

バブル・メッシュ法を用いた ラジオシティ法のための適応的メッシュ生成

伊藤 貴之 山田 敦 嶋田 憲司 青野 雅樹 大淵 竜太郎

日本アイ・ビー・エム (株) 東京基礎研究所
{itot, ayamada, kenjis, aono, ohbuchi}@trl.ibm.co.jp

概要: ラジオシティ法でデータ量の増加を抑えながら写実性を高めるためには、輝度変化を参照した適応的なメッシュ生成の適用が不可欠である。本報告では、1 回目のラジオシティ法の算出結果から、要素サイズを表すスカラ場を定義し、バブル・メッシュ法を用いてメッシュを再生成する適応的メッシュ生成手法を提案する。本手法は、要素形状、パラメータ設定の簡易さ、小さい陰影の欠落の防止、などの点において、従来の適応的メッシュ生成手法の問題点を解決するものである。

キーワード: ラジオシティ法, 適応的メッシュ生成, バブル・メッシュ法

Adaptive Meshing for Radiosity Using Bubble Mesh Method

Takayuki ITOH, Atsushi YAMADA, Kenji SHIMADA,
Masaki AONO, and Ryutarou OHBUCHI

IBM Research, Tokyo Research Laboratory

Abstract: Adaptive meshing methods are indispensable to obtain accurate and efficient radiosity solutions. This paper proposes an adaptive meshing method for radiosity calculation. After the first radiosity calculation, the method generates scalar fields on surfaces of objects, which represent the size of elements. Graded adaptive meshes are generated by using the bubble mesh method and the scalar fields. The method is superior to conventional adaptive meshing methods in the requirements such as: shape of elements, simplicity of parameters, and representation of small shadows.

Key Words: Radiosity Method, Adaptive Meshing, Bubble Mesh Method

1 はじめに

ラジオシティ法において、入力形状領域の分割数およびその形状は、処理時間と写実性の両面に大きな影響を与える、重要なファクタであることが知られている。近年では、有限要素法などの数値解析に広く用いられるメッシュ生成技術が、ラジオシティ法の形状分割技術に適用されることが多い。

ラジオシティ法の処理時間の増加を抑えながら写実性を高める有効な手法として、輝度変化の大きい部位のみを細かくメッシュ分割する、適応的メッシュ生成手法があげられる。最も典型的な適応的メッシュ生成手法として、ラジオシティ法の算出結果から、隣接要素間の輝度差の大きい要素を細かく分割し、ラジオシティ法を再実行する手法 [1] があげられる。しかし、この手法には、以下のような問題点が考えられる。

要素形状の歪み: 要素サイズが不連続であるので、細長い要素が生成されることが多い。細長い要素は、フォーム・ファクタ算出時の算出誤差や、レンダリング時のマッハバンドを生じる要因となることがある。

パラメータ設定の熟練: 従来の適応的メッシュ生成では、輝度差のしきい値をパラメータとして与えることが多い。しかし、輝度差のパラメータは直観的にわかりにくいことが多い。また、わずかなしきい値の設定差で要素数が激変することもあり、その設定は難しい。

小さい陰影の欠落: 1回目のラジオシティ法実行時の要素が大きい時に、陰影の本来の形状をメッシュ生成に反映できないことが多い。逆に、1回目のラジオシティ法実行時に要素を小さく生成すると、メッシュ再生成時に要素数を減らすことが出来ないの、データ量が大きくなることが多い。

本報告では、1回目のラジオシティ法の算出結果から要素サイズ関数を定義し [5]、バブル・メッシュ法 [2] というメッシュ生成手法を用いてメッシュを再生成する、ラジオシティ法のための適応的メッシュ生成手法を提案する。本手法では、連続的な関数で要素サイズを指定するので、細長く歪んだ要素が生成されることが少ない。また、輝度差ではなく要素サイズをパラメータ入力するので、パラメータの設定が比較的わかりやすい。さらに、図 1 に示すように、1回目のラジオシティ法実行時に細かく生成されたメッシュを利用し、メッシュ再生成時に

要素数を減らすことが出来るので、小さい陰影を欠落せずに、要素数を削減することが出来る。

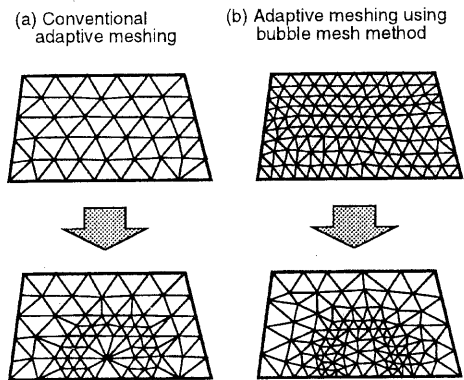


図 1: 従来手法による適応的メッシュ生成と本手法による適応的メッシュ生成

2 バブル・メッシュ法を用いた適応的メッシュ生成

本章では、本手法の基礎となる、バブル・メッシュ法、およびバブル・メッシュ法のための要素サイズ関数について述べる。図 2 は、要素サイズ関数の生成と、バブル・メッシュ法による三角メッシュ生成の、処理の流れを示したものである。

2.1 バブル・メッシュ法

バブル・メッシュ法 [2] は、質量と分子間力を仮想した球状物体 (バブル) を領域内部に最密充填して、メッシュ生成に必要な要素頂点を配置する手法である。バブル・メッシュ法では、図 2 (3)(4) に示すように、球状物体の中心点を要素頂点とみなして、デローニー三角メッシュ法 [3] を用いて連結することで、歪みの少ない三角メッシュを生成する。バブル・メッシュ法は、要素の歪みが少ない他に、適応的メッシュ生成に適していること、輝度不連続線などの幾何的特徴を付加された入力形状にも対応できることなどから、ラジオシティ法に適したメッシュ生成手法であることが示されている [4]。

バブル・メッシュ法では、スカラ場で表現した要素サイズ関数 (Element size control function) を入力するこ

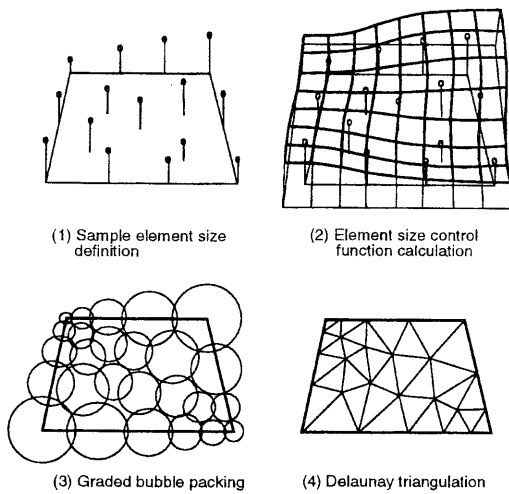


図 2: 要素サイズを指定する関数の生成とバブル・メッシュ法

とで、適応的にメッシュを生成することが出来る。要素サイズ関数を入力するバブル・メッシュ法では、球状物体が反復処理によって移動することにより、球状物体の中心点の座標値を要素サイズ関数に代入し、要素サイズ値を移動位置における球状物体の直径として、隣接する球状物体との分子間力を算出する。

筆者らの実装では、要素サイズ関数のスカラ場を、入力形状領域を覆う直交格子形状の格子点におけるサンプリング値の集合で表現している。図 2 (2) に示す格子状の太線が、直交格子形状の例である。

2.2 バブル・メッシュ法のための要素サイズ関数

バブル・メッシュ法において、歪みの小さいメッシュを生成するためには、数値がなめらかに変化する連続的なスカラ場で要素サイズを指定することが望ましい。この連続的なスカラ場を、入力形状領域を覆う直交格子形状の格子点に対して算出する手法が報告されている [5]。

この手法では、2次元領域内部（または境界上）の複数のサンプリング点における要素サイズ値を入力し（図 2 (1) 参照）、各格子点における要素サイズ値を補間算出する。この手法ではまず、入力形状領域を覆う直交格子形状を xy 平面上に配置し、サンプリング点における要

素サイズ値を、サンプリング点の周囲の格子点に割りつける。この処理の後に、要素サイズ値が割りつけられなかった格子点に、なめらかに数値が変化するように要素サイズ値を算出する。文献 [5] では、なめらかな数値変化が得られるモデルとして、薄板の成形過程における歪みエネルギー値が最小となる定式を適用し、有限要素法を用いて数値を算出している。この時、要素サイズ値が割りつけられた格子点を固定点と仮定し、割りつけられなかった格子点における z 値が算出される。格子点の z 座標値を要素サイズ値とみなすことで、なめらかに数値が変化する要素サイズ関数が生成される。

3 バブルメッシュ法を用いたラジオシテイ法のための適応的メッシュ生成

本章では、前章で示したバブル・メッシュ法、およびバブル・メッシュ法のための要素サイズ関数の生成手法を用いた、適応的メッシュ生成手法を示す。図 3 は、本手法のアルゴリズムを図解したものである。

本手法では、1 回目のラジオシテイ法の算出結果から、まず要素頂点ごとに、隣接要素頂点との単位距離あたりの輝度差を算出する。ただし、本報告では隣接要素頂点を、当該要素頂点と要素辺によって連結される要素頂点と定義する。続いて、輝度差の大きい位置では要素サイズが小さく、輝度差の小さい位置では要素サイズが大きくなるような要素サイズの算出式を、輝度差の関数で定義する。各要素頂点における要素サイズ値を算出した後に、それぞれの入力形状領域に対して、領域を覆う直交格子形状を定義し、要素サイズ値を算出する。この入力形状領域と直交格子形状を入力してバブル・メッシュ法によりメッシュを再生成した後、ラジオシテイ法を再実行する。

隣接要素頂点との輝度差の算出

本手法では、以下の数式を用いて、要素頂点における要素サイズ値を算出する。まず、要素頂点ごとに、隣接要素頂点との輝度差の平均値 dI を、

$$dI = \frac{1}{n} \sum_{i=1}^n \frac{|I_i - I|}{dist_i} \quad (1)$$

により算出する。ここで、 n は隣接要素頂点数、 I および I_i は当該要素頂点および i 番目の隣接要素頂点の輝

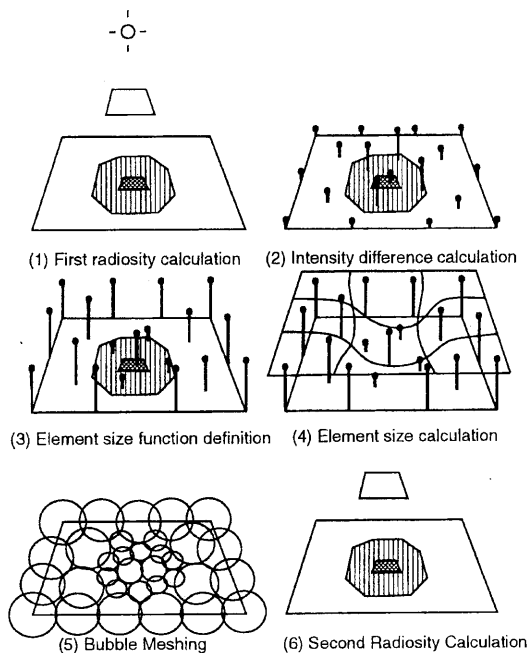


図 3: 本手法のアルゴリズム

度値, $dist_i$ は当該要素頂点と i 番目の隣接要素頂点の距離である。

要素サイズ値の算出式の定義

本手法では要素サイズ値の算出式 $f(dI)$ を, 下記の等式を満たす関数として定義する。

$$size_{min} = f(dI_{min}) \quad (2)$$

$$size_{max} = f(dI_{max}) \quad (3)$$

ただし, dI_{min} および dI_{max} は, すべての要素頂点における dI の最小値および最大値である。また, $size_{min}$ および $size_{max}$ は, 輝度差の平均値 dI がそれぞれ最小, 最大である時の要素サイズ値であり, ユーザーが入力するパラメータである。この時, $size_{min}$ および $size_{max}$ の値の大小関係は,

$$size_{max} < size_{initial} < size_{min} \quad (4)$$

であることが望ましい。ここで, $size_{initial}$ は, 1 回目のラジオシティ法における一般的な要素サイズ値である。

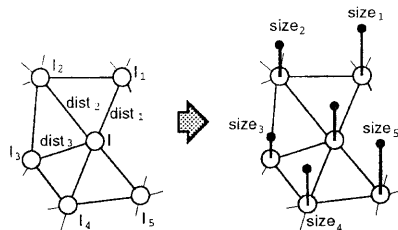


図 4: ノードにおける要素サイズ値の算出

本報告では関数 $f(dI)$ を, 下記の一次関数

$$f(dI) = adI + b \quad (5)$$

として実装した。この時, 係数 a および b は,

$$a = \frac{size_{max} - size_{min}}{dI_{max} - dI_{min}} \quad (6)$$

$$b = \frac{size_{min}dI_{max} - size_{max}dI_{min}}{dI_{max} - dI_{min}} \quad (7)$$

と表される。

関数 $f(dI)$ が定義されたら, 要素頂点ごとに関数 $f(dI)$ を用いて要素サイズ値を算出する。

バブル・メッシュ法を用いたメッシュ再生成

本手法では, すべての要素頂点に対して要素サイズ値を算出した後に, 入力形状領域ごとに, 2.2 節に示した手法で直交格子形状の格子点における要素サイズ値を算出し, バブル・メッシュ法を用いてメッシュを再生成する。筆者らの実装では, まずバブル・メッシュ法によって生成されたメッシュをバッチとし, それを単純な分割手法で細分割してエレメントを生成する。

4 適用例

本手法を適用して, IBM PowerStation RS/6000 (Model 560) で画像を生成した例を示す。

図 8 は, 3 枚の長方形が浮遊する情景を, メッシュ分割した結果である。図 5 は, 図 8 に示すメッシュを用いて, ラジオシティ法を実行して生成した画像である。

図 9 および図 10 は, 図 5 に示すラジオシティ法の算出結果を用いて, それぞれ従来の適応的メッシュ生成手法, および本手法により再生成したメッシュである。陰影の境界周辺や, 光源からの直射光によるハイライト

周辺の要素が細かく生成され、それ以外の部位の要素が大きく生成されている。

図 6 および図 7 は、それぞれ図 9 および図 10 に示すメッシュを用いてラジオシティ法を再実行して生成した画像である。図 5 に示す画像と比較して、床面の陰影が鮮明に表現されていることがわかる。

表 1 は、図 8、図 9、および図 10 に示すメッシュの要素数である。この結果からわかるように、従来の適応的メッシュ生成手法では要素数が大きく増加しているのに対し、本手法はメッシュ再生成によって要素数の削減を実現している。この差は、2 回目のラジオシティ法、およびラジオシティ法後のレンダリングの処理時間に大きく影響する。

表 1: メッシュ生成結果における要素数と要素頂点数

	要素数
1 回目のメッシュ	1976
従来手法による 2 回目のメッシュ	3205
本手法による 2 回目のメッシュ	1396

表 2 は、図 7 に示す画像の生成における、メッシュ生成およびラジオシティ法に費やした処理時間の一覧である。

表 2: 適応的メッシュ分割とラジオシティ法の処理時間

処理	処理時間 (秒)
1 回目のメッシュ生成	31.51
1 回目のラジオシティ法	277.31
要素サイズ関数の定義	171.75
2 回目のメッシュ生成	42.23
2 回目のラジオシティ法	225.54

メッシュ生成部の処理時間と比較して、要素サイズ関数の定義部における有限要素法に多くの処理時間を要している。本報告ではすべての入力ポリゴンに本手法を適用してメッシュを再生成しているが、輝度変化の大きいポリゴンのみを自動抽出し、直交格子形状の格子数を要素の大きさから自動的に決定することで、要素サイズ関数の定義部の処理量を大幅に低減できると思われる。

また、本報告では 1 回目のメッシュ生成にもバブル・メッシュ法を用いているが、入力ポリゴンが長方形等の

単純な形状のみで構成されている場合には、1 回目の一様なメッシュ生成にバブル・メッシュ法を用いる必要はない。むしろ単純なメッシュ生成手法を用いる方が、処理時間の点では有利であると思われる。

5 むすび

本報告では、1 回目のラジオシティ法の算出結果から要素サイズ関数を定義し、バブル・メッシュ法を用いてメッシュを再生成する、ラジオシティ法のための適応的メッシュ生成手法を提案した。本手法で示した適応的メッシュは、ラジオシティ法だけでなく、有限要素法を用いた計算力学や、形状モデリングなどにも適用が可能である。

参考文献

- [1] Baum D. R., Mann S., Smith K. P., and Winget J. M., *Making Radiosity Usable: Automatic Pre-processing and Meshing Techniques for the Generation of Accurate Radiosity Solutions*, Computer Graphics, Vol. 25, No. 4, pp. 51-60, 1991.
- [2] 嶋田憲司, 物理モデルによる自動メッシュ分割, シミュレーション, Vol. 12, No. 1, pp. 11-19, 1993.
- [3] Lo S. H., *Delaunay Triangulation of Non-Convex Planar Domains*, International Journal for Numerical Methods in Engineering, Vol. 28, pp. 2695-2707, 1989.
- [4] 嶋田, 伊藤, 土井, ラジオシティ法の写実性と処理速度向上のための自動三角メッシュ分割法, 第 10 回 NICOGRAPH 論文コンテスト論文集, pp. 44-55, 1994.
- [5] 山田, 嶋田, 伊藤, バブル・メッシュ分割法のための要素サイズの制御方法, 情報処理学会グラフィクスと CAD 研究報告, 96-CG-80, pp. 1-6, 1996.

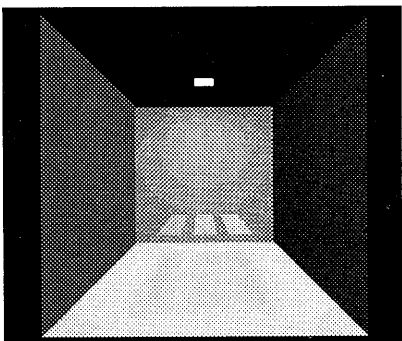


図 5: 1 回目のラジオシティ法算出結果による画像

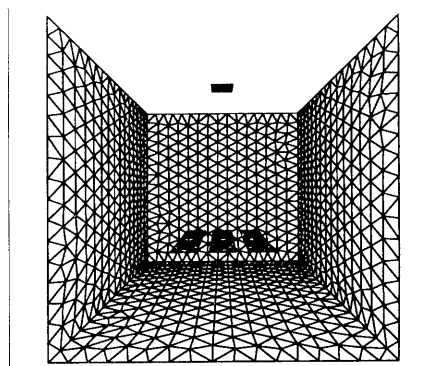


図 8: 1 回目のメッシュ生成

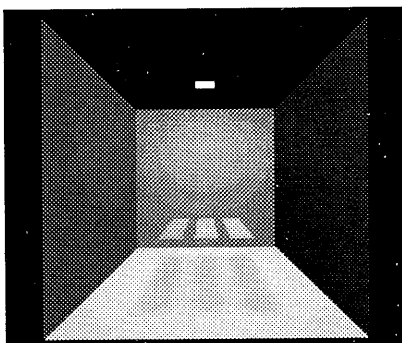


図 6: 従来手法の適応的メッシュ生成後の画像

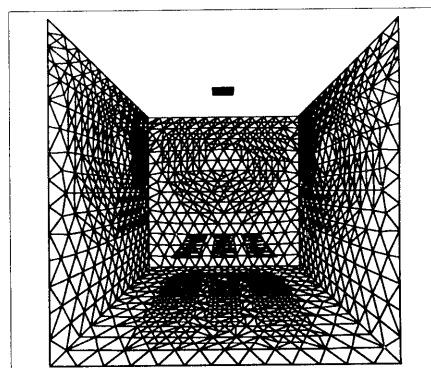


図 9: 従来手法の適応的メッシュ生成

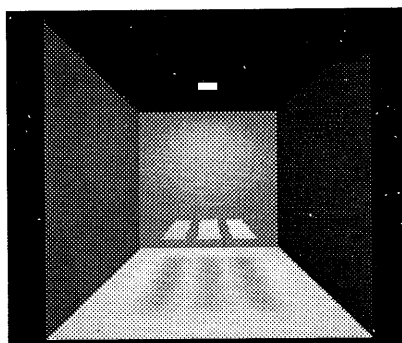


図 7: 本手法の適応的メッシュ生成後の画像

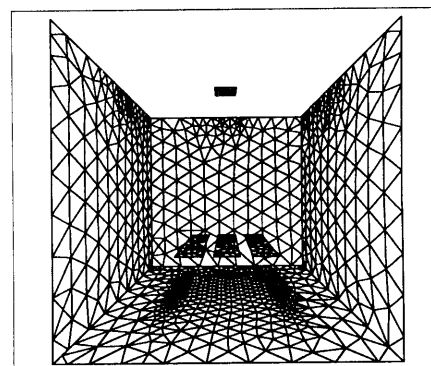


図 10: 本手法の適応的メッシュ生成