I-004

フォトンマップ分割に基づく並列画像生成アルゴリズム

A Parallel Image Generation Algorithm based on Partitioning of Photon Maps

田村壮秀* 滝沢寛之*† 小林広明*†
Masahide Tamura Hiroyuki Takizawa Hiroaki Kobayashi

1 緒言

大域照明モデルを用いた画像生成は光学的な現象を忠実に再現しているため，写実的な画像を生成することが可能であり，バーチャルリアリティやエンターテイメントなど，さまざまな分野で求められている．大域照明モデルを用いた最も基本的な手法であるレイトレーシング法に関しては，ソフトウェアとハードウェアの両面から数多くの高速化の研究がなされており [1] [2]，現在ではインタラクティブフレームレートでの画像生成が可能になっている [3]．

しかし，レイトレーシング法では全ての大域照明効果をシミュレートすることは非常に困難である．例えば，間接照明や集光模様などを扱うことは不可能である．このため，より高品質な画像を得るためには計算量が多いパストレーシング法やフォトンマッピング法が必要であり，それらの高速化，リアルタイムレンダリングが強く求められている．

そこで本研究では，フォトンマッピング法によるリアルタイムレンダリングを実現するための，専用ハードウェアの開発を目指している．本稿では，専用ハードウェアの実装を念頭において，リアルタイムレンダリングを実現するための並列フォトンマッピング法を提案する．特に，フォトンマッピング法を用いて画像生成を行う際に，高速化を妨げているフォトンマップの構築とフォトン検索を並列化する．

従来のフォトンマッピング法 [4] では，単一のフォトンマップを構築し，それを用いて特定のフォトンを探索していたため，処理が集中する．そこで，処理の集中を回避するためにフォトンマップを分割し，それぞれのフォトンマップでフォトン探索を並列に行う手法を提案する．フォトンマッピング法の高速化に関する研究として，集光模様の生成に特化した高速化手法 [6] がある．これは集光模様にのみフォトンマップを用いて，並列に計算を行っているが，パストレーシング法において高周波ノイズが問題となる間接照明などは扱っていない．また，並列化によるフォトンマップ高速化手法 [7] はフォトンマップの構築を並列化し，フォトン探索をする際にはそれらを統合して用いている．これらに対し，我々の提案手法では分割したグローバルフォトンマップを分割したまま並列に扱う．フォトンマップの構築とフォトン探索のそれぞれの処理が並列化されるため，全体の処理時間の大幅な短縮が可能である．本稿では提案手法の有効性を実験により評価する．

2 フォトンマッピング法

フォトンマッピング法とは，光をモデル化したフォトンを光源から追跡することによって，物体空間に格納されている全てのフォトンの分布を保持したフォトンマップを作成し，そのフォトンマップ内の照明情報を使用することでレンダリングする手法である．

以下，フォトンマッピング法の詳細について述べる．

- フォトン追跡
  1. 図 1 のように，光源からフォトンを射出し，その軌跡を追跡する．
  2. 拡散反射面に当たったら，そのフォトンをフォトンマップに格納する．
- レンダリング
  1. 図 2 のように，視点からレイ (視線) を射出する．
  2. 物体とレイが交差した地点の近くにあるフォトンを，フォトンマップを用いて探索する．
  3. 探索により発見されたフォトンを使って交点近傍のフォトン密度を推定し輝度を計算する．

この手法は，1) 全ての大域照明効果を計算することができる，2) 点集合であるフォトンを扱うため使用するメモリが少ない，といった利点がある．ただし，サンプリング手法を用いて照明情報を蓄積するため，フォトンの数が十分でない場合にはノイズが発生し，正確な値を計算することができなくなるという欠点もある．

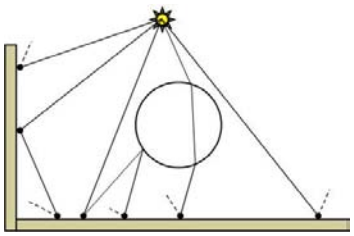

図 1 フォトン追跡

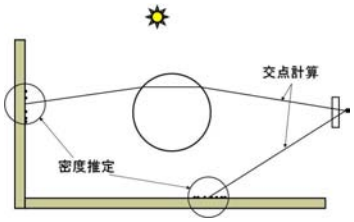

図 2 レンダリング

2.1 フォトン探索の高速化

レンダリングにおいてフォトンを探索する処理を高速化するため，フォトンマップのデータは kd ツリー構造に再構築されるのが一般的である．

kd ツリーの構築は以下のようにして行う．

* 東北大学大学院情報科学研究科
† 東北大学情報シナジー機構

1. 物体空間の次元軸の内，長さが最大のものを選択する．
2. 図 3 のように，選択した軸と直行する平面により，フォトンの数が均等になる位置で空間を分割する．
3. 上記の処理を繰り返してフォトンを並べ替え，kd ツリーを構築する．

N 個のフォトンを有する kd ツリーから 1 つのフォトンを探し出すための平均時間は O(log N) となる．

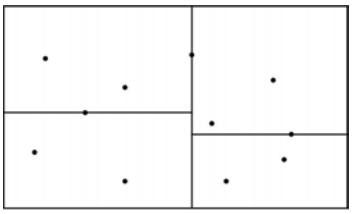

図 3　kd ツリーによるフォトンデータの構造化

2.2　フォトンマッピング法の問題点

フォトンマッピング法は，パストレーシング法などと比べ，大幅に計算効率が向上しているが，それでもなおインタラクティブフレームレートでのレンダリングは依然として困難である．以下に高速化を妨げている問題点を述べる．

2.2.1　フォトンマップの構築

静的なシーンやウォークスルーアニメーションを生成する場合には，フォトンマップを一度だけ構築すれば良い．しかし，動的なシーンを扱う場合には，フレームごとにフォトンマップを構築しなければならない．そのため，フレームごとにフォトンを追跡し，kd ツリーに再構築することが必要である．この処理の計算コストは非常に大きいので，高速化が求められる．

2.2.2　密度推定のコスト

図 2 に示されているレンダリングにおいては，レイと物体との交点におけるフォトン密度を推定する必要がある．このために，フォトンマップに格納されているすべてのフォトンの中から，交点近傍のフォトンだけを探し出す処理を繰り返す必要がある．そのフォトン探索処理に要する時間は，フォトンマッピング法による画像生成時間中で大きな割合を占めている．この問題は，使用するフォトンが多い場合にさらに深刻化するため，高品質な画像生成においてはフォトン密度推定処理の高速化が非常に重要である．

3　フォトンマップ分割に基づく並列アルゴリズム

前節では，フォトンマッピング法の高速化における問題点を述べた．これらの問題を軽減し，フォトンマップの構築，および密度推定を高速化するため，フォトンマップのデータ分割に基づく並列処理を提案する．

通常のフォトンマッピング法では，全体で単一のフォトンマップを作成し，それを用いて以下の計算を行うことで輝度を推定する．

$$L_r(x,\vec{\omega}) \approx \frac{1}{\pi r^2}\sum_{p=1}^{K} f_r(x,\vec{\omega}_p,\vec{\omega})\Delta\Phi_p(x,\vec{\omega}_p) \qquad (1)$$

ここで，L_r は放射輝度値，r は密度を計算する際に用いる半径，K は密度推定に用いるフォトン数，f_r は双方向反射率分布関数，$\Delta\Phi_p$ は各フォトンの出力である．

しかし，1 つのフォトンマップを共有しながらフォトンマップの動的再構築やフォトン探索を実行する場合，処理や通信の集中が発生するために高い並列化効率を達成するのが困難である．そこで，フォトンマップ構築に用いるフォトンの集合を分割し、フォトンマップ処理に内在するデータ並列性に着目する．分割によって得られた n 個の部分フォトンマップを考える．ただし，n 個の部分フォトンマップを全て合わせれば，元のフォトンマップと同じものになる．フォトンの集合は乱数で生成されるため，フォトンの集合を単純に n 個に分割すればよい．この場合，式 (1) を以下のように変形することができる．

$$\begin{aligned} L_r(x,\vec{\omega}) \approx \frac{1}{\pi r^2}\Big(&\sum_{p=1}^{K_1} f_r(x,\vec{\omega}_p,\vec{\omega})\Delta\Phi_p(x,\vec{\omega}_p)+ \\ &\sum_{p=K_1+1}^{K_2} f_r(x,\vec{\omega}_p,\vec{\omega})\Delta\Phi_p(x,\vec{\omega}_p)+ \\ &\vdots \\ &\sum_{p=K_{n-1}+1}^{K} f_r(x,\vec{\omega}_p,\vec{\omega})\Delta\Phi_p(x,\vec{\omega}_p)\Big) \end{aligned} \qquad (2)$$

このように変形することで，式 (2) のそれぞれの項を独立した複数の処理として並列に扱うことが可能となる．ただし，式 (2) からわかるように，r の値は全てのフォトンマップにおいて同じでなければならない．つまり，密度推定の際にフォトンを探索する範囲を固定する必要がある．そのため，従来のフォトンマッピング法で用いられる k 近傍探索 (交点に近い k 個のフォトンを探索し，フォトンまでの最長距離を r とする手法) による密度推定結果と一致しない．しかし，k 近傍探索による密度推定もフォトンの密度を近似する手法のひとつにすぎず，k の値を基準に範囲を指定するか，r の値を基準に範囲を指定するかの違いがあるだけである．よって，本手法でもフォトン密度の近似値を求めることができ，従来手法と遜色のない高品質な画像を非常に高速に生成可能である．

フォトンマップ数を n とした場合の，提案アルゴリズムを図 4 に示す．

まず，光源から N/n 個のフォトンを射出してフォトンマップを構築し，それを kd ツリー構造に再構築する．これらは n 個並列に処理し，n 個のフォトンマップを構築する．ここで，並列に行っているフォトン生成処理において，フォトンはランダムな方向に射出される．そのため，それぞれのフォトンマップにおいて，フォトンの分布に極端な偏りは生じないと考えられる．次に視点からレイを射出し，物体との交点を計算する．そして，各フォトンマップを用いて交点から半径 r の範囲内にあるフォトンを探索し，密度推定を行って輝度を計算する．この処理も n 個並列に実行し，n 個の計算結果を得る．最後に n 個の結果を合計し，最終的な値とする．

このようにして，全処理時間のうち大きな割合を占めるフォトンマップの構築と密度推定を並列化し，高速化を図る．さらに各フォトンマップが含むフォトン数を $1/n$ にすることができるため，フォトン探索のコストを削減することも可能である．

フォトンマッピング法のハードウェア実装を考える場合，本手法は式 (2) の各項の処理を分散メモリ型並列処理で実行可能

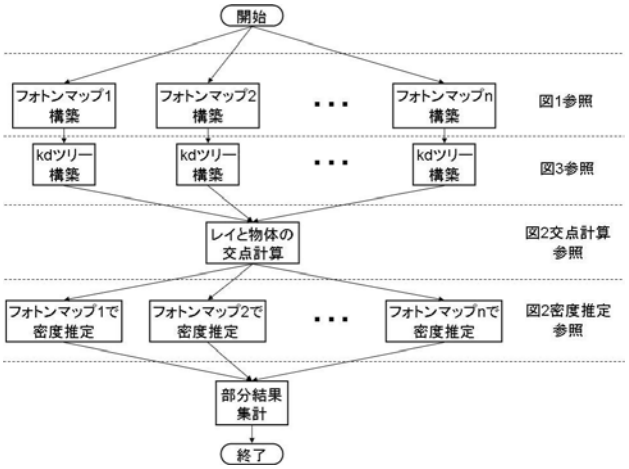

図 4　提案アルゴリズム

であるため，比較的簡素なハードウェア構成で効果的な並列処理を実現可能である．また，並列処理によって得られた計算結果を統合する際に集団通信 (gather) が必要となるが，送受信されるデータサイズが小さいために，並列化効率に与える影響は小さい．本手法の並列化効率と画質との関係に関しては次節で評価する．

4　性能評価

前節で提案した手法を検証するために実験を行った．実験には表 1 の 2 つのモデルデータを使用した．

表 1　モデルデータ

	happy_budha	cornel_box
解像度	480 × 480 [pixels]	480 × 480 [pixels]
パッチ数	20062	16190
反射面	拡散のみ	拡散・鏡面

フォトンの初期生成数は 10^5 個とした．r の値は k 近傍探索における k の値と同様経験則から，今回は物体空間全体の 1 辺の長さの 1/20 程度とした．フォトンマップ数を 1 個から 32 個まで変化させ，1 つのフォトンマップあたりの密度推定に必要となる計算時間を実機を使って計測した．実験に使用した PC の CPU は Pentium4 3.0GHz，メモリは 512MB である．

並列数を変化させた場合の密度推定時間の速度向上率を図 5 と図 6 に示す．この結果から，フォトンマップ数の増加に伴って，密度推定の計算速度は大幅に上昇していることが分かる．ここで，並列度 n の場合に計算速度が n 倍にならないのは，2.1 節で述べたようにフォトン探索の時間が O(log N) であるためだと考えられる．また，式 (2) より，理論上は 1 つのフォトンマップ内に 1 つのフォトンという N 並列度まで，計算速度を上昇させることが可能である．ただし，並列度が上がるにつれ，gather のオーバヘッドが表面に現れ、速度向上率を低下させることが考えられるが，そのオーバヘッドを考慮した全体の評価は今後の課題である．次に，従来のフォトンマッピング法と提案手法の画像を比較すると，図 7 と図 8，図 10 と図 11 のように，視覚的にはほとんど差がないことが分かる．また，図 9，図 12 に輝度の誤差を視覚化した差分画像を示す．これらの図より，誤差が特定の場所に集中していないことが分かる．

これらの結果から，提案手法で処理を並列化することによって，顕著な画質劣化なしに飛躍的な速度向上を得られることが明らかになった．また提案手法は，並列処理の際に各処理要素間でのデータの共有や同期を必要としない．このためハードウェア構成を簡素化することか可能であり，さらに高い並列化効率を達成できることが期待できる．そのようなハードウェア実装とその評価が今後の課題である．

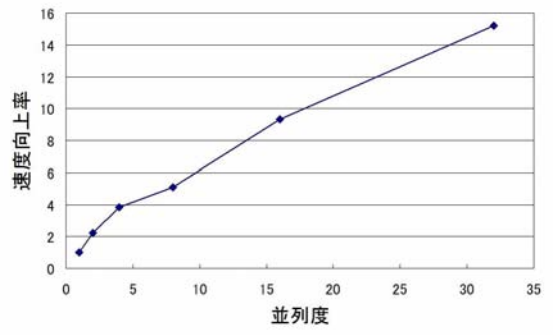

図 5　happy_budha の結果

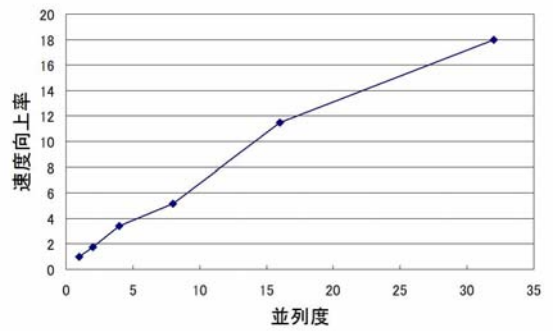

図 6　cornel_box の結果

5　結言

フォトンマッピング法による画像生成は，高品質な画像を生成できる反面，計算量が膨大になってしまうという欠点がある．そこで本稿では，フォトンマッピング法を用いて画像生成を行う際に，フォトンマップの構築やフォトン探索を並列化することで，高速化を行う手法を提案した．

評価実験より，提案手法が顕著な画質低下なしに高い並列化効率を達成できることが明らかになった．

今後の課題は，光源からフォトンを放射する際に生じるコヒーレンシ (空間的一貫性) のないレイの扱い方を改善することである．コヒーレンシがないレイの場合には，キャッシュ内の物体データを有効に再利用することが困難であるため，メモリへのデータアクセスが非常に多くなる．そのため，コヒーレンシのないレイはコヒーレンシのあるレイと比べて，数倍の計算時間が必要となる [5]．このため，高い実効性能を達成するためには，コヒーレンシを高める手法を検討する必要がある．

また，本稿の結果を踏まえて最適なフォトンマッピング法専

図 7　happy_budha 従来手法

図 8　happy_budha 提案手法

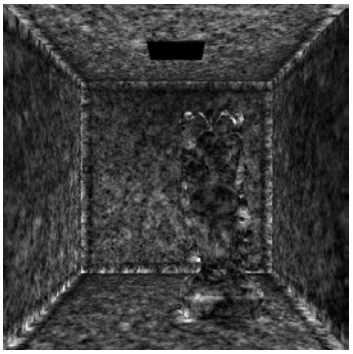

図 9　happy_budha 差分画像

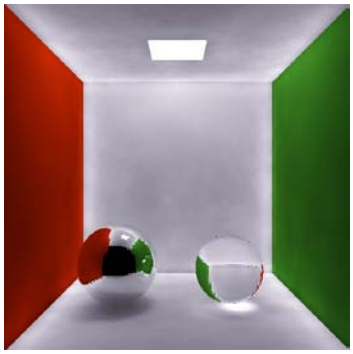

図 10　cornel_box 従来手法

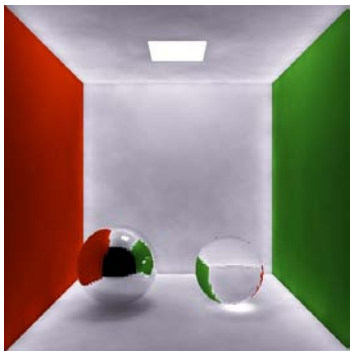

図 11　cornel_box 提案手法

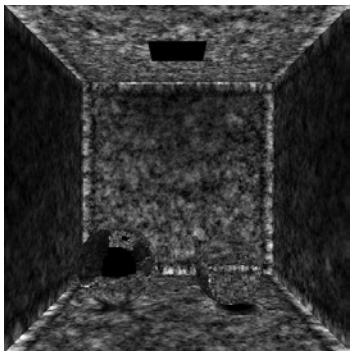

図 12　cornel_box 差分画像

用ハードウェアの構成を提案し，その構成の下で，画像生成の処理全体のシミュレーションを行い，性能評価することも重要な今後の課題である．さらに，シミュレーションの結果を基にして，フォトンマッピング法専用ハードウェアを FPGA ボードに実装し，リアルタイムレンダリング実現の可能性について評価する予定である．

参考文献

[1] Matt Pharr, Greg Humphreys, “ Physically Based Rendering: From Theory To Implementation (The Interactive 3d Technology Series) ” , Morgan Kaufmann Pub, 2004.

[2] Jorg Schimittler, Ingo Wald, Philipp Slusallek, “ SaarCOR - A Hardware Architecture for Ray Tracing ”, In Proceedingsof the ACM SIGGRAPH/Eurographics Conference on Graphics Hardware (2002), pp. 27-36.

[3] Sven Woop, Jorg Schimittler, Philipp Slusallek,“ RPU:A Programmable Ray Processing Unit for Realtime Ray Tracing ” , Proceedingsof the ACM SIGGRAPH, 2005.

[4] Henrik Wann Jensen, ”フォトンマッピング　ー実写に迫るコンピュータグラフィックス”, オーム社, 2002.

[5] Ingo Wald,“ Realtime ray tracing and interactive global illumination ” , 2004.

[6] Johannes Günther, Ingo Wald, Philipp Slusallek. ”Realtime Caustics Using Distributed Photon Mapping”, Rendering Techniques 2004: 15th Eurographics Workshop on Rendering, June 2004, pp. 111-122.

[7] Toshi Kato. ”Kilauea: parallel global illumination renderer”, Parallel Computing, Vol. 29, Issue 3, March 2003, pp. 289-310.