

Progressive Refinement Radiosity 法のための高速化手法

土井章男

東京基礎研究所、日本アイ・ピー・エム (株)

本論文では、プログレッシブ・リファインメント・ラジオシティ(Progressive Refinement Radiosity) 法を高速化するための手法について述べる。最初にラジオシティ法におけるフォームファクタ(幾何形状係数)を高速に計算するために Nusselt analog を利用したフォームファクタ計算方式を提案する。本方式は、フォームファクタ計算を半球の底面上で行うため、精度よく、かつ高速にフォームファクタ値を求めることができる。次に並列システムを用いた並列ラジオシティ法について述べる。我々は、Power Visualization System (PVS) 上で並列ラジオシティ法を開発し、本システムを種々の幾何データに適用してその有効性を評価した。

Acceleration Techniques for Progressive Refinement Radiosity

Akio Doi

Tokyo Research, IBM Japan, Ltd.

1623-14 Shimotsuruma, Yamato-shi, Kanagawa-ken 242, Japan

In this paper, we describe acceleration techniques for progressive refinement radiosity. For complex environments, the progressive refinement radiosity method with an adaptive subdivision scheme needs a large amount of memory and a huge computation time to capture all the details of illumination, such as highlights, penumbra, and sharp shadows. Our approach utilizes a new technique for calculating form factors, which projects each element simultaneously onto a hemisphere base, and samples visible objects by using a multi-scanline technique. This sampling method plays an important role in our rendering algorithm, and makes our approach practical and efficient. We also describe a parallel radiosity system on Power Visualization System (PVS). The system is applied to several geometric data generated by a CAD system, and is evaluated in various ways.

1 はじめに

プログレッシブ・リファインメント・ラジオシティ法は、相互反射を計算しながら対話的にフォット・リアリスティック (Photo Realistic) な画像を作成することができ、漸近的に最終結果に収束する [2]。しかしながら、プログレッシブ・リファインメント・ラジオシティ法の計算の大部分はフォームファクタ計算であり、より正確で高速なフォームファクタ計算アルゴリズムが求められている。本論文では、この問題点を解決するためにヘミスフィアベースを用いたアルゴリズムを提案する。本論文で用いる幾何データのデータ構造葉、ポリゴン/パッチ/エレメント構造 [1] とする。このデータ構造は、光の放射単位としてポリゴンをまずパッチに分割し、光を受ける単位として、さらにパッチをエレメントに分割する。階層は、常にポリゴン・パッチ・エレメントの3階層とし、それ以上、深い階層は作らない。

2 ヘミスフィアベースを用いたフォームファクタ計算アルゴリズム

Nusselt's analog として、フォームファクタは、ヘミスフィアベースに投影された面積に等しいことが証明されている。この変換は世界座標における平面が曲面に投影されるために、単純には投影できない。ここでは、投影するエレメントが4角形か3角形であることを用いて、エレメントを構成する領域を折れ線で近似することを考える。各頂点の投影は、最初にヘミスフィアベースの中心を視点とするビュー座標系に変換し、各頂点とサンプル点までの距離 $length$ を求める。ヘミスフィアベース上の座標 (x_{HS}, y_{HS}, z_{HS})

は、

$$(x_{HS}, y_{HS}, z_{HS}) = (x_{VC}/length, y_{VC}/length, 0.0) \quad (1)$$

$$length = \sqrt{x_{VC}^2 + y_{VC}^2 + z_{VC}^2} \quad (2)$$

で与える。この変換により、各エレメントは、半径1の円上に投影される。実際には、各面の奥行き計算を行うために、各頂点の座標 (x_{SC}, y_{SC}, z_{SC}) は、

$$(x_{VC}/length, y_{VC}/length, length) \quad (3)$$

を用いる。次に線分ごとの角度 t を

$$\cos(t) = |(x_{HSi}, y_{HSi})| * |(x_{HSi+1}, y_{HSi+1})| \quad (4)$$

を計算する。このとき、精度を保つために、この角度 t がしきい値より大きい場合は、世界座標におけるその線分間に中間点 $(x_{HSmid}, y_{HSmid}, z_{HSmid})$ を発生させて、しきい値以下になるまで折れ線近似を行う (図1)。我々は、しきい値として、1度を用いている。

中間点における z 座標は、投影したエレメントの平面性を保つため、(3) 式で用いた3点 $(x_{HS1}, y_{HS1}, length_1)$, $(x_{HS2}, y_{HS2}, length_2)$, $(x_{HS3}, y_{HS3}, length_3)$ を通る平面の方程式を求め、平面の方程式の係数 A, B, C, D から、

$$z_{SC} = \frac{A * x_{HSmid} + B * y_{HSmid} + D}{C} \quad (5)$$

により求める。

折れ線近似する回数は、エレメントの面積が大きい場合、増加するが、現実には、ラジオシティ法の場合、ポリゴンを分割する際にエレメントが小さくなるため、この計算は、増加しない。

投影されたポリゴンは、S/W Z-buffer 法、S/W スキャンライン法、あるいは、グラフィックワークステーションの H/W Z-buffer を用いて隠れ面処理を行う。Z-buffer 法に関しては、エレメントの Item buffer を作成し、各エレ

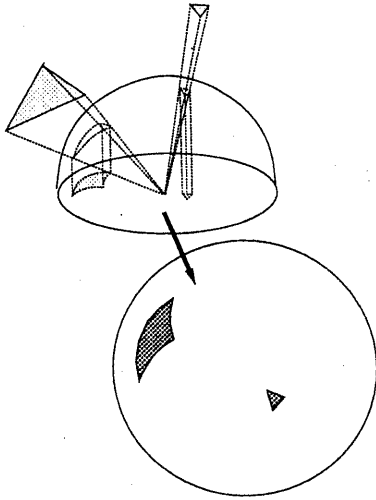


図 1: ヘミスフィアベースを用いたフォームファクタ計算

ントのピクセルの総和にスケールファクタ Unit をかければよい (式 (7))。

$$F_{ij} = \sum (\text{Item buffer pixel}_j) * \text{Unit} \quad (6)$$

$$\text{Unit} = \frac{4.0}{\pi * \text{image resolution}^2} \quad (7)$$

また、スキャンライン法では各スキャンラインの線分長を加算することにより、より正確なフォームファクタを求めることができる。

このとき、使用する Unit はスキャンライン数を N とすると次式で表される。

$$\text{Unit} = \frac{4.0}{\pi * N * 2.0} \quad (8)$$

どちらの方式を用いても、デルタフォームファクタは用いる必要がない。

3 評価

パッチ P_i からのフォームファクタ $F_{P_i-e_j}$ を計算する方法としては、1) ヘミキューブ法 [3]、2

) 四面体によるヘミキューブ法 [4]、3) "Proxels" を用いた方法 [5]、4) 解析的にフォームファクタを計算する方法 [6]、5) レイ・トレーシングを用いてフォームファクタを計算する方法 [7]、6) ヘミスフィアを用いる方法 [8] 等がある。各方法の詳細は参考文献に譲るとして、ここでは本アルゴリズムを最もよく使用されているヘミキューブ法と比較検討を行った。

表 1、2 に分割数をかえて生成した幾何データに対する CPU 時間を示す。計測に使用したワークステーションは、IBM RS/6000-340 である。スキャンコンバージョンには、どちらも、S/W Z-buffer 法を用いた。図 2 は、その生成結果である。約 2 倍以上のスピードアップを実現できた。

表 1: CPU Time and Visual Quality

Methods	CPU Time	Visual Quality
Hemi-Cube	21.6 sec.	Good
Hemi-Sphere	6.7 sec.	Good

* Faces: 18, Patches: 132, Elements: 548, Iterations: 50, Resol.: 100x100

表 2: CPU Time and Visual Quality

Methods	CPU Time	Visual Quality
Hemi-Cube	351.4 sec.	No Good
Hemi-Sphere	135.9 sec.	Good

* Faces: 18, Patches: 328, Elements: 3294, Iterations: 100, Resol.: 100x100

本アルゴリズムは、サンプリングする領域もヘミキューブ法に比べて小さく、投影する回数も 1 回でよい。また、精度を向上させるために点サンプリングだけでなく、水平及び垂直方向における可変型の線サンプリングを行うことができる。さらにデルタフォームファクタを用いる必要がないため、精度上、デルタフォームファクタを用いる従来手法に比べて、有利であ

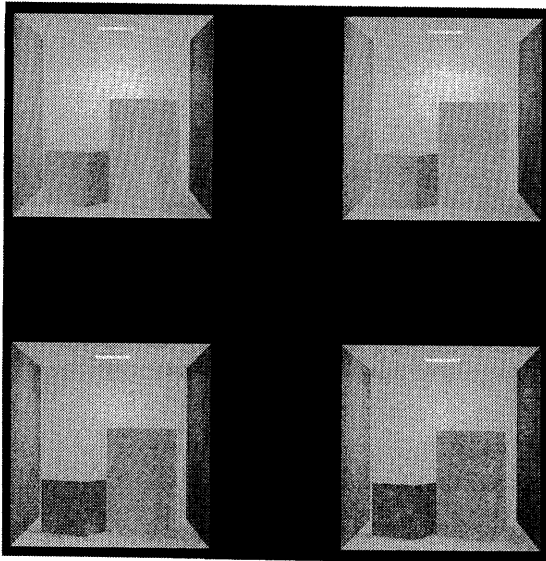


図 2: Visual Quality

る。また、本方式は、線サンプリングを行う際にもスキャンラインの線分長を加算して、最後にその合計に UNIT 分をかければ良いため、処理が簡単である。

4 双方向ラジオシティ法

2章で述べたフォームファクタ計算アルゴリズムを用いて、我々は、双方向ラジオシティ法を開発した。本手法では、以下の流れで処理される。

(1) 前処理 (3 角形分割、ボクセル分割、パッチ・エレメント分割) (2) 直射光の輝度計算および再帰的パッチ・エレメント分割 (3) 間接光の拡散反射成分、鏡面拡散成分の計算 (4) 画素単位での直射光の輝度計算および鏡面反射成分の計算 (5) 生成画像の表示

対象としている幾何データは、ポリゴン、2 次曲面、および、パラメトリック曲面であり、すべての面は、3 角形か 3 角形メッシュに分解されて、ボクセル空間に登録される。点光源、平行光源、および拡張フォームファクターの計算において、レイと物体の交差判定を高速化す

るために、本手法では空間分割法を用いている。物体空間を有限空間と見なし、この有限空間を $N_x * N_y * N_z$ 個のボクセルに分割して、レイの進む方向に関与するボクセルだけの交点計算に限定することで効率向上を図っている。ここで、 N_x, N_y, N_z は、それぞれ X 軸、Y 軸、Z 軸方向に空間を等分する際の分割数を表す。このボクセル空間を用いることにより、数十倍から数百倍の高速化が図れる。

直射光の輝度は各エレメントの頂点ごとに計算し、各エレメントの頂点の輝度値の差の大きいものは、さらに小さなエレメントに分割する。各頂点の輝度は、パッチ単位で集計して、未放射エネルギーとして各パッチに保管する。

間接光の計算は、未放射エネルギーの最大のパッチから順に、RMS エラーが十分、小さくなるまで放射を繰り返す。

安定したラジオシティ画像を生成するために直射光の輝度値の差が大きいエレメントに関しては、画素単位で直射光成分の再計算を行なう。点光源および平行光源の計算は、レイトレーシングを用いる。面光源には、完全拡散光源と仮定して、2章で述べた手法を用いている。ただし、この場合は面光源のヘミスフィア上における Mini-Max 値を用いて、処理の高速化を図っている。

画素単位で処理されるエレメントの選択には、次の判定基準を用いた。

$$\begin{aligned} \max(R_{max} - R_{min}, G_{max} - G_{min}, B_{max} - B_{min}) \\ > R_{diff} \end{aligned} \quad (9)$$

ここで、 $(R_{max}, G_{max}, B_{max})$ は、 $(\max(R_i), \max(G_i), \max(B_i))$ を $(R_{min}, G_{min}, B_{min})$ は、 $(\min(R_i), \min(G_i), \min(B_i))$ をそれぞれ、意味している。

また、 (R_i, G_i, B_i) は、各エレメントの頂点 i における輝度値である。 R_{diff} は、ユーザ指定

のしきい値である。

5 並列処理

ラジオシティ法は、使用するデータ量、計算時間が非常にかかるため、並列化が容易なアルゴリズムを考案することは、有効と思われる。並列処理システムは、大きく、共有メモリ型並列計算機と分散メモリ型並列計算機に分けられる。

我々が使用した並列システムは、共有メモリ型並列計算機、IBM 7245 Power Visualization System (PVS) で、Intel860-XP を 32 個、装備し、各 CPU が共用できるメモリ (共有メモリ) は 512 メガ・バイト持ち、各 CPU ごとにローカルメモリとして 16 メガ・バイトを持つ。

これは、ラジオシティ法が大量のバッチ・エレメントデータをメモリ空間に保持しながら、同時に更新する必要があることや、一般にレンダリングのプログラミングが複雑であることを考えると、使いやすい共有メモリ型並列計算機である PVS は、たいへん、並列レンダリングに向けたマシンであると思われる。

本インプリメントでは、幾何形状、属性データ、環境データ、テキストチャー、フレームメモリ等を共有メモリに保持し、レンダリング時に必要なワーク領域 (奥行きバッファ、アイテムバッファ等) は、ローカルメモリにおいた。

並列化を行った箇所は、頂点における直射光の輝度計算、フォームファクタ計算、画素単位での直射光計算、画像生成の部分である。

頂点における直射光の輝度計算は、独立に行なうことができるため、CPU を順次、割り当て、頂点計算が終了するたびに、次の CPU を割り当てた。フォームファクタ計算は、定数個のバッチに対するフォームファクタを並列に求めた後、順次、その値を用いて共有メモリ上のラジオシティ量を更新するようになった。

画像生成に関しては、隠れ面消去をスクリー

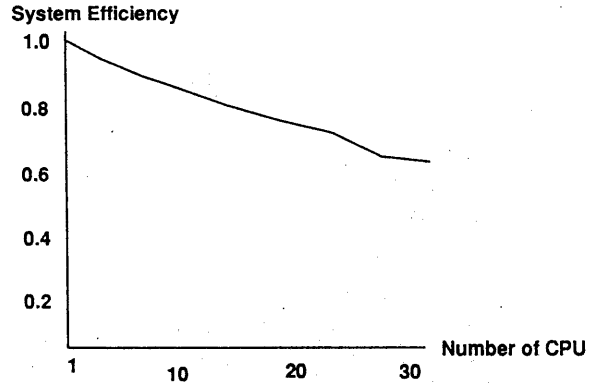


図 3: CPU に対するシステム有効性

ン領域単位で並列化した。あらかじめ分割された領域ごとに各 CPU を割り当て、計算を終了した CPU を順次、次の領域に割り当てるようにした。そして、1 回の投影が終了するたびにその結果を HDTV 画面に表示した。

図 3 に CPU を増加させた場合におけるシステムの有効性を示す。並列システムの有効性 T は、式 (10) を用いた。

$$T = T(1)/(T(C) * C) \quad (10)$$

ここで、 C は、CPU の数、 $T(C)$ は、 C 個の CPU を用いた時のソリューション時間である。

6 おわりに

現在、コンピュータ・グラフィックス (CG) で生成される画像は、よりリアルで、複雑な幾何形状を対象にしている。今後、その傾向はワークステーションの高速化にともない、より加速度を増していくと思われる。我々は、このような必要性に答えるためにフォームファクタを求める新しいアルゴリズムを提案した。また、並列処理システムを用いて、より複雑な幾何データにも適用できる環境を構築した。並列にフォーム

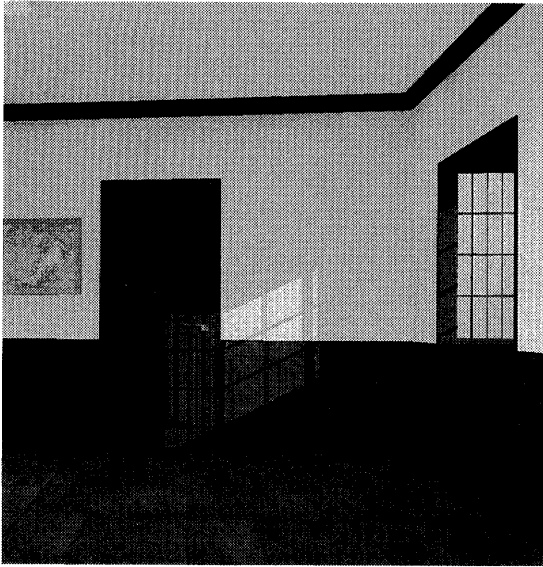


図 4: 表示結果

ファクターを求めることにより、全体の処理速度を向上させることができた。図4は、CADシステムから得られた幾何データを本システムで生成したものである。ポリゴン数は、10,288、生成されたパッチおよびエレメント数は、23,855、53,430である。作成時間は、1回のパッチの放射ごとに画像を生成しながら、約352回の収束計算を行ったものである。計算時間は、約2時間要した。

参考文献

- [1] M. F. Cohen, D. P. Greenberg, D. S. Immel and P. J. Brock, *A Radiosity Solution for Complex Environments*, IEEE Computer Graphics and Applications, Vol. 6, No. 3, pp. 26-35, 1986.
- [2] M. F. Cohen, S. E. Chen, J. R. Wallace, and D. P. Greenberg, *A Progressive Refinement Approach to Fast Radiosity Image Generation*, Computer Graphics, Vol.22, No.4,1988.
- [3] M. F. Cohen and D. P. Greenberg, *The hemi-cube: A Radiosity Solution for Complex Environments*, Computer Graphics, Vol.19, No.3, 1985.
- [4] James Arbo, *Graphics Gems II*, Academic Press Inc., pp. 299-302, 1991.
- [5] F. Sillion and C. A. *General Two-Pass Method Integrating Specular and Diffuse Reflection*, Computer Graphics, Vol.23, No.3, 1989.
- [6] D. R. Baum, H. E. Rushmeier and J. M. Winget, *Improving Radiosity Solutions Through the Use of Analytically Determined Form-Factors*, Computer Graphics, Vol.23, No.3, 1989.
- [7] J. R. Wallace, K. A. Elmquist, E. A. Haines, *A Ray Tracing Algorithm for Progressive Radiosity*, Computer Graphics, Vol.23, No.3, 1989.
- [8] T. Nishita, E. Nakamae, *Continuous Tone Representation of Three-Dimensional Objects Taking Account of Shadows and Interreflection*, Computer Graphics, Vol. 19, No. 3, pp. 23-30, 1985.