

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

2 V-6

土井章男 伊藤貴之

日本アイ・ビー・エム株式会社 東京基礎研究所

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。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。

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3角形であると仮定して、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

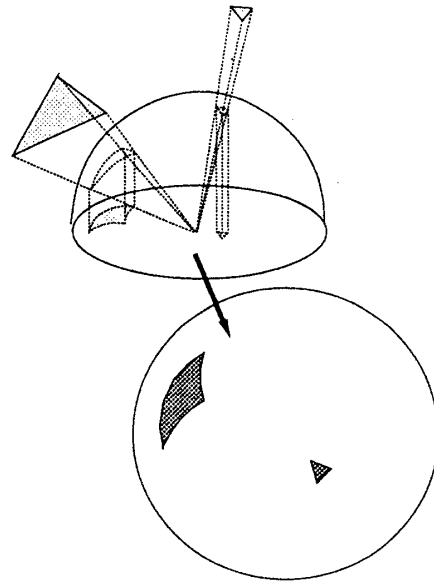


図 1: ヘミスフィアベースを用いたフォームファクタ計算

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、各エレメ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

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室内データに対する 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。ヘミキューブ法に比べて、約2倍以上のスピードアップを実現できた。図2は、右側がヘミキューブ法、左側が本手法で計算した表示結果である。

表 1: CPU Time and Visual Quality

Methods	CPU Time	Visual Quality
Hemi-Cube(100x100)	21.6 sec.	Good
Hemi-Sphere(100x100)	6.7 sec.	Good

* Faces: 18, Patches: 132, Elements: 548, Iterations:

50

表 2: CPU Time and Visual Quality

Methods	CPU Time	Visual Quality
Hemi-Cube(100x100)	351.4 sec.	No Good
Hemi-Sphere(100x100)	135.9 sec.	Good

* Faces: 18, Patches: 328, Elements: 3294, Iterations:

100

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、有利である。また、本方

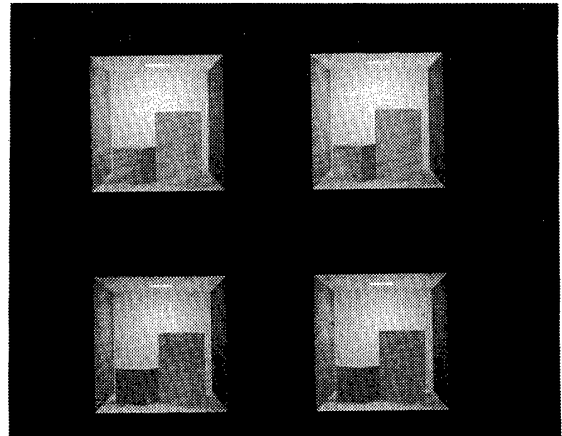


図 2: 測定データ

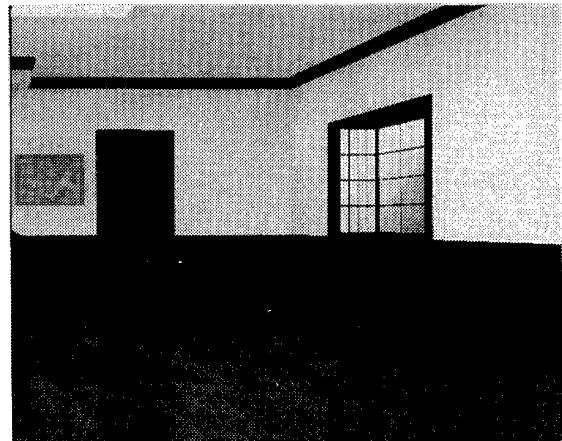


図 3: 表示例

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。

図3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 and D. P. Greenberg, *The hemi-cube: A Radiosity Solution for Complex Environments*, Computer Graphics, Vol.19, No.3, 1985.