) \\ 0 & (d_p > r \text{ の場合}) \end{cases} \quad (5.20)$$

ここで, N_{all} は体積フォトンマップに格納されているフォトン数, d_p は放射輝度
の計算位置とフォトンの位置の距離, r はフォトンの効果範囲を制御する定数で
ある.

5.5. 媒質の描画

本研究では, 媒質の在り方を, 均一なもの と不均一なもの の 2 つの場合に分
けている.均一に媒質が存在する場合では,いかなる場所でも散乱係数,吸収係数,

消滅係数は一定である.不均一な媒質では,領域ごとに散乱・吸収・消滅の各係数が異なる媒質が不均一に存在する.

5.6. 均一な媒質のレンダリング

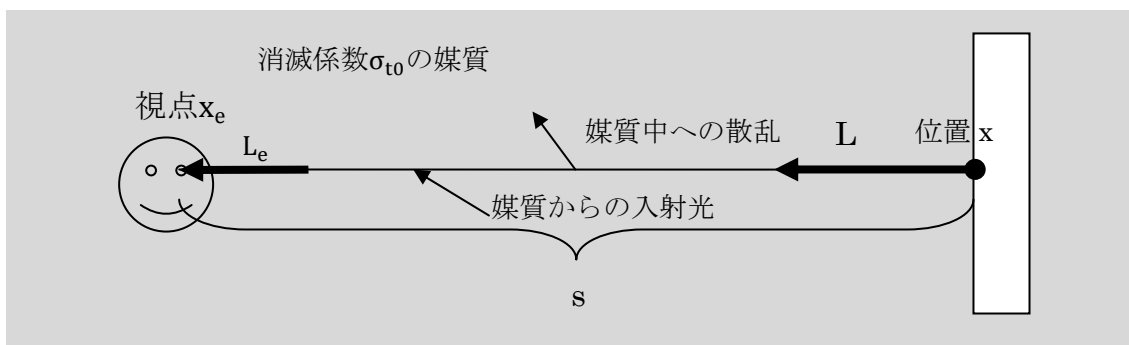


図 5.3 均一な媒質中の光.

均一な媒質中では消滅係数 σ_{t0} は一定である.

視線上の光は減衰するが,外部からの入射光の影響も受ける.

均一な媒質が存在する場合,物体表面の放射輝度は減衰してカメラや目に届く.図 5.3 は位置 x の散乱面からの放射輝度 L が均一な媒質(灰色部分)によって散乱され,また媒質から放射輝度を入射しながら視点へ届くことを表している.. 図 5.3 の場合,視点に届く輝度 L_e は式(5.21) で求められる.

$$L_e(x_e, \vec{w}) = e^{(-\sigma_{t0}s)}L(x, \vec{w}) + L_{in}(x_e, \vec{w}) \quad (5.21)$$

ここで L_{in} は媒質からの入射による影響である.媒質中の位置 x における体積放射輝度は,フォトンマップから式(5.19)を用いて求められが,物体表面の放射輝度同様,媒質中で減衰される.

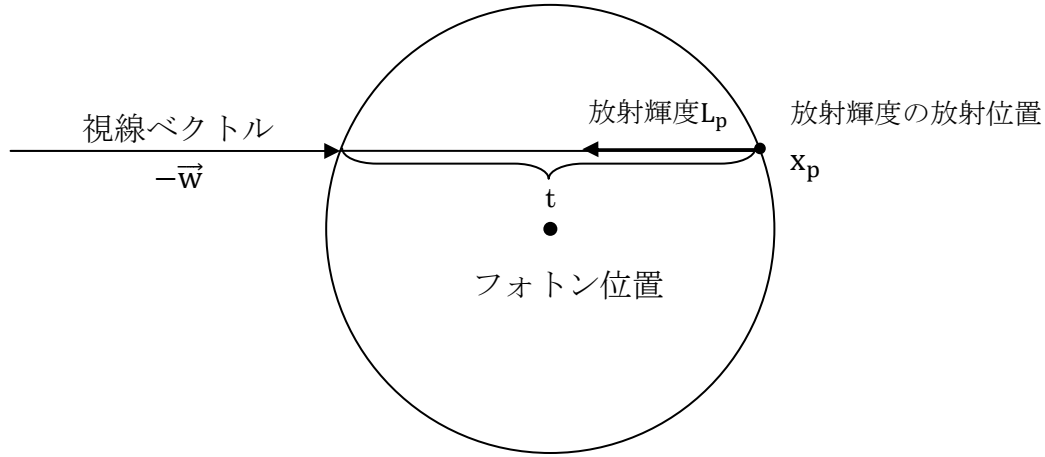


図 5.4 各フォトンからの放射輝度

各フォトンが媒質中の位置 x の体積放射輝度へ与える影響は式(5.19)から式(5.22)で与えられる.

$$(\vec{w} \cdot \nabla)L_p(x, \vec{w}) \approx f(x, \vec{w}'_p, \vec{w}) \frac{\Delta\Phi_p(x, \vec{w}'_p)}{(3/4)\pi r^2} w_p(d_p) \quad (5.22)$$

ただし,媒質のある場合は視線上のすべての位置で体積放射輝度を求め,積分する必要がある.そこで,本研究では,各フォトンの視線方向への放射輝度は図 5.4 中の位置 x_p から放射されていると仮定する.このとき,位置 x_p からの放射輝度 L_p を式(5.23) で求める.

$$L_p(x_p, \vec{w}) = f(x, \vec{w}'_p, \vec{w}) \frac{\Delta\Phi_p(x, \vec{w}'_p)}{(3/4)\pi r^2} t \quad (5.23)$$

放射輝度は視点に届くまで減衰するため,視点に届く輝度 L_e は

$$L_e(x_e, \vec{w}) = e^{(-\sigma_{t0}s)} L(x, \vec{w}) + \sum_{p=1}^{N_{\text{all}}} e^{(-\sigma_{t0}s_p)} L_p(x_p, \vec{w}) \quad (5.24)$$

で求められる.ここで, s_p は視点 x_e から各フォトンの x_p までの距離である.

描画には 3DAPI を使い,4.2 節と同じ方法でポリゴンを作成し,ピクセルシェーダで式(5.23)を計算し,体積放射輝度用のバッファに蓄積することで体積放射輝度を推定する.均一な媒質の場合のレンダリング結果を図 5.5 に示す.

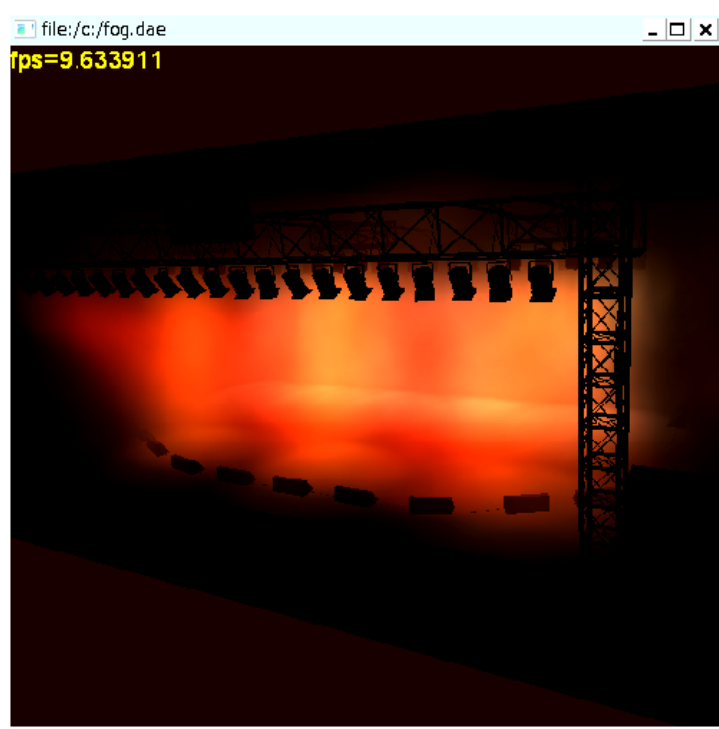


図 5.5 均一な媒質シーンのレンダリング結果
ステージを照らすスポットライトの光が,媒質中で散乱する光が表れている.

5.7. 不均一な媒質のレンダリング

本研究では散乱・吸収・消滅の各係数が一定の領域を複数設定することで不均一な媒質シーンのレンダリングを行う(図 5.6 参照).ここでは消滅係数 σ_{tA} の領域 A(赤い領域)と消滅係数 σ_{tB} の領域 B(青い領域)が存在し,領域 A と領域 B が重なる部分(緑の領域)では散乱係数が $\sigma_{tA} + \sigma_{tB}$ になる.領域 A,領域 B,領域 A,B が混在する混合領域を視線が通過する距離はそれぞれ d_1, d_3, d_2 する.視線上の領域 A の境界面から位置 x までの距離を s_1, s_3 ,領域 B の境界面から位置 x までの距離を s_2, s_4 とする.

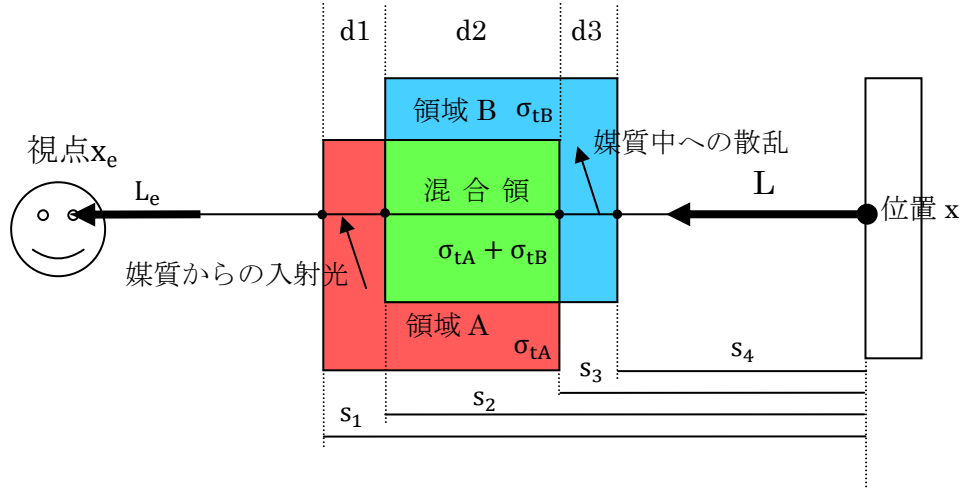


図 5.6 領域設定による不均一な媒質.

均一な場合と同様, 視線上の光は減衰するが, 外部からの入射光の影響も受ける.

図 5.6 で, L は領域 A および領域 B 中を通過し L' に変化する (L' には, 媒質から入射する放射輝度は含まれない. 媒質から入射した放射輝度は別に求める).

$$\begin{aligned} L' &= e^{-\sigma_{tA}d1} e^{-(\sigma_{tA}+\sigma_{tB})d2} e^{-\sigma_{tB}d3} L \\ &= e^{-(\sigma_{tA}d1+(\sigma_{tA}+\sigma_{tB})d2+\sigma_{tB}d3)} L \end{aligned} \quad (5.25)$$

式(5.25)を領域の境界面から位置 X までの距離で表すと

$$\begin{aligned} L' &= e^{-(\sigma_{tA}(s1-s2)+(\sigma_{tA}+\sigma_{tB})(s2-s3)+\sigma_{tB}(s3-s4))} L \\ &= e^{-(\sigma_{tA}s1-\sigma_{tA}s3+\sigma_{tB}s2-\sigma_{tB}s4)} L \end{aligned} \quad (5.26)$$

となる. 式(5.26)から, 放射輝度 L' は, 媒質境界面の x までの視線上の距離と放射輝度が, 境界面から領域へ入射か出射かによって計算できる. 式(5.26)を一般化すると式(5.27)を得る.

$$L' = e^{-(\sigma(x))L} \quad (5.27)$$

$$\sigma(x) = \sum_{b=1}^B \sigma_{tb} s_b O(b) \quad (5.28)$$

ここで, B は視点から位置 x の間にある媒質の境界面の数, σ_{tb} は境界面の領域における消滅係数, s_b は視線上の境界面から位置 x までの距離である. また $O(b)$ は, 光が媒質に入る境界面では $O(b) = -1$, 媒質から出る境界面では $O(b) = 1$ である.

視点が受ける輝度は式(5.29)で与えられる.

$$L_e(x_e, \vec{w}) = e^{-(\sigma(x))}L(x, \vec{w}) + L_{in}(x_e, \vec{w}) \quad (5.29)$$

不均一な媒質中の L_{in} をフォトンマップから求めるために、視線を等間隔に分割し、各フォトンの放射輝度が図 5.7 に示す x_f から放射されていると仮定して計算する。

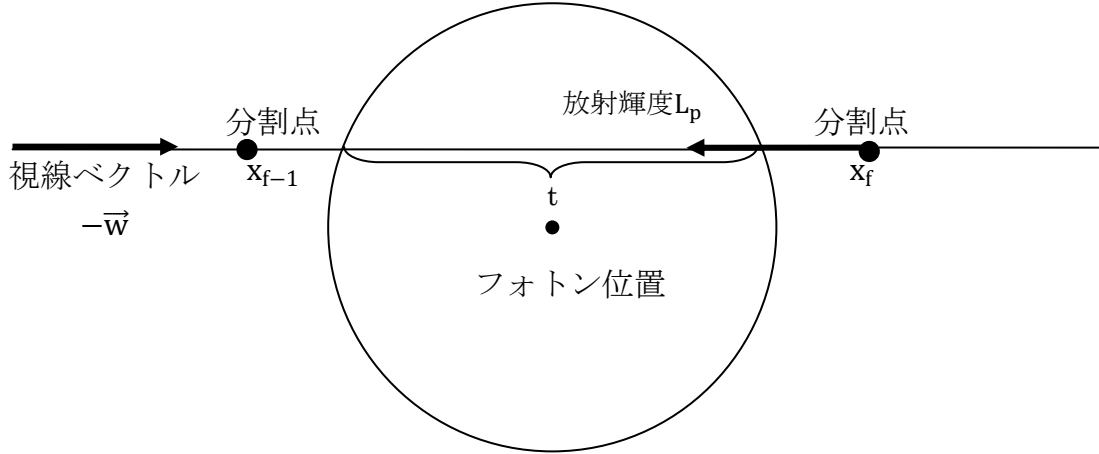


図 5.7 視線を分割した場合の各フォトンからの放射輝度
不均一な媒質の場合,放射輝度 L_p は分割点 x_f から放射されると仮定する。

放射輝度 L_p は式(5.23)で求め,図 5.7 が示すような分割点 x_f から放射すると,視点に届く輝度は

$$L_e(x_e, \vec{w}) = e^{-(\sigma(x))}L(x, \vec{w}) + \sum_{p=1}^{N_{all}} e^{-(\sigma(x_f))}L_p(x_p, \vec{w}) \quad (5.30)$$

で求められる。

均一な場合と同様に 3DAPI を用いてレンダリングを行った。3DAPI を用いるにあたり,媒質領域の境界をポリゴンメッシュで定義している。ポリゴンを使うため式(5.28)の $O(b)$ は,視点から見てポリゴンが表なら $O(b)=1$,裏なら $O(b)=-1$ と変更できる。不均一な場合,視線を分割するため図 5.8 のようなプロセスでレンダリングを行う。 x_s は分割点を表し, S は視線の分割数である。不均一な媒質の場合のレンダリング結果を図 5.9 に示す。均一な場合のレンダリング結果(図(5.5))とくらべ,媒質が存在するステージの床付近が媒質によって強く光っているのが分かる。

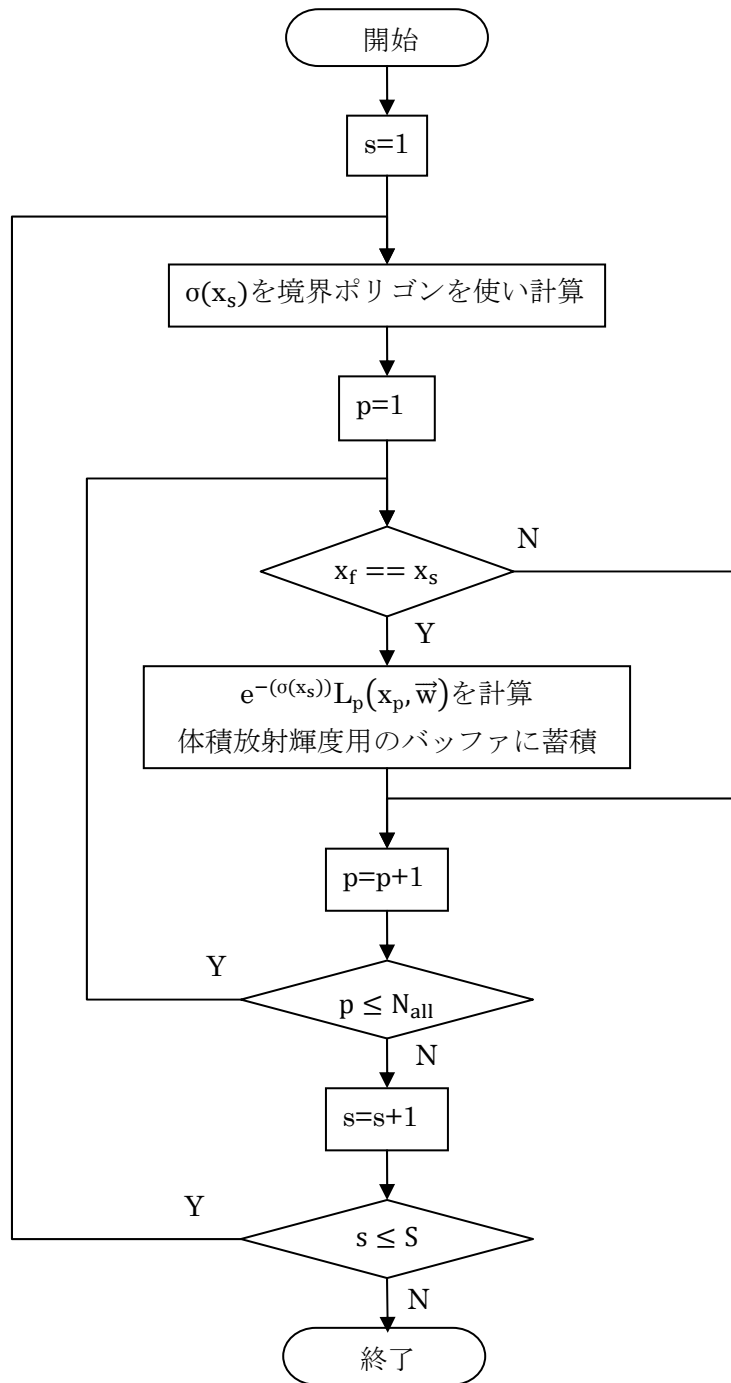


図 5.8 不均一時の $\sum_{p=1}^{N_{all}} e^{-\sigma(x_f)} L_p(x_p, \vec{w})$ 計算プロセス

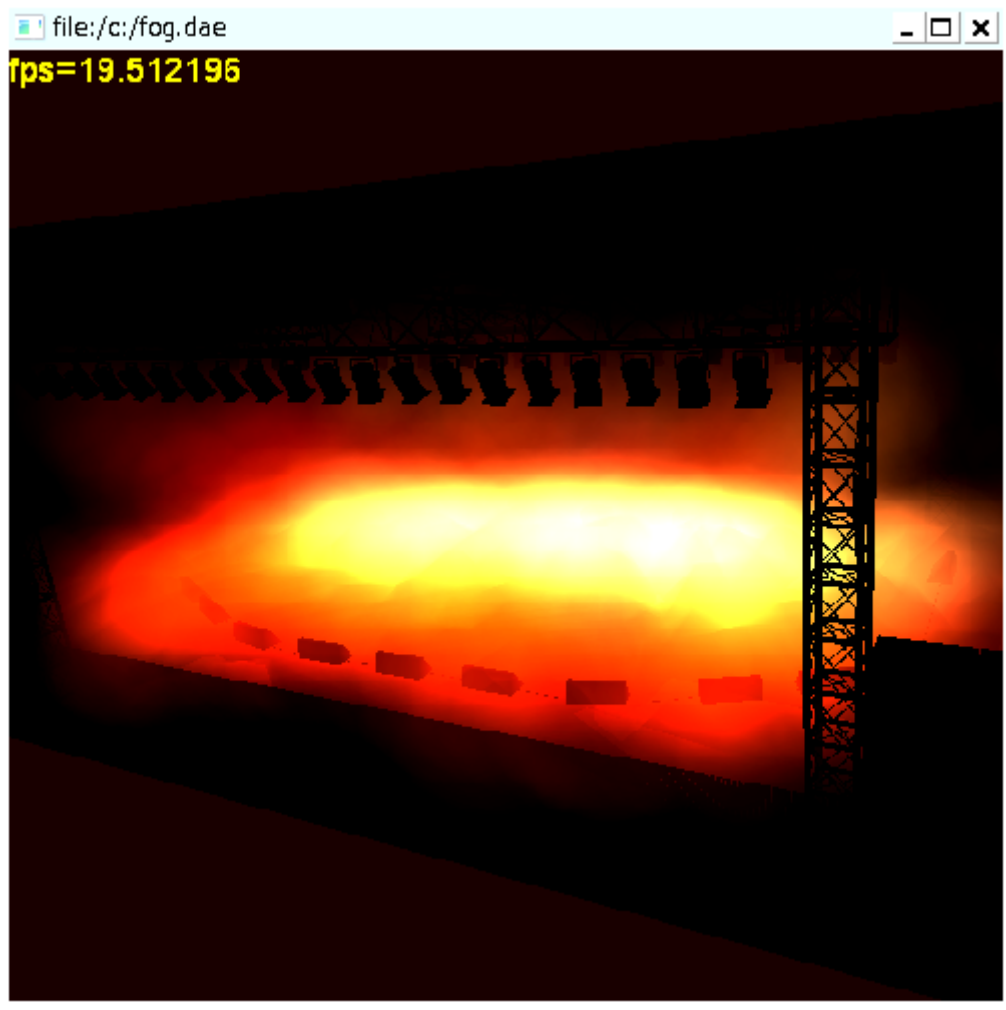


図 5.9 不均一な媒質シーンのレンダリング結果
ステージの床付近に媒質を配置している.
光が床付近の媒質で散乱し,輝く様子を表せている.

第6章 最適化

6.1. レベルオブディテール

レベルオブディテール(LOD)[13]とは,レンダリングするモデルをどれだけ詳細にレンダリングするかを,視点とモデルの距離に応じてモデルの詳細度を変える手法である.一般的には視点に近い位置にモデルがある場合は詳細に,離れている場合は簡略化してレンダリングする手法である.

本研究は LOD の概念をフォトンマッピング法に適用する. すなわち,放射輝度計算に使用するフォトン,詳細にレンダリングする箇所では多く,簡略化する場所では少なくすることで,より高速かつ効果的なレンダリングを目指す.以降,物体表面の放射輝度を計算する場合についてのみ述べるが,媒質における体積放射輝度の計算時も同様である.

LOD を実現するために,フォトンのグループ分けを以下のように行った. フォトンマップに格納される順番にフォトン,グループに振り分ける. グループ数は A とし,各グループに番号 C_p を付け, $0, 1, \dots, A - 1$ とする. 詳細度は Q とし,視点からフォトン位置までの距離 d から式(6.1)で求める.

$$Q = \text{smoothstep}(n, f, d) \quad (6.1)$$

ここで smoothstep は $n \leq f$ の場合は, $d < n$ ならば 0 , $d > f$ ならば 1 を, d が $[n, f]$ の範囲内であれば 0 と 1 の間での滑らかなエルミート補完の値を返し, $n > f$ の場合は $1 - \text{smoothstep}(f, n, d)$ を返す関数である.

本研究では詳細度 Q とグループを使い,それぞれのフォトンの出力の調整を行い放射輝度計算に使用するフォトンの数を削減する. 1 グループに属するフォトンのフォトンマップ全体の割合 dQ を式(6.2)で,各グループへの帰属判定用の値 G_p を式(6.3)で与える.

$$dQ = 1/A \quad (6.2)$$

$$G_p = C_p/A \quad (6.3)$$

各フォトン p の出力の調整は式(6.4)で行われる.ここで Φ_p はフォトンの出力, Q_p はフォトン位置における詳細度である.

$$\Phi_p^* = \begin{cases} \frac{1}{1-Q_p} & (Q_p \leq G_p \text{ のとき})(a) \\ \frac{1}{1-G_p} - \frac{\left(\frac{1}{1-Q_p} - \frac{1}{1-G_p}\right)(1-(G_p+dQ))}{dQ} & (G_p + dQ \leq G_p < Q_p \text{ のとき})(b) \\ 0 & (G_p < G_p + dQ \text{ のとき})(c) \end{cases} \quad (6.4)$$

式(6.4) の(a)はフォトン p の位置において有効なフォトンの総数が $1-Q$ 倍になったと仮定したとき,有効なフォトン全体の総出力を一定に保つための式である.

式(6.4) の(b)は,(a)と(c)の中間の状態,有効なフォトン全体の総出力を保ちながら出力を落とし,(c)へ移る.式(6.4)の1項目である $\frac{1}{1-G_p}$ は, (a)から(b)に移行

したときの出力である. $\frac{\left(\frac{1}{1-Q_p} - \frac{1}{1-G_p}\right)(1-(G_p+dQ))}{dQ}$ は,(a)状態である全フォトン

の, $\frac{1}{1-Q_p}$ からの増加量の合計を,(b)状態フォトンに分配した量である.(b)状態フォトンの出力を式(6.4)の(b)で求めることで,(b)状態のフォトンの出力を 0 にする.そして, Q の値が上がると最終的にはフォトン(c)状態になり,(c)状態のフォトンの L_{rp} (式(4.3))計算を簡略化する.

本研究ではフォトンマップに格納された各フォトンに式(6.4)をあてはめ,リアルタイムレンダリングを目指した.

コーネルボックスのシーンに式(6.4)を適用した結果を図 6.1 に示す.計測単位は 1 秒間の描画回数を表す **fps(Frame Per Second)**を用いている.この例では **LOD** を使うことで **fps** が約 2 倍になった.レンダリング結果はほぼ変わらないが,**LOD** が視点に近い部分のフォトンが無効化する設定なため若干暗くなっている.本手法は各フォトンの効果をポリゴンを用いて適用するため,視点に近いフォトンはスクリーン上の効果範囲が大きいいため,速度面で視点に近いフォトン削減する効果は,奥のフォトン削減するよりも大きい.フォトンマップの精度を落とし,2000 個のフォトンを使用してレンダリングした結果を図 6.2 に示す.精度を落とし,適切な設定を行うことで大域照明をリアルタイムにレンダリングすることが出来た.

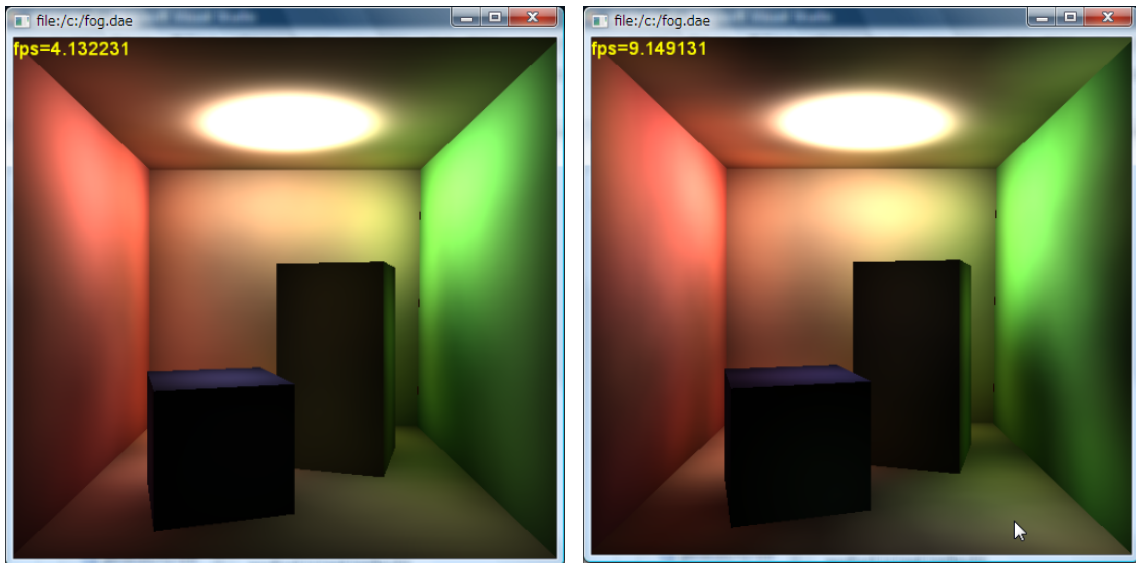


図 6. 1 LOD の有無による描画速度の比較.

10000 個のフォトンを使用.

モデルは視点から 9.6~19.6 の距離に配置されている.解像度 512x512.

(左)LOD 無効,(右)LOD 有効, $n=20$, $f=7$.約 2 倍のレンダリング速度になった.

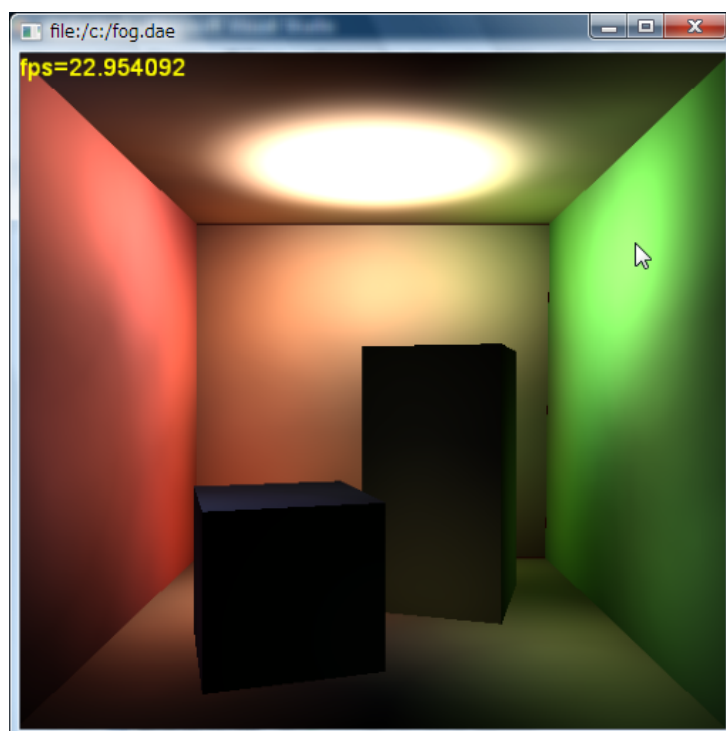


図 6.2 リアルタイムレンダリング出来るまで精度を落とした結果
図 6.1 と同様のシーンで 2000 個のフォトンを使用. 解像度 512x512.
LOD 設定 $n=20, f=7$ でレンダリングを行っている。

関与媒質に関してもステージのモデルを用いてレンダリングを行った.結果を図 6.3 に示す.この例は均一な場合のレンダリングである.媒質でも同様に LOD を用いることで,レンダリング速度が向上した.次に体積フォトンマップの精度を落とし,25000 個のフォトンを用いたレンダリング結果を図 6.4 に示す.この結果では遠い位置のフォトンが無効化している.媒質がある場合,距離が離れるほど光は届かなくなるため,媒質シーンでは遠い位置を簡略化する LOD の設定が有効な場合が多い.この結果でもリアルタイムに媒質が存在する場合の大域照明をレンダリングできた.

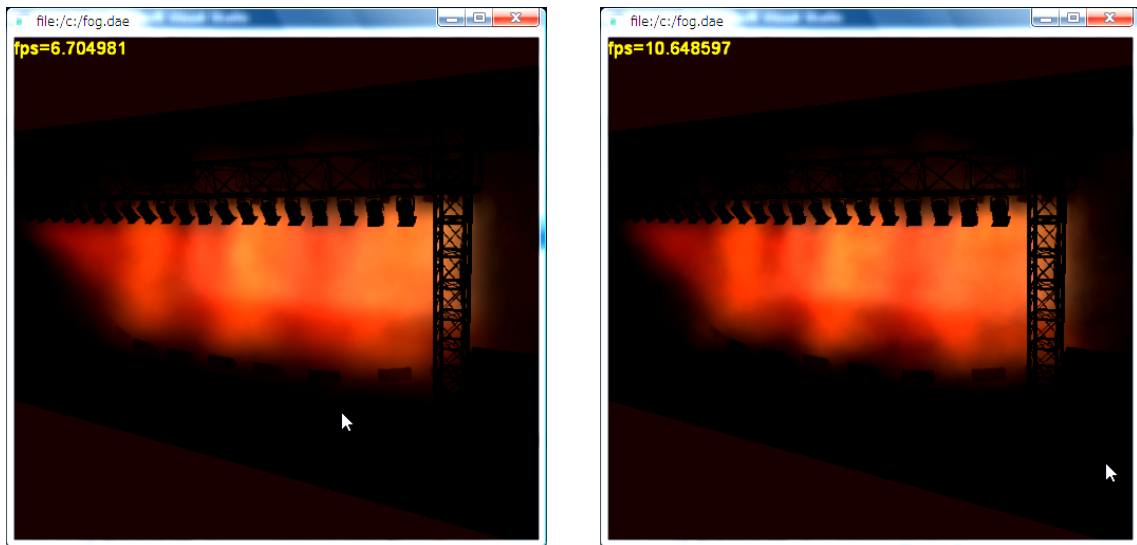


図 6.3 均一な媒質中のステージのレンダリング結果.フォトン 50000 個使用
 モデルは始点から 4.9~22.3 の距離は位置されている. 解像度 512x512.
 (左)LOD 無効, 6.7fps.(右)LOD 有効, $n=25, f=3$, 10.6fps.
 約 7 割程レンダリング速度が上がった.

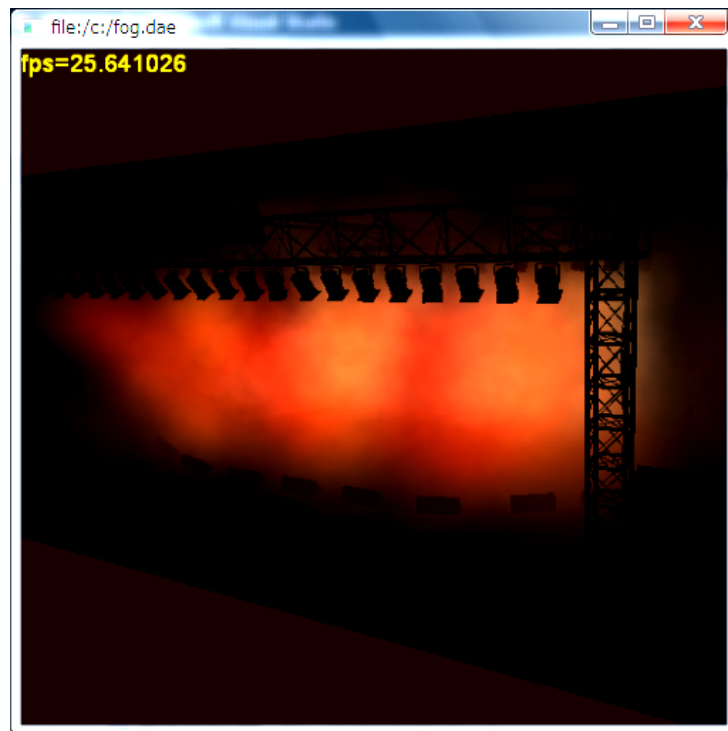


図 6.4 リアルタイムレンダリング出来るまで精度を落とした媒質シーンの結果
 図 6.3 と同様のシーンで 25000 個のフォトンを使用. 解像度 512x512.
 LOD 設定 $n=0, f=20$ でレンダリングを行っている。
 $N < f$ なので,視点から遠い位置にあるフォトンが無効化される.

第7章 まとめと課題

7.1. 本論文のまとめ

本論文では,3DAPI による大域照明を,フォトンマッピング法を用い,レンダリングする手法について述べた.

2 章では大域照明の概要と,多くの大域照明手法が基としているレイトレーシングについて述べた.また本研究で用いる 3DAPI および GPU の特徴と,3DAPI を用いて,大域照明をレンダリングする際の問題点を述べた.

3 章では,本研究で基となるフォトンマッピング法について述べた.フォトンマッピング法では,物体表面の放射輝度をフォトンマップから計算することで大域照明を 2 次レイを発生させずに計算することが可能である.この 2 次レイが発生しないという点が 3DAPI を用いて大域照明をレンダリングするうえで重要である.

4 章ではフォトンマッピング法を 3DAPI によってレンダリングする大域照明手法を提案した.放射輝度の計算時,フォトンマッピング法では,フォトンマップから利用する最近傍フォトンを検索によって見つけ出す処理が必要であった.しかし,レンダリング処理で利用される GPU は,分岐処理に不向きなハードウェアなため,検索処理はパフォーマンスの低下に繋がる.そこで,本研究では放射輝度計算時の検索処理を簡略化した放射輝度の計算手法を提案した.レンダリングには,3DAPI を用いてフォトンマップから放射輝度を計算する手法を提案し,レンダリング結果を示した.また,提案手法を応用することで集光模様のレンダリングを行った.集光模様はガラスや金属の表面で起きる,反射・屈折によって局所的に光が集まり現れる模様で,本研究では集光模様に関係する,反射屈折したフォトンに格納した集光フォトンマップを用いた.集光フォトンマップを用いることで,放射輝度と同様の手法によって集光模様のレンダリングを行い,レンダリング結果を示した.

5 章では 4 章で述べた手法を応用し,光を散乱,吸収する媒質のレンダリングについて述べた.まず,媒質が存在する場合のフォトン追跡法について述べ,媒質中のフォトンに格納する体積フォトンマップについて述べた.また,体積フォトンマップから媒質中で放射される体積放射輝度の算出方法についても述べた,次に媒質が在り方を,均一と不均一の場合にわけ,媒質の放射輝度を 3DAPI を用いて体積フォトンマップから計算する方法を提案した.

6 章では 4 章,5 章で述べた手法をレベルオブディテイルによって,レンダリン

グに用いられるフォトンの数を,レンダリング結果への影響を抑えながら高速化する手法について述べ,リアルタイムに近い,もしくはリアルタイムに準じる描画速度で大域照明をレンダリングすることを可能にした.

7.2. 今後の課題

本研究は 6 章で述べた手法によって,リアルタイムに近いレンダリング速度で大域照明をレンダリングすることが出来るようになった.しかし,映像メディアの fps と比較すると,映画フィルムは 24fps, NTSC 規格(アナログテレビ放送)では 29.97fps,ゲーム機などは 60fps であり [12],6 章の結果では映画フィルムには迫ったものの,その他のメディアには遠く及ばない.また解像度が低いため,高解像度での高速なレンダリングにも対応する必要があるため,今後はより効率的に大域照明をレンダリング出来る手法の開発を行いたい.

また,本研究では,ユーザーが操作できるのは視点移動のみに限られている.ゲームに代表されるインタラクティブコンテンツでは,視点はもちろん,キャラクター等の動くオブジェクトが存在し,光源も移動する.本研究で述べた手法では,こうしたインタラクティブコンテンツで利用する場合,視点以外を動かさないという制約になってしまい,多くの場合導入できない.今後は,視点だけでなく,物体や光源が動く動的シーンへ応用し,多くのインタラクティブコンテンツに導入できる手法にしていきたい.

本論文に関する研究発表

1. 柿原利政, 宮田一乗. "フォトンマップを利用したリアルタイム大域照明システム" NICOGRAPH 2007 秋. 2007
2. 柿原利政, 宮田一乗. "フォトンマッピング法を用いた 3DAPI による大域照明" NICOGRAPH 2008 春. 2008.

参考文献

- [1] Henrik Wann Jensen, 苗村 健, “フォトンマッピング—実写に迫るコンピュータグラフィックス”, オーム社, 2002.
- [2] Henrik Wann Jensen and Niels J. Christensen. “Efficiently rendering shadows using the photon map.” In *Compugraphics '95*, edited by Harold P. Santo, pp. 285-291, December 1995.
- [3] 後藤弘茂, 後藤弘茂の Weekly 海外ニュース,
<http://pc.watch.impress.co.jp/docs/2007/0326/kaigai346.htm>
- [4] .NRTTKR, “DirectX10 3Dプログラミング”, 工学社, 2007.
- [5] American National Standard Institute, “Nomenclature and Definitions for Illumination Engineering.”, ANSI report, ANSI/IESRP-16-1986, 1986.
- [6] F. E. Nicodemus, J. C. Richmond, J. J. Hsia, I. W. Ginsberg, and T. Limperis. “Geometric considerations and nomenclature for reflectance.” Monograph 161, National Bureau of Standards (US), October 1977.
- [7] Paul S. Heckbert. “Adaptive radiosity textures for bidirectional ray tracing.” *Computer Graphics(Proc. SIGGRAPH '90)*24(4): 145-154(August 1990).
- [8] Turner Whitted. “An improved illumination model for shaded display.” *Communications of the ACM* 23(6): 343-349 (June 1980).
- [9] James T. Kajiya. “The rendering equation.” *Computer Graphics(Proc. SIGGRAPH '86)* 20(4): 143-150 (August 1986).
- [10] Peter-Pike Sloan, Jan Kautz, and John Snyder. “Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments.” In *Proceedings of SIGGRAPH 2002*, pages 527-536, 2002.

- [11] Peter Shirley. “Discrepancy as a quality measure for sample distributions.” In Eurographics ’91, edited by Werner Purgathofer, pp.183-194, Amsterdam: North-Holland, September 1991.
- [12] Wikipedia, “高精細度テレビジョン放送”, <http://ja.wikipedia.org/wiki/>.
- [13] Pirre-Marie Gandoin, Olivier Devillers. “Progressive Lossless Compression Of Arbitrary Simplicial Complexes.” SIGGRAPH ’02, July 2002.
- [14] Holly E. Rushmeier, Kenneth E. Torrance. “The zonal method for calculating light intensities in the presence of a participating medium” SIGGRAPH ’87, August 1987.
- [15] Kyle Hegeman, Michael Ashikhmin, Shimon Premoze. “A lighting model for general participating media.” I3D ’05, April 2005.
- [16] Naty Hoffman, Arcot J Preetham. “Rendering Outdoor Light Scattering in Real Time.” GDC 2002, 2002.
- [17] Manfred Ernst, Tomas Akenine-Möller, Henrik Wann Jensen. “Interactive rendering of caustics using interpolated warped volumes.” Proceedings of Graphics Interface 2005, 2005.
- [18] Srinivasa G. Narasimhan, Mohit Gupta, Craig Donner, Ravi Ramamoorthi, Shree K. Nayar, Henrik Wann Jensen. “Acquiring scattering properties of participating media by dilution.” ACM Transactions on Graphics, July 2006.
- [19] Tomoyuki Nishita, Takao Sirai, Katsumi Tadamura, Eihachiro Nakamae. “Display of the earth taking into account atmospheric scattering.” SIGGRAPH ’93, September 1993.

謝辞

本研究を進める上で、的確なアドバイスとプレッシャーを与えていただいた宮田先生にこの場を借りて感謝の意を表したいと思います。宮田先生には今年のIVRCでの作品制作やその後の展示や投稿などでも大変お世話になりました。IVRCの一連の活動の中で経験したことは、これからの人生で大きな財産になると思います。有り難うございました。

IVRCでは様々な経験が出来ましたが、あれは、あのメンバーだからこそ出来たものだったと思います。共に戦った大内君、櫻井君、瀬井君、瀬崎君、谷本君に、この場を借りて感謝の意を表します。有り難うございました。

本研究では、インターン中の経験が少なからず活かされています。インターン中お世話になった米谷さん、原さん、志潟さん、古谷さんにこの場を借りて感謝の意を表します。また、インターンという貴重な機会を与えてくださった宮田先生に感謝します。

最後に、進学を急に決めた私を進学させてくれた両親に感謝します。