グラフィクスとCAD 95－4
（1999．4．22）

光束追跡のための再帰的領域分割法の改良

小山 和広†, 岡田 稔††, 内川 嘉樹†††

名古屋大学大学院工学研究科 † 電子情報学専攻, ††† 計算理工学専攻
†† 中部大学工学部工業物理学科

あらまし： 光線追跡法における膨大な計算量を削減するために，以前我々はビームトレーシング法を改良した一手法を提案したが，その手法では定義物体が単一凸多面体に限定されていた．そこで本論文では，これを凹多面体を含む複数の多面体を対象として拡張する．本手法は，光線追跡木のコヒーレンスに基づいてスクリーンをオーバーラップのない領域に再帰的に分割し，その各領域内で光線を追跡することにより画像を生成することができるというものである．この分割は画像生成全体にかかる時間に対して十分に短い時間で行うことができ，また画像サイズに依存しない．これにより，光線追跡法の大幅な高速化とビームトレーシング法におけるポリゴンの重なりによる反復計算の削減が可能である．さらに，将来的にはシーンが変化する状況への対応が期待される．

A Recursive Region Segmentation Method for Improved Beam Tracing

Kazuhiro KOYAMA†, Minoru OKADA†† and Yoshiki UCHIKAWA†††

†Department of Information Electronics, †††Department of Computational Science and Engineering,
Graduate School of Engineering, Nagoya University
††Department of Engineering Physics, College of Engineering, Chubu University

Abstract: We recently proposed an improved beam tracing method to reduce the enormous calculation cost in ray tracing, which is limited that a scene consists of a convex polyhedron. In this paper we expand this for concave and several polyhedra. Our proposed method is based that a projection screen is recursively divided into non-overlapping coherent region segments based on the coherence of ray-trace-tree structure of a given scene and an image can be synthesized after the segmentation by tracing one or several rays in each region segment. This segmentation can be done for a sufficient short time compared with the total time to synthesize an image. Therefore, we can reduce the computational cost in ray tracing rendering, and eliminate the recursive calculations caused by the overlapped polygons on a screen in beam tracing by using our proposed method. Finally, it is hoped that our method can apply to the moving scene.

1 はじめに

光線追跡法 [1] は反射，屈折，陰影などの様々な光学的現象をシミュレートすることができるため，リアルな CG (Computer Graphics) を生成するのに有効な手法の一つである．しかしながら，従来の光線追跡アルゴリズムは視点とスクリーン上の各画素とを結ぶ光線を再帰的に追跡するため，膨大な計算量が要求されるという大きな欠点があった．そこで，この欠点を解決するために以下のように 2 つに大別される様々な研究がなされてきた．

一つは各画素ごとの計算量を削減しようとするものである．これを目的とした手法としては，バウンディングボリュームを用いて物体を囲む手法 [2] やオクトリ構造を用いて三次元空間を分割する手法 [3] などが挙げられる．もう一つは，隣接する画素間のコヒーレンスを利用して，物体と光線との交差判定の回数そのものを削減しようとするものである．例えば，ビームトレーシング法 [4] や画素選択型光線追跡法 [5] などの手法が提案されている．

我々はすでに，ビームトレーシング法を改良した一手法を提案している [8][9]．しかしながら，この手法では定義物体が単一凸多面体に限定されていた．そこで本論文では，これを鏡面物体と拡散物体によるシーンに限定しつつ，凹多面体を含む複数の多面体に対応するように拡張した．これにより，シーン内の多面体の形状や数に関係なく本手法を適用することができる．

本手法とビームトレーシング法との大きな相違点は，ビームトレーシング法では可視領域部分の塗り重ねを基本としているのに対して，本手法は予め光線追跡木のコヒーレンスに基づいてスクリーンをオーバーラップのない領域に再帰的に分割し，その各領域内で光線を追跡することにより画像を生成することができるということである．分割された各領域内にある画素と視点とを結んだ光線は反射や屈折など同じような振る舞いをするので，各領域内のすべての画素に対して光線を追跡する必要はない．また，このスクリーンの再帰的分割は，画像生成全体にかかる時間に対して十分に短い時間で行うことができ，さらに画像サイズや光源の位置に依存しない．本手法を用いることにより，光線追跡法の大幅な高速化とビームトレーシング法におけるポリゴンが重なっている部分の画素に対する反復レンダリングの削減が可能である．

2 コヒーレンスと領域分割

2.1 光線追跡木のコヒーレンス

一般に，スクリーン上の近傍の画素間の光線は同じような振る舞いをする [4][5][6][7]．言い換えると，光線追跡木は同じ構造（コヒーレンス）を持っていることが多い．このようなコヒーレンスを持つ光線の束を本論文では光束（beam）と呼ぶ．多面体のみによるシーンを想定すると，多面体を構成するそれぞれのポリゴン上では光線は同じように反射，屈折する．従って，多面体のポリゴンあるいは頂点に注目することにより，この光線追跡木のコヒーレンスの変化の境界を決定することができる．

本論文では，光線追跡木の深さを以後‘レベル’と呼ぶことにする．レベル 1, 2, 3, n はそれぞれ 0, 1, 2, $n-1$ 回の光線の反射，屈折に対応している．

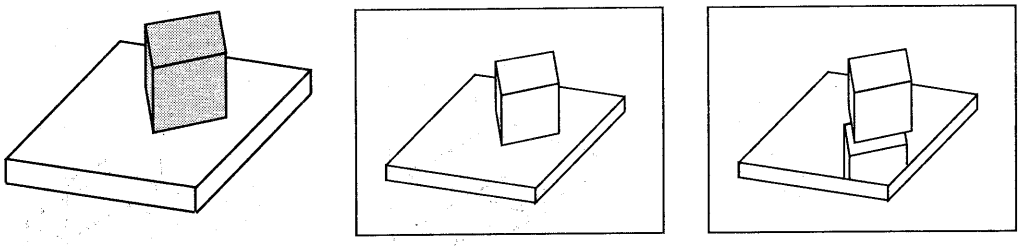
2.2 スクリーンの再帰的領域分割

図 1 に，本手法で対象としている 3D シーンの一例とそのスクリーンへの投影の例を示す．図 1(a) は，不透明な立方体と鏡面多面体である板であり，これらをスクリーンに投影しオーバーラップのない領域に分割すると，結果は図 1(b), (c) のようになる．

図 1(b) は，レベル 1 におけるスクリーンの領域分割の例である．背景を含めて 7 つの領域があり，その各領域内にある画素と視点とを結ぶ光線は**2.1** で述べたように同じような振る舞いをすることになる．また，この図は，シーンの中にあるすべての多面体が拡散面で構成されている場合を表しているとも言える．

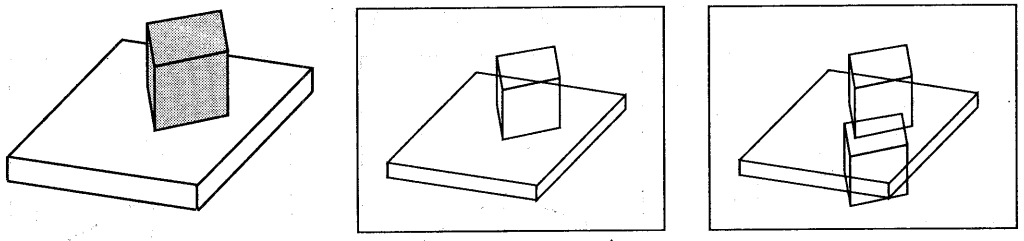
一方，図 1(c) は，レベル 2 におけるスクリーンの領域分割の例である．レベル 2 以上では光線の反射や屈折を考慮しなければならない．そこで，レベル 2 では多面体を構成する一つのポリゴンに注目し，それを含む無限平面に対して多面体の頂点を変換する．レベル 3 以上に対しては，その前のレベルで変換されたポリゴンに注目し，同様のことを再帰的に繰り返す．その後注目したポリゴンとそれに対して変換されたポリゴンをスクリーンに投影し，注目ポリゴンでクリッピングを行うことにより，スクリーンが分割される．

以上のことから，次節で述べるようなスクリーンの再帰的領域分割アルゴリズムを構築することができる．



(a) 不透明な立方体と鏡面反射板　(b) レベル 1 における領域分割　(c) レベル 2 における領域分割

図 1: 多面体で構成されるシーンの一例と本手法におけるそのスクリーンへの投影の例



(a) 不透明な立方体と鏡面反射板　(b) レベル 1　(c) レベル 2

図 2: ビームトレーシング法におけるスクリーン上でのイメー ジ図

ここで，本手法と Heckbert らによるビームトレーシング法 [4] との相違点について述べる．

ビームトレーシング法は，一つの可視領域に映り込むポリゴンを再帰的に探索し，それらを順に走査しながらポリゴンを一つずつ重ねてレンダリングしていくという手法である．しかしながら，この手法ではスクリーン上でポリゴンがオーバーラップしている領域が存在する可能性がある．この手法ではスクリーンを分割していないが，本手法と比較するために，もし仮に映り込むポリゴンをスクリーンに投影した場合，図 2(b), (c) のようになると考えられる．つまり，ポリゴンがオーバーラップしている領域内にある画素に対しては繰り返しレンダリングをしていることになる．

これに対して本手法は，スクリーンを予めポリゴンのオーバーラップのない領域に再帰的に分割するという手法である．そして，その分割された各領域内でレンダリングを行う．この点で，本手法とビームトレーシング法は大きく異なる．そして，このスクリーンの再帰的領域分割は画像サイズに依存しないため，一度スクリーンを分割しておけば，その後は任意のサイズの画像を高速に生成することができるという利点がある．さらに，物体が移動した場合に対しては，その前の分割結果を利用してスクリーンを分割することがができるため，シーンが変化する状況にも対応することが可能である．

2.3 基本アルゴリズム

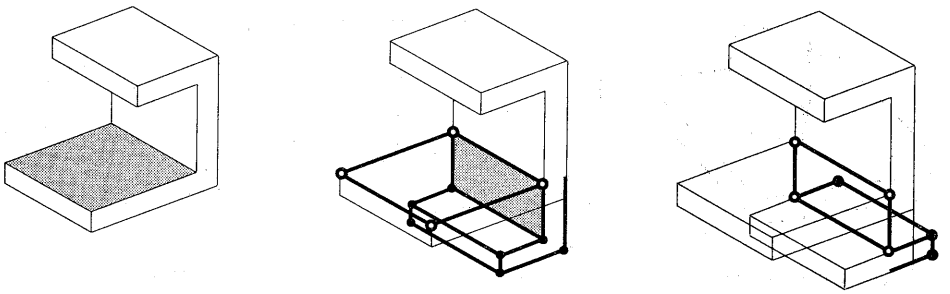
本節では，図 3(a) のような鏡面凹多面体を例に挙げて，本手法におけるスクリーンの再帰的領域分割の基本アルゴリズムを説明する．ここで，ある頂点を定義物体を構成する一つのポリゴンを含む無限平面に対して対称な位置（鏡像の位置）に変換することを反射変換と呼ぶことにする．また，図 3 は定義物体と反射変換の例を，図 4 は一つのポリゴンに注目した場合の各レベルにおけるスクリーンの領域分割の結果を示している．

[Algorithm]
step I

(1) 定義物体のすべての頂点を算出する（図 3(a) では 16 個）．
(2) 可視ポリゴンを見つけ（図 3(a) では 6 個），その一つに注目する（図 3(a) におけるグレーのポリゴン）．

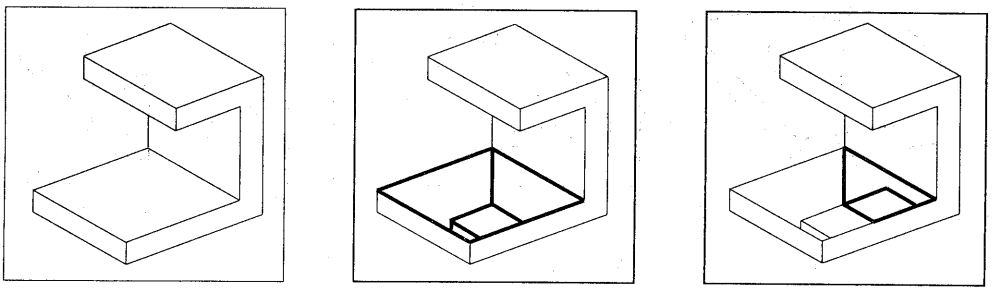
step II

(1) 視点を反射変換する．



(a) 鏡面凹多面体　(b) レベル 2 における反射変換　(c) レベル 3 における反射変換

図 3: 定義物体の一例と反射変換の例



(a) レベル 1　(b) レベル 2　(c) レベル 3

図 4: 一つのポリゴンに注目した場合の各レベルにおけるスクリーンの領域分割の結果

(2) 変換した視点から注目ポリゴンを含む無限平面を通して見ることのできるポリゴン上の頂点を反射変換する (図 3(b), (c) における●の頂点).
(3) 注目ポリゴンを含む無限平面上にある頂点は変換しない (図 3(b), (c) における○の頂点).

step III

(1) step II で反射変換された頂点を結ぶことより生成されるポリゴン (以後, 反射変換されたポリゴンと呼ぶ) と注目ポリゴンをスクリーンに投影する (図 4(b), (c) における太線).
(2) 反射変換されたポリゴンを注目ポリゴンでクリッピングする (図 4(b), (c) における太線).
(3) 反射変換されたポリゴン同士で隠面消去を行う.
(4) 反射変換されたポリゴンに対して他の可視ポリゴンで隠面消去を行う.
(5) クリッピングされたポリゴンの一つと step II で反射変換された視点をそれぞれ次の注目ポリゴンと視点とする (図 3(b) におけるグレーのポリゴン).

step IV

(1) step I で見つけたすべての可視ポリゴンと step III でクリッピングされたすべてのポリゴンに対して, step II と step III を再帰的に繰り返す.
(2) 可視ポリゴン同士で隠面消去を行う.
(3) あらかじめ設定したレベルを越えた場合, または分割された領域が拡散面あるいは無限遠平面である場合, このアルゴリズムを終了する.

本論文では, 基本アルゴリズムの step II において反射変換のみを考慮している. tangent の法則ではなく Snell の法則に基づいた屈折変換を導入した場合, 変換後の座標を求めるためには四次方程式を解く必要がある [8][9]. さらに, 定義物体が凹多面体あるいは複数の多面体である場合には, 2 回以上の屈折変換を施さなければならない. 例えば, シーンの中に図 5 に示すような単純な単一凹多面体のみが存在する場合でさえも, X_1, X_2 の 2 つの無限平面に対して屈折変換を施さなければならない. つまり, 連立四次方程式を解かなければならない. シーンが複雑になればなるほど, 屈折変換の回数も多くなり, 計算量は非常に膨大になる. 従って, 本論文では透明多面体は定義物体の対象外とした.

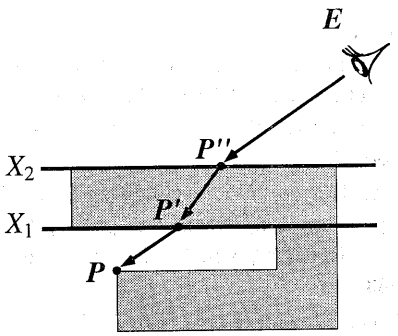


図 5: 凹多面体における屈折変換の一例

3 マッピング

本手法を用いて実際に画像を生成する際には，スクリーンの再帰的領域分割を行った後，その分割された各領域内でマッピングをすることが必要である．定義物体として本手法のように鏡面多面体を想定した場合，その表面上における輝度 I は，一般に次式のように表すことができる．

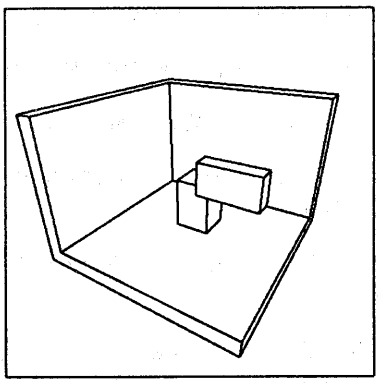
$$I = I_a + k_s I_s \qquad (1)$$

ここで，I_a は光源や視点の位置に依存せず一定である環境光成分であり，I_s, k_s はそれぞれ入射光成分，鏡面反射係数である．式 (1) を再帰的に適用することにより，定義物体の表面の実際の輝度を計算することができる．ビームトレーシング法では，輝度の計算に用いる係数はすべて定数としていた [4]．この鏡面反射係数 k_s は，厳密には Fresnel の式に従うが，本手法ではこれを補間によって求めることを考えている．これにより，リアルな画像が高速に生成できることが期待される．

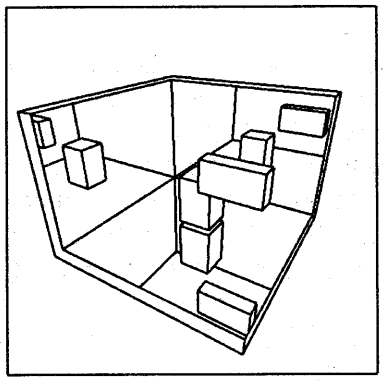
4 実験と結果

2.3 で述べたスクリーンの再帰的領域分割アルゴリズムを用いて実験を行った．シーンの中の多面体は CSG (Constructive Solid Geometry) によって定義されている．図 6 はその結果である．図 6(a), (b) において，分割された領域の数はそれぞれ 13, 37 であり，分割されるまでの時間はそれぞれ 0.112 秒，0.160 秒であった．使用機材は SGI ORIGIN200 (R10000/180MHz) である．今回の測定時間には，シェーディングやマッピングなどの画像を生成するための時間は含まれていない．しかしながらこの結果は，スクリーンの再帰的領域分割は画像を生成するためにかかる全体の時間と比較して十分に短い時間で行うことができるということを示している．また，スクリーンの分割は画像サイズに依存しないため，一度スクリーンを分割すれば，その結果が任意の画像サイズにそのまま適用することができる．本手法は，画像サイズが大きい程，また分割された領域の数が少ない程，言い換えると，光線追跡木のコヒーレンスが高い程，光線追跡法に対して大幅な高速化が可能であると言える．

最後に，以下のような仮定のもとで光線追跡法に対する本手法の相対的な有効性を定量的に評価した．



(a) レベル 1



(b) レベル 2

図 6: スクリーンの再帰的領域分割の結果

- 画像サイズは 512 × 512 画素である.
- スクリーンの再帰的領域分割時間は 5 秒である.
- 分割された領域の数は 1000 である.
- 分割された各領域内で一つの画素に対して光線を追跡する.
- 分割された各領域はフラットシェーディングである.

本手法を用いて画像生成を行った場合, 光線追跡法を用いた場合と比較して約 70 分の 1 に計算時間が削減されることが期待される.

5 まとめ

本論文では, 以前我々が提案したビームトレーシング法を改良した一手法において, 定義物体を鏡面物体と拡散物体に限定しつつ, 単一凸多面体から凹多面体を含む複数の多面体に対応することができるように拡張して, スクリーンの再帰的領域分割を行った. これにより, シーン内の多面体の形状や数に関係なく本手法を適用することができる. また, この分割に要する時間は画像生成全体に要する時間に対して十分に短いため, 光線追跡法の大幅な高速化が可能である.

今後の課題としては, 本手法のアルゴリズムの改良, 分割された各領域内に対するマッピング手法の考案と, 実際にマッピングを行って画像を生成することによる評価が挙げられる.

最後に, 例えば物体が移動する場合には, それに関係する部分だけを再び再帰分割すればよい. つまり, 大部分はその前の分割の情報を利用することができると考えられる. 従って, 将来的にはシーンが変化する状況に対応させることが可能であろう.

参考文献

[1] T. Whitted, "An Improved Illumination Model for Shaded Display", *Communication of the ACM*, Vol. 23, No. 6, pp. 343–349, 1980.

[2] J. T. Kajiya, "New Techniques for Ray Tracing Procedurally Defined Objects", *Proc. of SIGGRAPH'83*, Vol. 17, No. 3, pp. 91–102, 1983.

[3] G. Wyvill, T. L. Kunii, and Y. Shirai, "Space Division for Ray Tracing in CSG", *IEEE Computer Graphics and Application*, Vol. 6, No. 4, pp. 28–34, April 1986.

[4] P. S. Heckbert, and P. Hanrahan, "Beam Tracing Polygonal Objects", *Proc. of SIGGRAPH'84*, Vol. 18, No. 3, pp. 119–127, 1984.

[5] T. Akimoto, K. Mase, and Y. Suenaga, "Pixel-Selected Ray Tracing", *IEEE Computer Graphics and Application*, Vol. 11, No. 4, pp. 14–22, 1991.

[6] J. Amanatides, "Ray Tracing with Cones", *Proc. of SIGGRAPH'84*, Vol. 18, No. 3, pp. 129–135, 1984

[7] M. Shinya, T. Takahashi, and S. Naito, "Principles and Applications of Pencil Tracing", *Proc. of SIGGRAPH'87*, Vol. 21, No. 4, pp. 45–54, 1987.

[8] 冨澤 良明, 河合 善之, 岡田 稔, "光束追跡法における領域分割のための基礎検討", 情処技報, GCAD86–4, pp. 17–22, 1997

[9] K. Koyama, Y. Kawai, Y. Tomizawa, and M. Okada, "A Recursive Region Segmentation Method of Projection Screen for High Speed Ray Tracing", *Proc. of VSMM'98*, Vol. 1, pp. 339–343, 1998

[10] 中前 栄八郎, 西田 友是, "3 次元コンピュータグラフィックス", 昭晃堂, 1997