非均質関与媒質のレンダリングのためのアンバイアスで効率的な重点的サンプリング手法
An Unbiased and Efficient Importance Sampling Technique for Rendering Inhomogeneous Participating Media

楽 詠灝† 岩崎 慶‡ 陳 炳宇†† 土橋 宜典‡‡ 西田 友是†
Yonghao YUE† Kei IWASAKI‡ Bing-Yu CHEN†† Yoshinori DOBASHI‡‡ and Tomoyuki NISHITA†

†東京大学 †The University of Tokyo ‡和歌山大学 ‡Wakayama University
††國立臺灣大學 ††National Taiwan University ‡‡北海道大学 ‡‡Hokkaido University

E-mail: †{yonghao, nis}@nis-lab.is.s.u-tokyo.ac.jp, ‡iwasaki@sys.wakayama-u.ac.jp
††robin@ntu.edu.tw, ‡‡doba@ime.ist.hokudai.ac.jp

1. はじめに

煙，炎，空（雲と大気）は日常的に見られる非均質な関与媒質である．関与媒質の写実的なレンダリングでは光の散乱計算が重要であり，その効率的な計算法は CG 分野における重要な研究テーマの一つである．写実的なレンダリングにおいて，モンテカルロ積分に基づく方法は広く用いられており，これまで様々な方法[1-4]が開発されてきた．これらの手法では，各ピクセルの輝度を計算するために大量の光路を生成し，それらの寄与を積分する．非均質な関与媒質においては，こうした光路を生成する計算コストが高い．

本稿では，光路を効率よく生成する方法に焦点を当てる．光路の生成において，統計的に偏りがないことは重要であり，次のような利点がある：1) 解は厳密解に収束する，2) 偏りのある計算法では計算誤差の推定は難しいが，偏りのない手法では分散を計算することによって容易に推定できる[5]，3) 偏りに起因するアーティファクトを気にする必要がない．

関与媒質で光路を生成するには，ランダムに一連の散乱事象を生成する．新たな散乱事象の位置を決定するには，直前の散乱事象からの距離である自由行程(free path)を，乱数にしたがって決定する必要がある．自由行程を効率よく定めるため，しばしば free path sampling と呼ばれる重点的サンプリング法が適用される．Free path sampling では，光学的距離に関連する確率密度関数が与えられ，その分布に従うように自由行程を定める[3]．しばしばレイマーチングがこの目的のために利用される[3]が，得られる解には偏りがある．統計的に不偏な解を得る手法として，原子力科学の分野では Woodcock tracking[7] が提案されており，Raab ら[4]によって初めて CG 分野に導入された．自由行程は，散乱事象が起こったかどうかを判断しながら，微小な距離を確率的にインクリメントしながらサンプリングされる．これらの微小距離は，媒質の最も密な領域をサンプリングできるように十分小さくとる．媒質が非均質であるほど，Woodcock tracking は非効率であることが知られている[8]．その理由は疎な領域における平均自由行程は密な領域よりも長く，次の散乱事象を見つけるまでに微小距離が何十回から何千回もインクリメントされるためである．

この問題を解決するために，我々はアダプティブでかつ統計的に不偏な方法を提案する．前処理で，提案法は媒質の解析領域を複数の部分空間に分割し，各部分空間内での媒質の分布がなるべく一様になるようにする．こうした部分空間は kd-木によってあらわす．レンダリング時は，散乱事象の位置を媒質の平均自由行程の空間変化に応じてアダプティブに決定する．また我々は提案法が統計的に不偏であることを保証する．提案法の重要な貢献は，サンプリングの効率を評価するコストモデルに基づいた自動な空間分割スキームである．提案法ではコストモデルに基づいた最適な分割位置を見つけるのに，largest empty rectangle problem を解く．また，提案法は図 1 に示すような空全体のような広大なシーンを含むさまざまなシーンを扱うことができる．非均質の度合いが高いシーンでは，提案法は従来法に比べて一桁から二桁以上の高速化を実現する．また，提案法は CUDA で実装されている．

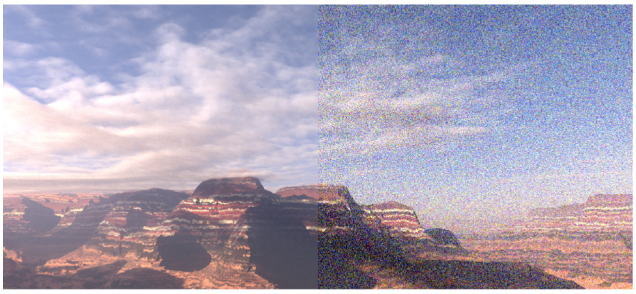

図 1. 同じ時間かけた場合の提案法(左)と従来法(右, Raab らの手法[4])の計算結果．この例では，提案法では同じレンダリング時間内に，従来法よりも 380 倍の光路をサンプリングできる．

提案法の重要な利点として，提案法は，関与媒質のレンダリング時に光路の生成を行うすべての従来法の高速化に用いることができる．そのような従来法には，メトロポリス光伝達法[3]やフォトンマップ法[2]も含まれる．また提案法は CG 分野に限らず，原子力科学やメディカル科学の分野にも応用できる．

2. 関連研究と本研究の位置づけ

関与媒質をレンダリングするには，放射輸送方程式(radiative transport equation) を解く(文献[9])．これまで様々な手法が提案されており，詳細なレビューについては文献[10,11]を参考されたい．Cerezo ら[10]によれば，これらの手法は，文献[12,13]のようなグリッドベースの手法や光路追跡法のようなモンテカルロ積分に基づくサンプリングベースの手法[2,3]がある．グリッドベースの手法では，精度の高い結果を得るためには細かくグリッドを分割する必要があり，複雑なシーンのレンダリングには向かない．

サンプリングベースの手法では，各ピクセルの色を計算

するために，散乱事象の集合から構成される多数の光路を生成する．散乱事象の位置を決めるため，レイマーチング法[2,3,14](またはその変形版，たとえば文献[15])がよく利用されるが，計算結果にはバイアスがあり，厳密解に収束しない．

不偏なサンプリングを行うため，Raab ら[4]は，原子力科学の分野から Woodcock tracking [7,16]（別名 delta-tracking, pseudo- scattering)を導入した．Woodcock tracking が不偏であることは，Coleman [6]によって証明されたが，非均質な関与媒質のレンダリングにおける計算効率が悪いことが知られている[8]．しかし過去数十年間，この問題を改善するための手法はほとんど提案されていない．近年，Woodcock tracking の GPU 実装が，例えば Badal ら[17]によって提案された．また，Leppänen[8]は，Woodcock tracking の改良法として，吸光の強い領域を光学的に薄い領域から分け，これら二つの領域を別々に扱う手法を提案した．しかし文献[8]の問題設定と異なり，CG で扱う媒質は通常連続的な分布をしており，いかにしてこのような二段階の分割を行うかが自明でない．

本研究では，Woodcock tracking [7]に，空間分割の概念を導入し，不偏なサンプリングを保証しつつ，レンダリングの効率化を図る．また，計算効率をなるべく最適にするような空間分割を自動的に行う方法を提案する．さらに提案法はシーンの複雑さに対してスケーラブルである．

3. Free Path Sampling

本節では free path sampling について述べ，レイマーチングと Woodcock tracking の問題点について述べる．

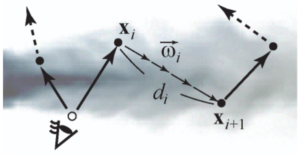

図 2. 光路の例．黒丸は散乱位置を表す．

モンテカルロ法に基づく方法では，ピクセルの輝度を推定するため，大量の光路をサンプリングして各光路の寄与を積分する．各光路は，図2に示すように，散乱事象の集合から構成される．散乱事象 i がすでに生成されたとして，散乱事象(i+1)を生成するには，散乱方向 $\vec{\omega}_i$ と次に散乱が起こる位置までの自由行程 d_i をサンプリングする．散乱方向をサンプリングするには，散乱位置における位相関数に基づいて，wavelet importance sampling 法[18]などにより重点的サンプリングを行う．自由行程 d_i は次の確率密度関数に従うように選ぶ[3]．

$$pdf_{dist}(\mathbf{x}_{i+1} = \mathbf{x}_i + d_i\vec{\omega}_i) = e^{-\tau(\mathbf{x}_i,\mathbf{x}_{i+1})}k(\mathbf{x}_{i+1}) \tag{1}$$

ここで，$\mathbf{x}_i$ と $\mathbf{x}_{i+1}$ は散乱事象 i と(i+1)の位置を表し，$\tau(\mathbf{x}_i,\mathbf{x}_{i+1})$ は $\mathbf{x}_i$ と $\mathbf{x}_{i+1}$ 間の光学的距離，$k(\mathbf{x}_{i+1})$ は $\mathbf{x}_{i+1}$ における消散係数を表す．上記の確率分布関数に従って距離を選ぶ方法として，逆関数法とレイマーチングを組み合わせた方法(乱数 ξ が与えられたとき，$\tau(\mathbf{x}_i + d_i\vec{\omega}_i, \mathbf{x}_i) = -\ln(1-\xi)$ を満たす自由行程 d_i をレイマーチングによって求める方法)がよく利用されるが，この方法は区分求積により光学的距離を求めるため，結果にバイアスが生じる．

Algorithm 1 WoodcockTracking ($\mathbf{x}_0, \vec{\omega}, k_M, d_{min}, d_{max}$)

Input: $\mathbf{x}_0, \vec{\omega}$: The ray starting at $\mathbf{x}_0$ in direction $\vec{\omega}$.
k_M : The majorant extinction coefficient.
d_{min}, d_{max}: The interval of the ray to evaluate.
Output: The free-path d to the next scattering event.

```
d ← d_min − ln(rand())/k_M
while d < d_max ∧ k(x_0 + dω)/k_M < rand() do
    d ← d − ln(rand())/k_M
end while
return d
```

Woodcock tracking (Algorithm 1 参照)は，棄却サンプリングに基づき，次に示すように自由行程を不偏にサンプリングする．まず媒質の消散係数の最大値 k_M を求める．次に，Woodcock tracking は，媒質を消散係数が k_M である一様媒質とみなして，擬似散乱事象をサンプリングする．こうした擬似散乱事象は確率 $k(\mathbf{x}_i + d_i\vec{\omega}_i)/k_M$ で，真の散乱事象として採用される．この方法は，Coleman[6]によって，得られる d_i が式(1)の確率分布に従い，不偏であることが証明されている． 乱数を生成しながら，自由行程 d_i は $-\log(rand())/k_M$ だけインクリメントされ，その期待値は $1/k_M$ である．非均質な媒質では，$k(\mathbf{x}_i + d_i\vec{\omega}_i)$ は k_M よりも小さく，密度の薄い領域では確率 $k(\mathbf{x}_i + d_i\vec{\omega}_i)/k_M$ は小さい．このため，真の散乱事象を見つけるまでに何度も乱数が生成され，Woodcock tracking は媒質の非均質な度合いが高いほど効率が悪い．

4. 不偏なサンプリングと空間分割

自由行程を効率的にサンプリングするため，提案法では，図 3 に示すように，空間を消散係数の分布(媒質の濃度分布とほぼ等価と考えてよい)に応じて分割し，kd-木で表現する．また，それぞれの部分空間の中で，k_M をその空間中の消散係数の最大値と設定する．これにより，疎な領域における一反復辺りに進む距離の平均が長くなり，かつ式(2)の条件を満たしやすくなるため，反復回数が減ることが期待される．

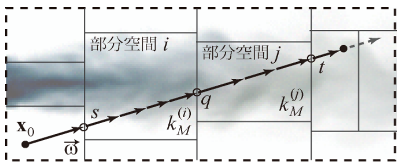

図 3. 提案法における空間分割とサンプリングの概念図

以下ではまず空間分割が与えられた場合に不偏なサンプリングを行う方法について議論し，続いてサンプリング効率をあげるための空間分割法について述べる．最後に得られた kd-木を用いたサンプリング法について示す．

4.1. 分割された空間における不偏なサンプリング

$\mathbf{x}_0$ を始点とし，方向 $\vec{\omega}$ に向かう光線を考え，図 3 に示すように部分空間 i と j を通過するとする．各部分空間内の最大消散係数を $k_M^{(i)}$ 及び $k_M^{(j)}$ とする．また，$\mathbf{x}_0$ からこれらの部分空間との交点までの距離を s, q, t とする．区間[s,t)に対して自由行程をサンプリングするため，まず一つ目の区間[s,q)に対して，Algorithm 1 で k_M，d_{min}，d_{max} の代わりに $k_M^{(i)}$, s, q を用いてサンプリングを行う．この結果，区間[s,q)内でおこる散乱事象がサンプリングされるか，または自由

行程が q を超えるかのいずれかである．もし自由行程が q を超えた場合，自由行程を q にリセットしてから二つ目の区間[q,t)に対して Algorithm 1 で k_M，d_{min}，d_{max} の代わりに $k_M^{(j)}$，q，t を適用してサンプリングを行う．このリセットの処理は文献[19]で異なる媒質境界を扱うために用いられている．この処理によって不偏なサンプリングを得られることは，Algorithm 1 で k_M，d_{min}，d_{max} の代わりに，$\max_{s\le z\le t} k(\mathbf{x}_0+z\vec{\omega})$，$s$，$t$ を用いた場合と上記の処理によって得られた自由行程 d がどちらも次式の確率分布に従うことを示すことにより確認できる(証明は付録を参照)．

$$\begin{cases} P(\mathbf{x}'=\mathbf{x}_0+d\vec{\omega}\wedge s\le d<t)=e^{-\tau(\mathbf{x}',\mathbf{x}_0+s\vec{\omega})}k(\mathbf{x}') \\ P(d\ge t)=e^{-\tau(\mathbf{x}_0+t\vec{\omega},\mathbf{x}_0+s\vec{\omega})} \end{cases} \tag{2}$$

4.2. 空間分割法

4.1 節で示した方法の処理効率は，空間分割の仕方に依存する．もし均質に近い空間を分割すると，リセットの処理を行う分反復回数が増えてしまう可能性がある．反対に空間を分割することで，部分空間の最大消散係数が空間全体の最大消散係数よりも十分に小さくなれば，反復回数が減ることが期待できる．これらの考察をもとに，以下では空間分割の終了条件と最適な分割位置について議論する．まず 1 次元の場合について述べ，次に 3 次元の場合へ拡張を行う．

一次元の場合．x 軸上に媒質が分布し，その消散係数の分布が与えられている例を考える．w 軸を消散係数の値を表す軸とし，区間[s,t)を考える．まず簡単な例として図 4(a)に示すように，区間[s,t)内で消散係数がただ一つ極大値を持つ例を考える．この区間を分割するかどうか，また最適な分割位置を決めるため，以下に示すような反復回数の増減に着目したコストモデルを考える．

区間[s,t)内の消散係数の最大値を k_M とすると，まず，区間を分割しない場合は，一反復あたりに進む距離が $1/k_M$ であり，区間の長さが(t-s)であるから，この区間を光線が通過するときに要する反復回数は平均で $N=k_M(t-s)$ である．次に，位置 q で分割した場合，q より左の区間内の最大消散係数を k_{ML} とすると，左右の区間を通過するのに要する反復回数の平均はそれぞれ $k_{ML}(q-s)$ と $k_M(t-q)$ となる．ここで反復の際に q をまたぐ場合は，位置を q にリセットする必要があるので，分割後の区間[s,t)を通過するのに要する反復回数の平均は，$N_{part}=k_{ML}(q-s)+k_M(t-q)+1$ となる．したがって，分割による反復回数の減少 N^{Δ} は

$$N^{\Delta}(q)=N-N_{part}=(k_{t,M}-k_{t,ML})(q-s)-1 \tag{3}$$

と表される．位置 q をこの反復回数の増減を最小にするように，すなわち

$$q=\underset{q'\in[s,t)}{\arg\max}\,N^{\Delta}(q') \tag{4}$$

によって選ぶとき，$N^{\Delta}>0$ ならば，分割により反復回数が減少するので，q での分割を採用するとレンダリングの計算コストを減らすことができる．

次に，より複雑なケースを扱えるように，式(4)の問題を

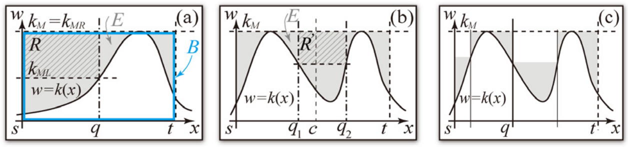

図 4. 一次元での媒質の消散係数の空間分布の例.

図 4(a)に示した最大長方形 R を見つける問題に帰着させる．図 4(a)を見ると，式(3)の $(k_{t,M}-k_{t,ML})(q-s)$ はグラフ上部の領域 E に接する長方形 R の面積を表すことがわかる．同様に，分割を行わない場合の平均反復回数 $k_M(t-s)$ は，バウンディングボックス B の面積を表すことがわかる．直感的に，B に含まれる領域 E はサンプリングにおける無駄な反復に相当し，E に含まれる最大の長方形 R を見つけ，これを除去することによってサンプリングの効率を改善できる．

上記の問題は次のように式(4)を

$$R=\underset{R'\subset E}{\arg\max}\,\Delta N(R') \tag{5}$$

と置き換えて定式化できる．ここで，N^{Δ} は，R'を E から取り除くことに対応する反復回数の減少を表し，$N^{\Delta}(R')=A(R')-T(R')$ と与えられる．$A(R')$ は，単に R'を E から取り除いたときの反復回数の減少で，R'の面積に等しい．$T(R')$ は分割によって増加する反復回数を表し，R'が媒質の境界に接する場合は 1，そうでない場合は 2 となる．

式(5)を満たす R を求めるには，2D largest empty rectangle problem[20]の変形版を解く．我々の問題では，長方形の面積を N^{Δ} に置き換える．解法として，E を x 軸に沿って n 個のビンに離散化し，動的計画法を用いて最大長方形を探索する．計算量とメモリ使用量はどちらも O(n)である．

図 4(b)中の R'が最大長方形として見つかったとする．q_1 と q_2 を R'の両端とする．Kd-木を構築するため，区間を一回だけ分割する．R'の両端のうち，区間の中心 $c=(s+t)/2$ に近いほうを選び，分割位置 q とする(図 4(c))．これによって，左右の区間の大きさをなるべく均等に保つ．空間分割は図 4(c)のように，これらの区間に対して再帰的に適用する．Kd-木は，内部ノードが分割位置を表し，葉ノードが対応する区間内の最大消散係数を記憶するように構築する．

三次元の場合．媒質の区間は三次元領域 V に置き換わり，V を分割する平面を決めることが目的になる．Kd-木を構築するため，座標軸に垂直な平面のみ考慮する．一次元の解法の単純な拡張として，$w=k(\mathbf{x}), w=k_M, V$ で囲まれた四次元領域内の最大超直方体を探索することが考えられるが，四次元の largest empty hyper-rectangle problem は計算量が高い(文献[21])．

その代わり，我々はヒューリスティックなアプローチをとり，kd-木を 1 分以内に構築できる方法を提案する．提案法は常に最適な木を構築するわけではないが，5.2 節に示すような複雑な媒質にも対応できる．提案法では，分割平面を見つけるために，各座標軸に沿った三つの一次元の問題を考え，各座標軸に対して候補を一つずつ選定し，最後にこれらの中からもっともよいものを選ぶ．

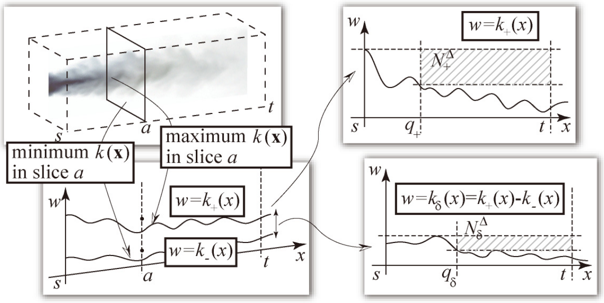

図 5. 関数$k_+(x)$と$k_\delta(x)$の構築方法と分割位置の計算について．各位置'a'について，x 軸に垂直なスライス'a'内を走査して，最大消散係数と最小消散係数($k_+(a)$と$k_-(a)$)を計算する．$k_\delta(a)$は$k_+(a)-k_-(a)$により定義する．

まず，x 軸について分割候補を見つける方法を説明する．まず x 軸に沿って，一次元関数$k_+(x)=\max_{y,z}k(x,y,z)$を構築する．この関数に対して，一次元の場合で述べた largest empty rectangle problem の解法を適用して最大長方形をさがしN_+^Δを得る．もし$N_+^\Delta>0$なら，対応する分割位置q_+を得る．この単純なアプローチはたいていの場合にうまくいくが，$k_+(x)$だけでは，密度の濃い領域に囲まれた薄い領域を見つけるのに十分ではない．そこで，二つ目の一次元関数$k_\delta(x)=\max_{y,z}k(x,y,z)-\min_{y,z}k(x,y,z)$を導入する．図 5 に，$k_+(x)$と$k_\delta(x)$の構成法を示す．$k_\delta(x)$は，$x$ 軸に垂直な x を通る平面内における媒質の分布が非均質であるほど大きく，反対に均質であるほど小さい．もしそのような均質な領域を領域全体から取り除けば，残るのは密度の薄い領域が密度の高い領域に囲まれている可能性の高い非均質な領域である．こうした均質な領域を見つける方法として，一次元の場合で述べた largest empty rectangle problem の解法が利用できる．$k_+(x)$に対してN_+^Δを得る操作と同じ操作を$k_\delta(x)$に適用してN_δ^Δを得る．もし$N_\delta^\Delta>0$なら，二つ目の分割位置q_δを得る．最後のステップは，こうして得られた二つの分割位置から，一つを選ぶことである．実験から，q_+をq_δより優先して使うほうがパフォーマンスがよいことから，我々は，N_+^Δと$F\cdot N_\delta^\Delta$を比較して，値の大きいほうに対応する分割位置を採用し，またこのときの値を x 軸で分割したときの利得の数値表現とする．Fは定数であり，0.7 に設定するのがよい．

上記の処理を，y 軸と z 軸に対しても適用する．サンプリングのコストを低くするために，各軸での分割の利得の中で最大の軸を探す．もしその値が 0 以下ならば，Vは分割しない．そうでないとき，対応する軸の分割位置に垂直な平面で分割する．

物体表面を扱うには，まず上記の kd-木構築のプロセスを媒質に対して適用する．次に各葉ノードについて，もし物体表面を含むならば，葉ノード内に物体表面のためのサブ kd-木を構築する．これによって，別々の kd-木を使って，物体との交点と散乱事象の計算を別々に計算する場合に比べて，処理時間を短縮することができる．

4.3. トラバーサル

構築された kd-木を用いた光線トラバーサルアルゴリズムを Algorithm 2 に示す．

Algorithm 2 kdTreeFreePathSampling ($\mathbf{x}_0$, $\vec{\omega}$)

Input: $\mathbf{x}_0$, $\vec{\omega}$: The ray starting at $\mathbf{x}_0$ in direction $\vec{\omega}$.
Output: The free-path d to the next scattering event.

```
1: loop
2:     kdTreeNode p ← findNextLeafNode()
3:     if p = nil then   return INFINITY   end if
4:     (d_min, d_max) ← intersectionBetweenRayAndNode(p)
5:     k_M ← majorantExtinctionCoefficientStoredIn(p)
6:     d ← WoodcockTracking(x_0, ω, k_M, d_min, d_max)
7:     d_isect ← INFINITY
8:     if intersectionWithObjectSurfaces(x_0, ω, p) then
9:         d_isect ← distanceToTheIntersection()
10:    end if
11:    if d ≥ d_isect then   return IntersectionEvent   end if
12:    if d < d_max then   return d   end if
13: end loop
```

Algorithm 2 にでは，まず現在のサンプリング位置に対応する葉ノードを探す．4 行目のd_{min}とd_{max}は，ノードの最近点と最遠点までの距離である．各葉ノード内で，Algorithm 1 を適用して散乱事象をサンプリングする．もし 8 行目に示すように，物体表面との交点が見つかれば，d_{isect}を交点までの距離として設定する．もし得られた自由行程 d がd_{isect}よりも長ければ，物体表面との交差を返す．もし d が葉ノード内であれば，散乱事象を返す．それ以外の場合は，光線の次の区間を処理するために次の葉ノードに移る．Algorithm 2 は極めてシンプルであり，一般的な kd-木を用いたポリゴンに対するレイトレーシングのアルゴリズム(例えば文献[22])に似ており，Algorithm 2 の 2, 4, 8, 9 行目は文献[22]のコードを共有する．

5. 評価

我々は三種類の実験を行った: 1) 計算の収束に関する比較実験，2) 詳細なパフォーマンスの比較実験，3) 自然な関与媒質のレンダリング．実験には，Intel Core2 Extreme QX9650 の CPU 及び nVidia GeForce GTX 295 の GPU を搭載した PC を使用した．5.1 節及び 5.2 節の実験には，CPU のコアを一つのみ使用した．評価を簡単にするため，光路追跡法に対して free path sampling の方法を複数実装して比較を行った．レンダリングでは，多重散乱と相互反射を考慮し，ルシアンルーレットによって光路の打ち切りを決めることによって，確率的に任意の回数の散乱と反射を扱えるようにした．すべての実験で，空間分割に要した時間は 1 分以下であり，ノイズが十分に低減されるまでに要するレンダリング時間と比べて無視できる．

5.1. 収束に関する比較実験

まず，レイマーチング，Raab らの手法[4]と提案法の計算時間と結果の収束に関する比較結果を図 5 に示す．この実験では，図 5 左に示す煙をそれぞれの手法でレンダリングした場合の，計算時間に応じた RMS 誤差を図 5 右上のグラフに対数-対数スケールで示した．レイマーチングについては，ランダムオフセットによってエリアシングの問題を回避し，シーンの平均自由行程の最小値($1/k_M$)を基準にして，サンプリング間隔をその 4 倍，1 倍，1/4 倍，1/16 倍に設定した場合を考慮した．レイマーチングによる計算にはバイアスがあるので，特に 4 倍と 1 倍の結果に顕著に見られるように，グラフが途中で曲がり厳密解に収束していないこと

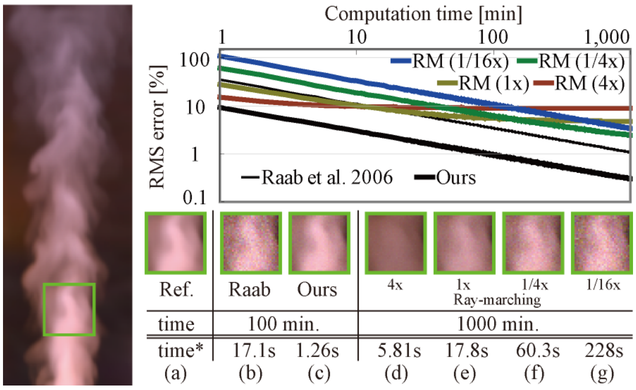

図 6. 左: 参照画像. (a): 参照画像. (b), (c): Raab ら[4]の方法と提案法でレンダリングした場合. (d): レイマーチングにてサンプリング間隔を $1/k_M$ の 4, 1, 1/4, 1/16 倍に設定した場合. time*: 10^6 本の光線をサンプリングするのに要した時間

がわかる. サンプリング間隔を細かくすれば, より正しい解に収束するが, 計算に要する時間が長くなる. 提案法はこの実験で比較した他のどの手法よりも速く収束し, Raab らの方法との比較では, 誤差を同じレベルまで減らすのにかかる時間は提案法の方が 10 倍以上短い.

5.2. 詳細なパフォーマンス評価

Algorithm 1 と 2 を用いた場合の free path sampling に要する時間を様々な媒質に対して比較する. 幅広く媒質をカバーするため, 濃度の違い(消散係数の違い), 分布の局在性の違い, 濃度変化の激しさの違いを勘案してテスト用の媒質を作成した. その際, 密度が薄くて大きな媒質と密度が濃くて小さな媒質は同じ外観であるという事実(位置と消散係数, 散乱係数, 吸光係数の無次元化により確かめられる)に基づき, 媒質の大きさを固定して, 消散係数の大きさを変化させた. 濃度変化の激しさの違いは三次元のパーリンノイズ[23]のオクターブ値 (大きな値ほど濃度変化が激しい)によりコントロールする.

まず, オクターブ値を 1 から 8 まで変化させて 8 種類のベースとなる媒質を用意し, 消散係数を[0,1]の範囲に正規化する. 次に, 各ベース媒質に対して, コントラストを $2^j, (j=1,\ldots,9)$ に変更して 9 種類のバリエーションを準備する. ここで, コントラストは消散係数の平均値に対する消散係数の最大差異の比と定義する. コントラストを変更するために, 適切な n 乗を適用した. こうして得られた 72 種類の媒質に対して, 最大消散係数を基準値の 1/8, 1/4, 1/2, 1, 2, 4, 8 倍に変更して 7 セットの媒質を準備し, これら 504 種類の媒質に対してパフォーマンス評価を行った.

Algorithm 1 と 2 は同じ確率分布に従って自由行程をサンプリングするので, 10^6 本の光線を生成するのに要する時間は公平に比較できる. 最大消散係数が基準値である場合の媒質のレンダリング結果を図 7(a)に示し, また Algorithm 2 の Algorithm 1 に対する高速化比を図 7(b)に示した. 他の 6 セットについての高速化比の傾向は同様であり, スケールのみが異なるので, 6 セットすべてについて高速化比を示す代わりに, 最大高速化比と最小高速化比を図 7(c)に示した. 媒質の消散係数が大きな値であるほど,

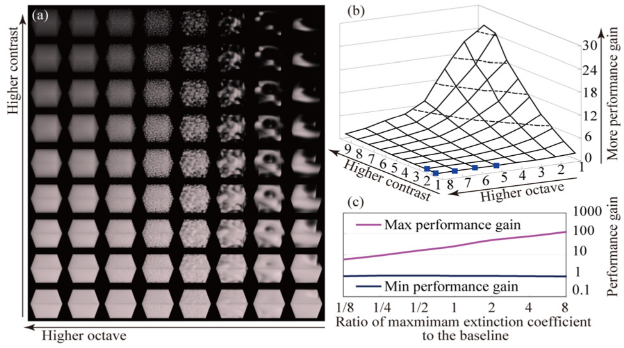

図 7. (a): 最大消散係数が基準値である媒質のセットの外観. (b): (a)のセットの場合の高速化比. (c): 各セットについての最大高速化比と最小高速化比.

提案法が高速であることがわかる.

媒質のコントラストが高く, 低いオクターブ値から生成されたものであるほど, 提案法による高速化比が高い. 図 7(a)に示すように, 密度の高い領域が狭い領域に密集しているほど, 提案法は効率的である. 自然界に存在する関与媒質はこの性質を持つものが多い. 図 7(b)の青い四角で示したようにいくつかの例で提案法が遅い場合があるが, これらは媒質がほぼ均質場合であり, またパフォーマンスの低下はわずか数%である.

次に, レンダリングプロセス全体の高速化比について論じる. レンダリングプロセス全体では, free path sampling のほかに, シェーディングや散乱方向のサンプリングの処理がある. 我々の実装では, これらの処理は全体の半分程度の時間を占めるため, レンダリングプロセス全体の高速化比は free path sampling の高速化比の約半分である.

5.3. レンダリング結果

提案法によるレンダリング例について, 炎, 蒸気, 雲と大気の例を図 8 に示す. Raab らの方法を用いた場合と比較して, 同等の画質を得るのにかかるレンダリング時間は, 炎, 蒸気, 雲と大気の例でそれぞれ 1/2.7, 1/8.9, 1/380 に短縮された. 一般に, 雲の消散係数は大気の 100～1000 倍であり, 空は極めて非均質な媒質である. このため, 提案法による高速化の度合いが大きい. なお, kd-木の構築にかかった時間はいずれも数秒程度であり, レンダリングにかかった時間に比べて無視できる. なお, CUDA を用いて提案法を実装し, GeForce GTX 295 で計算を行った場合は, CPU による計算に比べて, 50 倍程度高速であった.

6. まとめと今後の課題

本稿では, 非均質な関与媒質の効率的なレンダリングのためのアンバイアスな重点的サンプリング法を提案した. 提案法では, 空間を分割して, 一反復で進む距離の平均を媒質の消散係数に適応させることにより, 疎な領域での反復回数を減らして高速化を実現した. また, 空間分割問題を *Largest Empty Rectangle Problem* に帰着して, 高速化の度合いをなるべく最大化する方法を提案した. 提案法は従来のアンバイアスな方法と比べて, シーンによっては数桁の高速化を実現した. また CUDA により GPU で計算することで CPU による計算よりも 50 倍程度高速に計算すること

図 8. 提案法によるレンダリング結果.

ができた．今後の課題として，提案法を他のモンテカルロ積分に基づくレンダリング法(例えばメトロポリス光輸送法など)に実装することが挙げられる．

参考文献

[1] E. P. Lafortune, Y. D. Willems, “Rendering participating media with bidirectional path tracing,” Proc. EGWR 1996, pp. 91-100 (1996)

[2] H. W. Jensen, P. H. Christensen, “Efficient simulation of light transport in scenes with participating media using photon maps,” Proc. SIGGRAPH 1998, pp. 311-320 (1998)

[3] M. Pauly, T. Kollig, A. Keller, “Metropolis light transport for participating media,” Proc. EGWR 2000, pp. 11-22 (2000)

[4] M. Raab, D. Seibert, A. Keller, “Unbiased global illumination with participating media,” Proc. Monte Carlo and Quasi-Monte Carlo Methods 2006, pp. 591-605 (2006)

[5] E. Veach, “Robust Monte Carlo methods for light transport simulation,” PhD thesis, Stanford University (1998)

[6] W. Coleman, “Mathematical Verification of a Certain Monte Carlo Technique and Applications of the Technique to Radiation Transport Problems, ” Nuclear Science and Engineering, 32, pp. 76-81 (1968)

[7] E. Woodcock, T. Murphy, P. Hemmings, T. Longworth, “Thechniques used in the GEM code for Monte Carlo neutronics calculations in reactors and other systems of complex geometry,” Proc. Conference on the Application of Computing Methods to Reactor Problems, ANL-7050, pp. 557-579 (1965)

[8] J. Leppänen, “Development of a New Monte Carlo Reactor Physics Code,” PhD thesis, Helsinki University of Technology (2007)

[9] S. Chandrasekhar, “Radiative Transfer,” Dover Publications (1960)

[10] E. Cerezo, F. Pérez, X. Pueyo, F.J. Serón, F.X. Sillion, “A survey on participating media rendering techniques,” The Visual Computer, 21 (5), pp. 303-328 (2005)

[11] D. Gutierrez, H.W. Jensen, W. Jarosz, C. Donner, “Scattering,” SIGGRAPH ASIA Courses (2009)

[12] J. Stam, “Multiple scattering as a diffusion process,” Proc. EGWR 1995, pp. 41-50 (1995)

[13] T. Nishita, Y. Dobashi, E. Nakamae, “Display of Clouds Taking into account multiple anisotropic scattering and sky light,” Proc. SIGGRAPH’96, pp. 379-386 (1996)

[14] K. Perlin, E.M. Hoffert, “Hypertexture,” Proc. SIGGRAPH’89, pp. 253-262 (1989)

[15] F.B. Brown, W.R. Martin, “Direct sampling of Monte Carlo flight paths in media with continuously varying cross-sections,” Proc. ANS Mathematics & Computation Topical Meeting (2003)

[16] I. Lux, L. Koblinger, “Monte Carlo Particle Transport Methods,”

[17] A. Badal, A. Badano, “Accelerating Monte Carlo simulations of photon transport in a voxelized geometry using a massively parallel graphics processing unit,” Medical Physics, 36(11), pp.4878-4880 (2009)

[18] P. Clarberg, W. Jarosz, T. Akenine-Moller, H.W. Jensen, “Wavelet importance sampling: efficiently evaluating products of complex functions,” Proc. SIGGRAPH 2005, pp. 1166-1175 (2005).

[19] L. Carter, E. Cashwell, W.M. Taylor, “Monte Carlo sampling with continuously varying cross-sections,” Nuclear Science and Engineering, 48, pp. 403-411 (1972)

[20] A. Aggarwal, S. Suri, “Fast algorithms for computing the largest empty rectangle,” Proc. Third Annual Symposium on Computational Geometry, pp. 278-290 (1987)

[21] J. Edmonds, J. Gryz, D. Liang, R.J. Miller, “Mining for empty spaces in large data sets,” Theoretical Computer Science, 296(3), pp. 435-452 (2003)

[22] A. Keller, “Quasi Monte Carlo methods for realistic image synthesis,” PhD thesis, University of Kaiserslautern (1998)

[23] K. Perlin, “Improving noise,” Proc. SIGGRAPH 2002, pp.681-682 (2002)

付録．不偏性の証明

証明 Algorithm 1 によって得られる自由行程が式(2)の確率分布に従うことは Coleman[5]によって証明されているので，ここでは 4.2 節の方法が式(2)の確率分布に従うことを示す．Algorithm 1 が自由行程をサンプリングする確率分布より，一つ目の区間においては，

$$P_{(\mathrm{i})}(d, s \le d < q) = e^{-\tau(d,s)} k_t(d), \quad P_{(\mathrm{i})}(d \ge q) = e^{-\tau(s,q)}$$

なる確率で，d_1 がサンプリングされることが分かる．同じように，二つ目の区間においては，

$$P_{(\mathrm{ii})}(d, q \le d < t) = e^{-\tau(d,q)} k_t(d), \quad P_{(\mathrm{ii})}(d \ge t) = e^{-\tau(q,t)}$$

なる確率で d_2 がサンプリングされる．二つ目の区間のサンプリングは，一つ目の区間に対するサンプリングで，d_1 が q を超えたときしか実行されないので，結局，

$$P(d, s \le d < q) = P_{(\mathrm{i})}(d, s \le d < q) = e^{-\tau(d,s)} k_t(d)$$

$$P(d, q \le d < t) = P_{(\mathrm{i})}(d \ge q) \cdot P_{(\mathrm{ii})}(d, q \le d < t)$$
$$= e^{-\tau(q,s)} e^{-\tau(d,q)} k_t(d) = e^{-\tau(d,s)} k_t(d)$$

$$P(d \ge t) = P_{(\mathrm{i})}(d \ge t) \cdot P_{(\mathrm{ii})}(d \ge t) = e^{-\tau(q,s)} e^{-\tau(t,q)} = e^{-\tau(t,s)}$$

となる．■