

コンピュータ・グラフィックスのための ポリゴン生成技術に関する研究

1997年2月

伊藤 貴之

Takayuki ITOH

目次

1	序論	5
1.1	序言	6
1.2	本論文の背景	8
1.3	本論文の概要	11
1.4	本論文の位置付け	15
1.5	本論文の構成	15
2	ラジオシティ法と2次元メッシュ生成手法	19
2.1	はじめに	20
2.2	ラジオシティ法	20
2.2.1	ラジオシティ法の概要	20
2.2.2	プログレッシブ・レファインメント・ラジオシティ法	23
2.2.3	ラジオシティ法に用いるデータ構造	26
2.2.4	ラジオシティ法におけるパッチ/エレメント生成技術の要件	29
2.3	2次元メッシュ生成手法	30
2.3.1	2次元メッシュ生成手法の概観	30
2.3.2	デローニー三角メッシュ生成法	32
2.3.3	バブル・メッシュ法	32
2.4	適応的メッシュ生成手法	34
2.4.1	適応的メッシュ生成手法の概観	35
2.4.2	適応的メッシュ生成手法を用いたラジオシティ法	36
2.4.3	輝度不連続線を用いたラジオシティ法	37
2.5	むすび	40

3	簡略な輝度不連続線法	43
3.1	はじめに	44
3.2	ラジオシティ法のための簡略な輝度不連続線法の概要	45
3.3	ラジオシティ法のための簡略な輝度不連続線法の各処理	49
3.3.1	影多面体の生成	50
3.3.2	受光面のデータ構造	51
3.3.3	影多面体との交点計算による輝度不連続線の生成	54
3.3.4	ループ処理による受光面の領域分割	55
3.3.5	照射状態の判定	57
3.3.6	受光面の分割とメッシュ生成	57
3.4	実行例	60
3.5	むすび	63
4	輝度分布参照型バブル・メッシュ法	73
4.1	はじめに	74
4.2	ラジオシティ法のための輝度分布参照型バブル・メッシュ法の概要	75
4.3	ラジオシティ法のための輝度分布参照型バブル・メッシュ法の各処理	79
4.3.1	輝度変化から要素サイズ値への変換式	79
4.3.2	要素サイズのサンプリング値の算出	80
4.3.3	要素サイズ関数を参照するバブル・メッシュ法	83
4.4	実行例	86
4.5	むすび	88
5	3次元数値解析結果のグラフィックス表示	93
5.1	はじめに	94
5.2	ボリューム・データのグラフィックス表示手法	94
5.2.1	ボリューム・データのデータ構造	94
5.2.2	ボリューム・データの種類	95
5.2.3	ボリューム・データのグラフィックス表示手法	98
5.3	等値面生成	99
5.3.1	等値面生成	99

5.3.2	等値面生成手法の問題点	102
5.4	等値面生成の高速化手法	103
5.4.1	等値面生成の高速化手法	104
5.4.2	自己増殖的な等値面生成手法	107
5.5	むすび	109
6	極点グラフを用いた等値面高速生成手法	111
6.1	はじめに	112
6.2	極点グラフを用いた等値面生成の概要	113
6.3	極点グラフを用いた等値面生成手法の各処理	117
6.3.1	極大点・極小点の抽出	117
6.3.2	極点グラフの生成	118
6.3.3	境界面要素リストの生成	122
6.3.4	自己増殖的手法を用いた等値面生成	123
6.4	極点グラフを用いた等値面生成手法の処理量	125
6.5	実行例	126
6.6	むすび	131
7	細線化手法を用いた等値面高速生成手法	135
7.1	はじめに	136
7.2	ポリウムデータと等値面の位相的性質	137
7.3	細線化手法	139
7.4	細線化手法を用いた等値面生成手法の概要	140
7.5	細線化手法を用いた等値面生成手法の各処理	143
7.5.1	細線化手法のためのデータ構造の整備	144
7.5.2	細線化手法による要素の除去	145
7.5.3	空洞周辺の非除去要素層の除去	148
7.5.4	非除去要素の登録	150
7.5.5	自己増殖的手法を用いた等値面生成	150
7.6	細線化手法を用いた等値面生成手法の処理量	151
7.7	実行例	151

7.8	ボリューム・レンダリングへの応用	156
7.8.1	レイ・キャスティングを用いたボリューム・レンダリング	156
7.8.2	ポリゴンの半透明表示手法を用いた高速ボリューム・レンダリング	158
7.8.3	ボリューム・レンダリングへの応用	159
7.9	むすび	161
8	結論	173
	参考文献	179

第 1 章

序論

1.1 序言

近年の計算機能力の発達にともない，コンピュータ・グラフィックス (Computer Graphics) を用いた画像生成技術の研究開発が，ハードウェア・ソフトウェアの両面において活発に行なわれており，その用途は現在も拡大し続けている．初期のコンピュータ・グラフィックスの研究成果は，一枚の静止画をバッチ処理で生成することを目的としたものが多かったが，近年では対話的な操作にしたがって連続的に画像を生成することを目的とするものも多くなっている．

コンピュータ・グラフィックスのアルゴリズムは，入力形状である情景や物体を数式化するモデリング (Modeling) と，情景や物体をスクリーンに投影し画像を生成するレンダリング (Rendering) に大別することが出来る．現在までに報告されている，コンピュータ・グラフィックスのレンダリングアルゴリズムは，

- レイ・トレーシング法 (Ray tracing) [66] やラジオシティ法 (Radiosity) [22] に代表される，情景の写実的な再現を目的とした手法．
- ボリューム・ビジュアライゼーション (Volume Visualization) に代表される，測定結果や数値解析結果の仮想的なグラフィックス表示 (可視化) を目的とした手法．

などに大別することが出来る．このうちレイ・トレーシング法は，存在領域法，空間分割法，画素選択法などの効率化アルゴリズムの研究，および並列処理やハードウェア化の技術が，すでに活発に議論されてきた．一方，比較的新しい手法であるラジオシティ法やボリューム・ビジュアライゼーションの分野においては，現在も効率化のための手法が議論されている．

一方，近年では，三角形・四角形などの凸多角形 (ポリゴン (Polygon) と呼ばれることが多い) の集合を入力形状として，高速にレンダリングを実行するハードウェアおよびソフトウェア技術が，非常に活発に発表されている．対話的な操作と連続的な画像生成を目的とする近年のコンピュータ・グラフィックスの研究において，これらのレンダリング技術を有効に利用することは不可欠である．そのためには，

高速レンダリングに必要なポリゴンの集合を，効率的に生成する手法の研究が重要な課題であり，現在も非常に活発な議論が進んでいる．

ラジオシティ法 (Radiosity) は、情景中に仮想した光エネルギーの平衡状態を解き、情景の各部位における輝度を算出することにより、写実的に陰影やハイライト等の輝度変化を表現する画像生成手法である。ラジオシティ法は、建築物の設計や、歴史的建造物の再現などのための景観シミュレーションの目的で、実用化が進んでいる。一般にラジオシティ法では、メッシュ生成 (Meshing) という技術を応用して、情景中の物体をポリゴンの小片 (要素) に分割し、要素ごとに光エネルギー値を算出することで、離散的に輝度分布を算出する。ラジオシティ法の結果は、輝度値を与えられた要素の集合として出力されるので、前述のグラフィックスハードウェアやソフトウェアを用いて、視点などを対話的に操作しながら、連続的に画像を表示することが出来る。ここで、画像表示の処理時間は、入力するポリゴン数にほぼ比例するので、視点操作による画像表示の対話性は、ラジオシティ法における要素数に依存することがいえる。そこでラジオシティ法では、要素数を増加せずに写実的な画像を生成するために、さまざまなメッシュ生成手法が議論されている。

ボリューム・ビジュアリゼーション (Volume visualization) は、3次元空間中のスカラー値を与えられた離散点の集合で表現されるボリューム・データ (Volume Data) を、コンピュータ・グラフィックスを用いて仮想的に表示 (可視化) する技術である。ボリューム・ビジュアリゼーションは、有限要素法などの3次元数値解析結果や、CT スキャンなどの医療技術を代表とする3次元測定結果の表示などの目的で、実用化が進んでいる。ボリューム・ビジュアリゼーションの代表的な技術に、等値面 (Isosurface) [43] がある。等値面の本来の定義は、スカラー場 $S(x, y, z)$ をもつ3次元空間中において、式 $S(x, y, z) - C = 0$ を満たす点の集合であり、一般的には曲面を形成する。等値面は通常、ボリューム・データを構成する四面体や六面体の小片 (3次元要素) との交差部分をポリゴンで近似生成し、そのポリゴンの集合として表現される。よって、前述のグラフィックスハードウェアやソフトウェアを用いることで、視点や数値 C などの情報を対話的に操作しながら、連続的に等値面を生成し、高速に画像表示することが出来る。ここで、数値 C の操作にともなう等値面の連続表示の操作性は、等値面の生成速度に大きく依存する。そのため近年では、等値面の高速生成手法が活発に議論されている。

本論文は、コンピュータ・グラフィックスを用いた対話的な連続画像表示技術に重要な役割を果たす、ポリゴン生成技術に関する研究の成果をまとめたものであり、主にラジオシティ法や等値面生成に関して報告されている。

ラジオシティ法については，ポリゴン数の増加を抑えながら，陰影やハイライトなどの輝度変化を写実的に表現するための，適応的なポリゴン生成手法を提案する．

等値面生成については，等値面と交差しなく多くの3次元要素との交差判定処理を省略することにより，等値面生成処理を高速化する手法を提案する．

本章では，まずラジオシティ法および等値面生成に関する従来研究成果を概観する．続いて，本論文で報告するラジオシティ法のためのポリゴン生成手法，および等値面の高速生成手法に関する概要を述べる．さらに，本論文の位置付け，および構成を示す．

1.2 本論文の背景

ラジオシティ法は，情景中に仮想した光エネルギーの平衡状態の連立方程式を解き，情景の各部位における輝度分布を算出する手法である．ラジオシティ法では，入力形状はポリゴンの小片(要素)の集合に分割され，要素ごとに光エネルギー値が算出される．ここで，ラジオシティ法の処理時間や写実性，データ量は，入力形状の分割技術に大きく依存することが知られている．一般に，入力形状を細かく分割した方が，輝度値を正確に算出できるので写実性が向上するが，処理時間やデータ量は増大する．また，極端に形状の歪んだ分割は，輝度値の算出誤差を招きやすいので，避けられるべきである．

ラジオシティ法における入力形状の分割技術の要件は，有限要素法などの数値解析の分野で研究開発が進んでいる，メッシュ生成手法の要件に類似している．そこでラジオシティ法では，入力形状の分割にメッシュ生成手法が適用されていることが多い．特に，デローニー三角メッシュ生成法 [42]，およびバブル・メッシュ法 [53] は，複雑な入力形状領域においても要素形状の歪みが少ないメッシュを生成する点で，優れた手法として知られている．

また，数値解析の分野では，数値変化に応じて要素の大きさや形状等を制御することにより，短い処理時間で比較的正確な解を得る，適応的メッシュ生成手法 (Adaptive Meshing) が多用されている．適応的メッシュ生成手法は，数値解析前に得られる数値的特徴を参照してメッシュを生成する前処理型適応的メッシュ生成と，数値解析結果を参照してメッシュを修正した後に数値解析を再実行する後処理型適応的メッシュ生成に大別される．

ラジオシティ法で最も多く用いられている，前処理型の適応的メッシュ生成手法の代表的な手法として，輝度不連続線法 (Discontinuity Meshing) [40] が提案されている．

輝度不連続線法は，光源から照射される直射光による輝度値が不連続となる境界線によって，物体を分割してからメッシュ生成を実行することにより，陰影の境界の両側における輝度値の不連続性を表現する手法である．輝度不連続線法では通常，光源や物体の形状を多角形・多面体に限定する．輝度不連続線法ではラジオシティ法を実行する前に，光源の辺と物体の頂点を結んでできる平面，および光源の頂点と物体の辺を結んでできる平面を生成し，その平面との交線（輝度不連続線）で他の物体を分割し，その後輝度不連続線をまたがないように要素を生成する．輝度不連続線法は，鮮明な陰影を写實的に表現できる特徴がある．しかしその反面，

- 大量の輝度不連続線が生成されたときに，輝度不連続線の交点計算に多くの計算量を要する．
- 多数の輝度不連続線が複雑に交差する部位では，非常に小さい要素や細長い要素が生成され，ラジオシティ法の精度が悪くなる可能性が高くなる．

などの問題点があり，実用性に欠ける．

ラジオシティ法で最も多く用いられている，後処理型の適応的メッシュ生成手法として，1回目のラジオシティ法の実行によって各要素の輝度値を算出した後に，輝度変化の大きい部位の要素をさらに細かく分割してラジオシティ法を再実行する，適応的再分割法（Adaptive Subdivision） [4] が提案されている．適応的再分割法は，実装が単純であり，再分割そのものの処理量が小さいなどの特徴がある．しかしその反面，

- 処理の閾値を隣接頂点の輝度差で与えるので，利用者にとって直観的でなく，熟練を要する．
- 1回目のラジオシティ法に用いる要素が小さいと，2回目のラジオシティ法における要素数が非常に多くなる可能性が高くなる．
- 逆に1回目のラジオシティ法に用いる要素が大きいと，要素数の増加を抑えることができる反面，小さな陰影の表現を欠落する可能性が高くなる．

などの問題点がある．そのため，適応的再分割法を用いて，要素数の増加を抑えながら，細かい輝度変化を表現するには，熟練を要する．

ボリウム・ビジュアリゼーションの技術は，ボリウム・データを直接入力して画像を生成するダイレクト・アプローチ (Direct Approach) と，ボリウム・データから線や面を抽出して画像を生成するインダイレクト・アプローチ (Indirect Approach) に大別される．前者の代表的な手法に，ボリウム・レンダリング (Volume Rendering) [38] [39] があげられる．後者の代表的な手法に，等値面 (Isosurface) 生成 [43] [71] があげられる．ボリウム・レンダリングは計算量が大きく，しかも初期の手法ではグラフィックスハードウェアやソフトウェアによる高速表示を実現することが困難であった．そのため，ボリウム・データの対話的なグラフィックス表示には，等値面が用いられることが多かった．

等値面は，ボリウム・データ中の3次元要素との交差部分をポリゴンで近似生成し，そのポリゴンの集合として表現されることが多い．等値面を生成する最も単純なアルゴリズムは，ボリウム・データ中のすべての3次元要素について等値面との交差判定を実行し，交差すると判定された要素に対して，その交差部分をポリゴンを近似生成するアルゴリズムである．しかし，この単純なアルゴリズムには，

- 等値面と交差する要素はボリウムデータ中の一部であることが多く，他の多くの非交差要素との交差判定に，不必要な多くの処理時間を費やす．

という問題点があった．そこで近年では，等値面と交差しない要素との交差判定を減らす高速化手法が多く報告されている [17] [21] [25] [41] [57] [67]．しかし，これらの手法は，

- 前処理量が $O(n)$ より大きい．
- 等値面生成処理量が $O(n)$ 程度であるので，大規模なボリウムにおいて処理量が大きくなる．
- 要素が直交座標軸方向に整列していない不規則なボリウム・データへの実装が難しい

のいずれかの問題点を持っており，このすべての問題点を解決した手法はまだ報告されていない．

ボリウム・レンダリングは，ボリウム・データの対象となる3次元空間中に，不均一な半透明物体を仮想し，その色彩や濃度をボリウム・データのスカラ値の関数で与

えることで、仮想的にボリューム・データのスカラ場をグラフィックス表示する技術である。一般的には、レイ・キャスティング (Ray Casting) という手法 [38] を用い、視点から伸ばした半直線上の離散点におけるスカラ値を、色彩に変換して累算する。ボリューム・レンダリングの高速化手法の中で、レイ・キャスティングと同等な画質を実現する手法として、視点からの等距離面を複数生成し、この等距離面を半透明な面として重ね描きする手法が提案されている [35]。近年では、グラフィックスハードウェアやソフトウェアを用いて、ポリゴンの半透明な重ね描きを用いた高速表示が可能になっている。そのため、視点からの等距離面をポリゴンの集合として高速生成することが出来れば、従来は困難であると思われていた、レイ・キャスティングと同等な画像を、対話的な速度で連続生成することが可能になると思われる。

1.3 本論文の概要

本論文の目的は、ラジオシティ法やボリューム・ビジュアライゼーションなどの比較的新しいレンダリング技術を用いて、対話的な操作で連続的に画像を表示するための、ポリゴン生成手法を構築することである。本論文の前半では、ラジオシティ法において、要素数の増加を抑えながら写実性の高い結果を算出するためのメッシュ生成手法を提案する。また、本論文の後半では、等値面生成の高速化手法を提案する。

本論文の前半では、まずラジオシティ法の概要を紹介し、その入力形状の分割技術がラジオシティ法の写実性、処理速度、ラジオシティ法の算出結果の画像生成の速度、などの面において重要であることを示す。続いて、ラジオシティ法に適用されている主要な2次元メッシュ生成技術について紹介する。特に、複雑な入力形状領域においても要素形状の歪みが少ないメッシュを生成する、デローニー三角メッシュ生成法 [42]、およびバブル・メッシュ法 [53] を詳しく紹介する。続いて、数値解析結果の数値分布を考慮することにより、要素数の増加を抑えながら解の精度を向上する、適応的メッシュ生成手法を紹介する。

ラジオシティ法に適用されている前処理型適応的メッシュ生成手法の代表例に、輝度不連続線法 [40] があげられる。前節で述べた通り、輝度不連続線法では、多数の輝度不連続線が複雑に交差する部位において、輝度不連続線どうしの交点計算に大きな計算量を要するとともに、非常に小さい要素や細長い要素が生成される可能性が高くなる。

ここで、多数の輝度不連続線が集中する領域は、光源からの直射光の一部のみが受光面に到達する、半影領域 (penumbra) である。一般的には半影領域は他の領域よりも輝度変化が複雑であり、他の領域よりも細かく要素が分割されていることが望ましい。本論文では、照射領域、半影領域、本影領域の境界線のみを輝度不連続線として生成する、簡略な輝度不連続線法 (Simplified Discontinuity Meshing) [27] を提案する。本手法では、半影領域や本影領域を半無限な多面体で表現する影多面体 [46] と各物体の交線を求め、交線に囲まれる各領域と影多面体との内外判定を行なう。この内外判定結果を参照して、輝度変化の大きい半影領域のみにおいて要素を細かく生成し、他の領域では要素を大きく生成する。本手法は輝度不連続線の生成数を減少するので、輝度不連続線どうしの交点計算の処理量を低減すると同時に、極端に小さい要素や細長い要素の生成を抑えることができる。また、本手法では、半影領域以外の領域で大きく要素を生成することができるので、要素数の増加を抑えることが可能である。

ラジオシティ法に適用されている後処理型適応的メッシュ生成手法の代表例に、適応的再分割法 [4] があげられる。前節で述べた通り、適応的再分割法では、要素数が非常に増加することが多い。要素数の増加を抑えるためには、1 回目のラジオシティ法における要素数を減少する必要があるが、この場合には要素の大きさよりも小さな陰影を表現することが難しい。

本論文では、この問題を解決するために、輝度変化の大きい部位では小さく、輝度変化の小さい部位では大きく要素を生成する、輝度分布参照型バブル・メッシュ法 (Intensity-based Bubbue Mesh) [28] を提案する。輝度分布参照型バブル・メッシュ法を用いることにより、細かく生成された要素を用いて 1 回目のラジオシティ法を実行し、その算出結果から輝度変化の小さい部位の要素を減少して 2 回目のラジオシティ法を実行することが可能である。本手法では 1 回目のラジオシティ法の実行後、それぞれの入力形状領域において、隣接要素頂点間の単位あたり輝度差の最大値 D を求める。続いて、この単位あたり輝度差の最大値を用いて、各要素頂点の位置における要素サイズ (要素の辺の長さ) を算出する関数 $f(D)$ を決定し、 $f(D)$ を用いて各要素頂点の位置における要素サイズ値を算出する。ここで、関数 $f(D)$ には、輝度変化の大きい部位では小さく、輝度変化の小さい部位では大きく算出されるような関数が用いられる。続いて、入力形状領域を覆う直交格子形状を定義し、要素サイズ値が算出された要素頂点に近隣する格子点に、その要素サイズ値をわりつける。さらに、要素サイズ値がわりつけられなかった格子点にお

る要素サイズ値を，要素サイズ値がわりつけられた格子点を用いて補間によって求める．ここで，歪みの少ない要素を生成するためには，要素サイズ値が位置変化に対してなめらかに変化することが望ましい．本手法では，金属等の薄板の成形過程の歪みエネルギーが最小となる状態を算出する有限要素法の定式を適用して，要素サイズ値がわりつけられなかった格子点における要素サイズ値を算出する．続いて，この直交格子形状の格子点における要素サイズ値を参照しながら，バブル・メッシュ法 [53] によってメッシュを再生成する．このメッシュ再生成をすべての入力形状領域について実行した後に，2 回目のラジオシティ法を実行する．

本論文の後半では，まずボリューム・ビジュアライゼーションの概念と，ボリューム・データの基本的なデータ構造を紹介する．続いて，等値面生成手法，および等値面生成の高速化手法と，その問題点について紹介する．ボリューム・データは，対象となる 3 次元空間を分割してできる，多数の四面体や六面体の小片 (3 次元要素) によって構成される．等値面は，各々の 3 次元要素との交差部分をポリゴンで近似生成される [43]．一般的に，1 枚の等値面と交差する 3 次元要素は，ボリューム・データ中のごく一部分であることが多いので，等値面と交差しない 3 次元要素をあらかじめ処理から除外する高速化手法が多く報告されている．

本論文では，等値面生成の高速化手法として，まず 極点グラフ法 [29] を提案する．閉曲面となる等値面に囲まれた領域の内部および外部には，それぞれ少なくとも 1 点以上の極大点または極小点が存在する．これらの極大点や極小点を連結するグラフ (極点グラフ) の 1 本以上の辺は，閉曲面となる等値面と交差する．一方，閉曲面でない等値面は，ボリュームの境界面と交差する．本手法では，極点グラフの辺が通過する 3 次元要素，およびボリュームの境界面に接する 3 次元要素と等値面の交差判定により，等値面と交差する要素を抽出し，抽出された交差要素を出発点として隣接する交差要素を再帰的に処理することで，等値面を生成する．本手法では，等値面と交差しない多くの 3 次元要素との交差判定を省略するので，効率的に等値面を生成する．なお，本手法では，ボリュームの境界面に接する 3 次元要素を，要素頂点におけるスカラー値でソートすることで，その多くを交差判定処理から省略している．

本論文では続いて，ボリュームの細線化手法 (Volume Thinning) [30] を用いた等値面生成手法を提案する．非構造ボリュームが貫通穴をもつ場合には，閉曲面でない等値面の非連結なすべての部分と極点グラフの交差を保証しない場合がある．これは，極点グ

ラフがボリュームの貫通穴のトポロジーを必ずしも保持しないために起こることである。逆にいえば、極点グラフがボリュームの貫通穴のトポロジーを保持すれば、極点グラフは閉曲面でない等値面の非連結なすべての部分と交差するので、ボリュームの境界面に接する3次元要素との交差判定を行なわなくても等値面との交差要素が抽出され、極点グラフ法よりも高速に等値面を生成することができる。本手法では、ボリュームの貫通穴のトポロジーを保持する極点グラフを生成するために、画像認識の分野で用いられている細線化手法に類似した手法（ボリュームの細線化手法）を適用する。画像認識における細線化手法では、画像の特定領域を画素単位で削り、領域の穴などのトポロジーを保持しながら、領域の骨格に相当する画素の線列を生成する。ボリュームの細線化手法では、まず極大点や極小点を抽出し、抽出された各点に接する3次元要素をマークする。続いて、画像認識における細線化手法が領域内の画素を除去するのと同様に、マークされなかった多くの3次元要素を除去し、3次元要素の線列群を生成する。本手法では、マークされた3次元要素どうしの連結性と、ボリュームのトポロジーを保持する範囲内で、3次元要素を除去するものとする。続いて、この線列群を構成する3次元要素と等値面との交差判定を実行し、抽出された交差要素に隣接する交差要素を再帰的に処理することで、等値面を生成する。本手法では、ボリュームの境界面に接する3次元要素との交差判定を必要としないので、極点グラフ法よりもさらに高速に等値面を生成することが出来る。なお、本手法では、線列を構成する3次元要素を、要素頂点におけるスカラ値で階級化することで、その多くを交差判定処理から省略している。

ボリューム・レンダリングの代表的な手法に、レイ・キャスティング [38] があげられる。近年では、半透明な視点からの等距離面を重ね描きすることで、レイ・キャスティングと同等な画像を高速生成する手法 [35] が提案されている。文献 [35] では、この手法は処理量のかなり多くを等距離面の生成に費やされており、この高速化が重要な課題であるとしている。ここで、細線化手法を応用した等値面生成手法は、視点からの等距離面の生成にも適用することが出来る。よって、本手法の適用によって、ボリューム・レンダリングを高速化することも可能である。

1.4 本論文の位置付け

ラジオシティ法を用いた画像生成手法は，例えばインテリア設計，歴史的建造物の再現，などの景観シミュレーションの目的で，実用が進んでいる．近年では，ラジオシティ法の算出結果から一枚の静止画像を生成するにとどまらず，視点位置を対話的に操作して画像を連続表示し，あたかも利用者がその情景を移動している (Walkthrough) ような感覚を与えるソフトウェアが活発に開発されている．ラジオシティ法の画像の写実性は，一般的には入力形状を細かく分割し，かつ歪みが生じないように分割した方が向上する．一方，ラジオシティ法の算出結果を用いた画像生成の速度は，一般的には入力形状の分割数を小さくした方が向上する．本論文では，入力形状の分割数の増加を抑えながら，輝度変化の大きい部位のみにおいて形状を細かく分割することで，写実性の高い画像生成を実現する手法を提案している．この研究成果は，景観シミュレーションにおける対話性と写実性の両者をともに向上した点において，大きな意義があるものと思われる．

一方，有限要素法などの数値解析技術は，自然現象の理解や製造物設計などの目的で実用が進むとともに，近年ではスーパーコンピュータや並列計算機などの大規模な計算機資源を有効に用いた，大規模な数値解析が行なわれることが多くなっている．数値解析の大規模化につれて，その解析結果の理解のためのグラフィックス表示技術の高速化の必要性も高くなっている．本論文では，等値面と交差しない3次元要素を処理から除外することによる等値面生成の高速化手法を提案している．この研究成果は，大規模な数値解析結果においても対話的なグラフィックス表示を容易にし，数値解析結果の理解を容易にした点において，大きな意義があるものと思われる．

1.5 本論文の構成

本節では，本論文の構成を示す．

第2章では，まずラジオシティ法の概要を紹介し，ラジオシティ法における入力形状の分割技術の要件を示す．続いて，有限要素法等の数値解析で用いられている2次元メッシュ生成技術について述べる．特に，本論文で用いるデローニー三角メッシュ生成法，およびバブル・メッシュ法について，詳しく紹介する．続いて，数値分布を考慮した適応的メッシュ生成手法，およびラジオシティ法に適用されている適応的メッシュ生成手法について紹介する．特に，適応的再分割法，輝度不連続線について詳しく紹介し，その問題点

を列挙する．

第 3 章では，簡略な輝度不連続線法を提案する．まず，照射領域，半影領域，本影領域の境界線のみを輝度不連続線として生成することによって，輝度不連続線の生成数を減少し，かつ非常に小さい要素や細長い要素の生成を低減することができることを示す．続いて，簡略な輝度不連続線法の各処理について，詳しく示す．本章では，本手法で用いる影多面体の生成手法，影多面体と受光面の交線を生成する手法，輝度不連続線および受光面の境界によって囲まれる各領域を囲む辺の集合 (ループ) のデータを構築する手法，輝度不連続線および境界線に囲まれる各領域においてメッシュを生成する手法，の順に述べる．最後に，本手法によって生成された画像を示し，本手法の有用性を確認する．

第 4 章では，輝度分布参照型バブル・メッシュ法を提案する．まず，本手法の概要を述べる．続いて，本手法の各処理について，詳しく述べる．本章では，輝度変化を算出する関数 $f(D)$ の導出手法，金属等の薄板の成形過程の歪みエネルギーが最小となる状態を算出する有限要素法の定式を適用して要素サイズ値を算出する手法，要素サイズ値を参照するバブル・メッシュ法の順に述べる．最後に，本手法によって生成された画像を示し，本手法の有用性を確認する．

第 5 章では，ボリューム・ビジュアライゼーションの概念と，基本的なデータ構造を紹介する．続いて，ボリューム・データのグラフィックス表示手法は，ボリューム・データを直接参照するダイレクト・アプローチと，ボリューム・データの一部を点，線，面などの形状で抽出するインダイレクト・アプローチの 2 種類に分類されることを示す．続いて，インダイレクト・アプローチの代表的な手法である，等値面生成手法について紹介する．さらに，等値面と交差する 3 次元要素はボリューム・データ中の一部であることが多く，他の多くの非交差要素との交差判定に不必要な多くの処理時間を要するという問題点について述べる．続いて，従来報告されている等値面生成の高速化手法を列挙し，その問題点について述べる．また，第 6 章，および第 7 章で提案する手法の前提となる，自己増殖的な等値面生成手法について紹介する．

第 6 章では，極点グラフ法を提案する．まず，等値面を閉曲面と境界つき曲面に分類し，境界つき曲面はボリュームの境界面に接することを示す．続いて，閉曲面となる等値面が，極点グラフの辺が交差する最低 1 個の 3 次元要素と交差することを示す．よって，任意の等値面との交差要素は，極点グラフおよびボリュームの境界面から抽出される．本手法では，等値面との交差要素を抽出し，その交差要素を出発点として隣接交差要素を

順に処理することにより，多くの非交差要素との処理を省略して等値面を生成する．続いて，極点グラフ法の各処理について詳しく示す．本章では，極大点および極小点の抽出手法，極大点および極小点を連結するグラフの辺を生成する手法，ボリュームの境界面に接する要素の抽出手法，等値面の生成手法の順に示す．最後に，本手法による等値面生成の処理速度を示し，本手法の有用性を示す．

第 7 章では，ボリュームの細線化手法を用いた等値面生成手法を提案する．まず，極点グラフ法において，ボリュームの境界面に接する 3 次元要素との交差判定の必要性について述べる．続いて，画像認識の分野で用いられている細線化手法について紹介し，細線化手法に類似したアルゴリズムを適用したボリュームの細線化手法の概要を述べる．続いて，ボリュームの細線化処理を用いた等値面生成手法の各処理について，詳しく述べる．本章では，細線化処理のために必要なデータ構造，細線化手法における 3 次元要素の探索順と除去条件，等値面の生成手法の順に述べる．続いて，本手法による等値面生成の処理速度を示し，本手法の有用性を示す．本章ではさらに，ボリューム・レンダリングへの本手法の適用について述べる．まず，ボリューム・レンダリングの従来研究成果を紹介する．続いて，視点からの等距離面を半透明に重ね描きするボリューム・レンダリング手法をとりあげ，ボリュームの細線化手法を用いた等値面生成手法によって視点からの等距離面を高速生成するアルゴリズムについて述べる．

第 8 章では，本論文の結論を述べる．

第 2 章

ラジオシティ法と2次元メッシュ生成手法

2.1 はじめに

本章では，ラジオシティ法，およびラジオシティ法に適用されている 2 次元メッシュ生成手法について，従来の研究成果を概観し，その問題点を列挙する．

2.2 節では，ラジオシティ法の概要を紹介する．幅広く用いられているプログレッシブ・レファインメント・ラジオシティ法 [13] では，入力形状をポリゴンの小片 (パッチ) の集合に分割し，パッチをさらに再分割してエレメントを生成し，エレメントごとに輝度値を算出する．ラジオシティ法におけるパッチやエレメントの生成手法は，ラジオシティ法の処理時間，生成される画像の写実性や画像生成時間に大きな影響を与えることが知られている．近年では，有限要素法等の数値解析で用いられている 2 次元メッシュ生成手法が，ラジオシティ法の入力形状の分割に適用されていることが多い．

2.3 節では，ラジオシティ法に適用されている 2 次元メッシュ生成手法について，従来の研究成果を概観する．特に，本論文で用いるデローニー三角メッシュ生成法 [42]，バブル・メッシュ法 [53] について詳しく示す．

2.4 節では，ラジオシティ法に適用されている適応的メッシュ生成手法について，従来の研究成果を概観する．適応的メッシュ生成手法は，数値分布を考慮して入力形状を分割する手法であり，画像生成時間を抑えながら写実性を高めることの出来る有用な手法である．本節では特に，適応的再分割法 [4]，輝度不連続線法 [40]，およびこれらの手法の問題点について，詳しく示す．

2.2 ラジオシティ法

本節では，ラジオシティ法の概要を示し，ラジオシティ法に用いられる入力形状の分割技術の要件について考察する．

2.2.1 ラジオシティ法の概要

ラジオシティ法 (Radiosity) [22] は，光エネルギーの平衡状態を算出することにより，入力物体の輝度を算出する手法である．レイ・トレーシング法などの他のレンダリングアルゴリズムと比較して，ラジオシティ法は緩やかな輝度分布の表現，例えば線光源や面光源のつくる不鮮明な陰影の表現に適している．

ラジオシティ法では、まずすべての入力形状において、光エネルギーの放射および反射を理想的な拡散であると仮定する。この時、ある面が発する光エネルギー（ラジオシティ）は、自己放射エネルギーの放射量と、他の物体から放射された光エネルギーの反射量の和である。そのため、面 i と面 j の光エネルギーの関係式は、式 2.1 のように示される (図 2.1 参照)。

$$B_i = E_i + \rho_i B_j F_{ij} \quad (2.1)$$

ここで、それぞれの変数は、

B_i : 面 i から発する光エネルギーの総量

E_i : 面 i の自己放射エネルギーの総量

ρ_i : 面 i の反射率

F_{ij} : 面 j から発した光エネルギーが面 i に到達する割合。フォーム・ファクタ (Form Factor) という。

を示すものである。

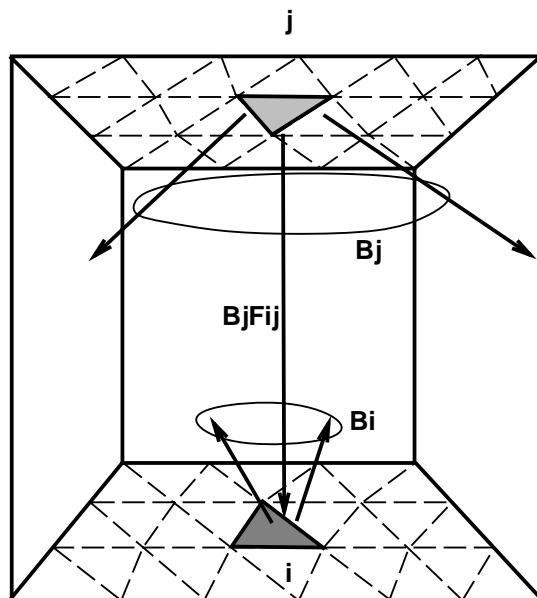


Figure 2.1: ラジオシティ法における光エネルギーの関係式

ある情景の輝度分布を求めるために，入力形状を n 枚の多角形の小片 (パッチ) に分割すると，式 2.1 より，各パッチの光エネルギーは，

$$\begin{aligned} B_1 &= E_1 + \rho_1 \sum_{j=1}^n B_j F_{1j} \\ B_2 &= E_2 + \rho_2 \sum_{j=1}^n B_j F_{2j} \\ &\dots \\ B_n &= E_n + \rho_n \sum_{j=1}^n B_j F_{nj} \end{aligned} \quad (2.2)$$

のように，連立方程式で表現される．これを行列を用いて表現すると，

$$\begin{pmatrix} 1 - \rho_1 F_{11} & -\rho_1 F_{12} & \dots & -\rho_1 F_{1n} \\ -\rho_2 F_{21} & 1 - \rho_2 F_{22} & \dots & -\rho_2 F_{2n} \\ \dots & \dots & \dots & \dots \\ -\rho_n F_{n1} & -\rho_n F_{n2} & \dots & 1 - \rho_n F_{nn} \end{pmatrix} \begin{pmatrix} B_1 \\ B_2 \\ \dots \\ B_n \end{pmatrix} = \begin{pmatrix} E_1 \\ E_2 \\ \dots \\ E_n \end{pmatrix} \quad (2.3)$$

と表わすことができる．この連立方程式は， E_i および ρ_i を適当に入力し， F_{ij} を算出した後に，ガウス=ザイデル (Gauss-Seidel) の反復法を用いて解くことができる．この時， E_i には光源パッチのみに 0 より大きい値を与えることが多い．

一般的にはラジオシティ法は，図 2.2 に示す通り，物体のパッチへの分割部と，光エネルギーの連立方程式算出部を総称した名称である．ラジオシティ法は入力形状をパッチに分割し，そのパッチの形状と輝度値を出力する．ラジオシティ法の出力結果をグラフィックス表示するためには，出力結果をレイ・トレーシング法や Z バッファ法などのレンダリング手法に入力する必要がある．

フォーム・ファクタは，半立方体 (hemicube) 法 [11] によって算出されることが多い．半立方体法では，パッチ i が他のパッチから収集するフォーム・ファクタを算出するために，パッチ i の重心点を中心とする半立方体を，5 面の Z バッファで形成する．続いて， i 以外のすべてのパッチを，5 面の Z バッファに投影する．フォーム・ファクタ F_{ij} は，パッチ j の Z バッファ上での面積の，5 面の Z バッファ全体の面積に対する比として算出される．

フォーム・ファクタを算出する別の手法として，十分大きな平面 1 枚を投影面として用いる手法 [56]，半球の底面を投影面として用いる手法 [15] などがあげられる．これらの手法はすべて，有限の解像度をもつ投影面を用いたサンプリング処理によって，フォー

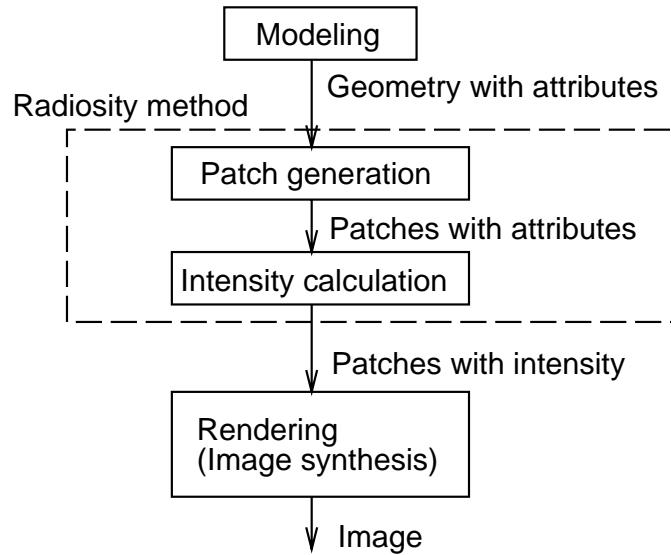


Figure 2.2: ラジオシティ法の処理の流れ図

ム・ファクタを算出している。投影面に用いる Z バッファのサンプリング誤差を減らすために、解析的算出を併用した手法 [3] も報告されているが、実用性の点で問題が残っている。

フォーム・ファクタ算出は処理量が大きく、ラジオシティ法の処理時間の大半を占めることが多い。ラジオシティ法で処理時間を短縮する最も単純な手段は、フォーム・ファクタ算出に用いる投影面の解像度を小さくすることである。しかし、投影面の解像度を小さくすることによって、フォーム・ファクタ算出のサンプリング誤差が大きくなり、不当に明るい(または暗い)パッチを生じることがある。この不当に明るい(または暗い)パッチは、エイリアシングを生じる要因となる。

2.2.2 プログレッシブ・レファインメント・ラジオシティ法

文献 [22] に示されるラジオシティ法のアルゴリズムでは、パッチの 2 乗のフォーム・ファクタ値をすべて算出した後に、すべてのパッチの光エネルギー値をガウス=ザイデル法で一気に算出する。そのため、フォーム・ファクタ値の保持のためにパッチの 2 乗のメモリ使用量を要する。また、すべてのフォーム・ファクタ値が算出されるまで、ラジオシティ法の算出経過をグラフィックス表示することは出来ない。この問題に対して Cohen ら

は、パッチ数に比例するメモリ使用量 (厳密にはメモリ使用量は半立方体などの投影面の解像度にも依存する) で、漸近的に収束解に近い近似解を得る、実用的なラジオシティ法 (Progressive Refinement Radiosity) [13] のアルゴリズムを報告した。

式 2.3 によって求められる光エネルギー値は、パッチ i が自己放射および他のパッチからの放射によって得る、つまりパッチ i が収集する (Gathering) 光エネルギーを算出する。これに対して文献 [13] では、パッチが他のパッチに放射する (Shooting) 光エネルギーを算出する数式に基づいてアルゴリズムを構築した。

このアルゴリズムでは、すべてのパッチは交互に、光エネルギーを瞬時に放射するモデルを仮定する。この時、パッチ i がパッチ j に放射する光エネルギー値は

$$B_{ji} = \rho_j B_i F_{ji} \quad (2.4)$$

で表わされる。ここで、それぞれの変数は、

B_{ji} : パッチ j がパッチ i から受けた光エネルギーを発する総量

ρ_j : 面 j の反射率

B_i : パッチ i が放射する光エネルギーの総量。

F_{ji} : パッチ j からパッチ i へのフォーム・ファクタ。

を示すものである。また、フォーム・ファクタの性質として、

$$A_j F_{ij} = A_i F_{ji} \quad (2.5)$$

が成立する。ここで、 A_i はパッチ i の面積を表わす。式 2.5 を用いることで、式 2.4 は

$$B_{ji} = \rho_j \frac{A_j}{A_i} B_i F_{ij} \quad (2.6)$$

と変形される。

一方、文献 [13] では、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向上するために、パッチをさらに細かいエレメントという小片に分割することを提案している。ここで、収集する光エネルギー量はエレメントを単位として算出される。また、未放射エネルギーの放射は、パッチを単位として算出される。パッチおよびエレメントの総数をそ

それぞれ $n, m (n < m)$ とし, パッチおよびエレメントの番号をそれぞれ p_i, e_i と表すと, 式 2.6 は,

$$\begin{aligned}
 B_{e_1} &= E_{e_1} + \rho_{e_1} \sum_{i=1}^n \frac{A_{e_1}}{A_{p_i}} B_{p_i} F_{p_i e_1} \\
 B_{e_2} &= E_{e_2} + \rho_{e_2} \sum_{i=1}^n \frac{A_{e_2}}{A_{p_i}} B_{p_i} F_{p_i e_2} \\
 &\dots \\
 B_{e_m} &= E_{e_m} + \rho_{e_m} \sum_{i=1}^n \frac{A_{e_m}}{A_{p_i}} B_{p_i} F_{p_i e_m}
 \end{aligned} \tag{2.7}$$

と表される. ここで, それぞれの変数は,

B_{e_m} : エレメント e_m が光エネルギーを発する総量

E_{e_m} : エレメント e_m が自己放射する光エネルギーの総量

ρ_{e_m} : エレメント e_m の反射率

B_{p_i} : パッチ p_i が光エネルギーを発する総量

A_{e_m} : エレメント e_m の面積

A_{p_i} : パッチ p_i の面積

$F_{p_i e_m}$: パッチ p_i から発した光エネルギーがエレメント e_m に到達する割合.

を示すものである.

従来のラジオシティ法では, 光を収集するエレメントの重心点に半立方体を設定してフォーム・ファクタを算出する. それに対して式 2.7 は, 光を放射するパッチ p_i の重心点に半立方体を設定して, エレメント e_j に放射する光エネルギーのフォーム・ファクタ $F_{p_i e_j}$ を算出するアルゴリズムを構築することが可能であることを示している. 文献 [13] では, この点に着目して, 下記のようなアルゴリズムを提案している.

1. 未放射光エネルギーの最も大きいパッチ p_i を選択する.
2. パッチ p_i の重心点に半立方体を設定し, パッチ p_i に含まれないすべてのエレメント e_j へのフォーム・ファクタ $F_{p_i e_j}$ を算出する.

3. 式 2.7 を用いて，パッチ p_i からエレメント e_j へ放射される光エネルギー値を算出し，エレメントの光エネルギー値 B_{e_j} に加算する．
4. 1. ~ 3. の処理を反復し，すべてのパッチの未放射光エネルギーがじゅうぶん小さくなったら処理を終了する．

このアルゴリズムでは，未放射光エネルギーの大きいパッチの光エネルギーが処理の前半で分配されるので，比較的高速に収束解に近い近似解を得ることができる．また，反復途中経過をグラフィックス表示することが容易なので，実用的である．

2.2.3 ラジオシティ法に用いるデータ構造

本論文では，ラジオシティ法の入力形状を，下記の通り定義する．

定義 2.1 本論文では，ラジオシティ法の入力形状を，同一平面上にある頂点で構成される多角形の集合と定義する．入力形状である多角形には，多角形の空洞(穴)が存在することがあるものとする．

続いて，ラジオシティ法における入力形状の分割結果を格納するデータ構造を，下記の通り定義する (図 2.3 (a)(b) 参照)．

定義 2.2 本論文では，ラジオシティ法の入力形状を細かく分割して出来るポリゴンの小片を，パッチ (Patch) と定義する．本論文では，パッチを三角形に限定する．パッチ P は，3 辺 $\{e_1, e_2, e_3\}$ ，3 頂点 $\{N_1, N_2, N_3\}$ ，および後述するエレメントの集合 $\{E_1, E_2, \dots\}$ より構成されるものとする．パッチ P の辺は，入力形状である多角形の边上にあるものを除いて，すべて 2 パッチに共有されているものとする．

定義 2.3 本論文では，パッチをさらに細かく分割して出来るポリゴンの小片を，エレメント (Element) と定義する．本論文では，エレメントを三角形に限定する．エレメント E は，3 辺 $\{e_1, e_2, e_3\}$ ，および 3 頂点 $\{N_1, N_2, N_3\}$ より構成される．エレメント E の辺は，入力形状である多角形の边上にあるものを除いて，すべて 2 エレメントに共有されているものとする．

定義 2.4 本論文では，パッチおよびエレメントの頂点を，ノード (Node) と定義する．ノード N は，座標値 $\{x, y, z\}$ ，および輝度値 $\{I\}$ より構成されるものとする．

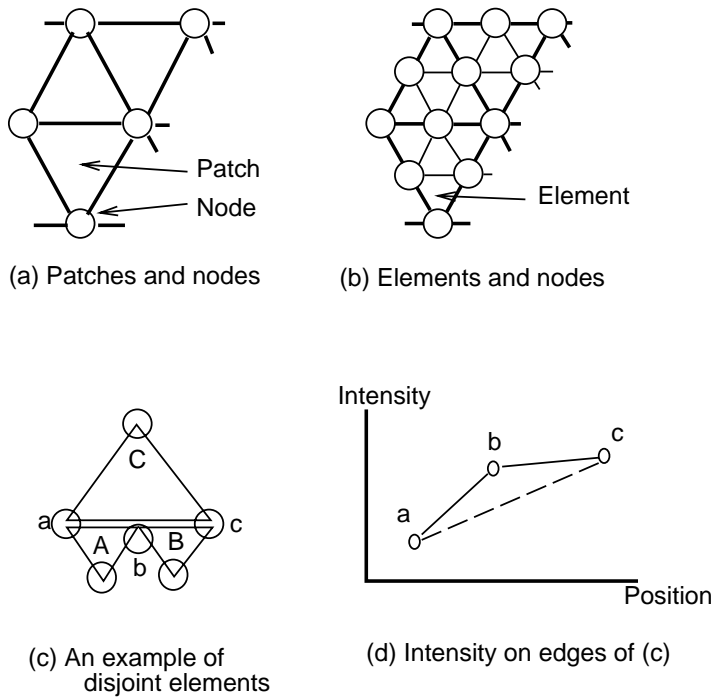


Figure 2.3: ラジオシティ法の形状データ構造

2.2.1 節で述べた通り，ラジオシティ法はなめらかな輝度分布をもつ情景の表現に適した手法である．ラジオシティ法の出力結果をなめらかにグラフィックス表示するためには，隣接エレメントの辺の両側における輝度値の連続性が保持される必要がある．そのため，ラジオシティ法では，定義 2.3 に示す通り，エレメントは隣接エレメントの辺が両端のノードを共有するように生成される必要がある．図 2.3 (c) は，隣接エレメントの辺が両端のノードを共有していない例である．この図において，辺 ab ， bc ， ac は，それぞれエレメント A ， B ， C のみに所有される．文献 [4] では，図 2.3 (c) のノード b のように，2 個のエレメントやパッチに共有されていない辺 ac の間に生成されるノードを，下記の通り定義している．

定義 2.5 隣接エレメントに共有されていないノードを，T 頂点 (T-vertex) と呼ぶ．ラジオシティ法の入力形状の分割において，T 頂点の生成は極力避ける必要がある．

後述するグーロー・シェーディングによって，エレメントの各辺上の輝度値は，ノードの輝度値から線形補間される．図 2.3 (d) に示す折れ線グラフは，図 2.3 (c) に示すエレメ

ントの辺上の輝度値である．実線は A の辺 ab , および B の辺 bc 上における輝度値を示す．点線は C の辺 ac 上における輝度値を示す．この折れ線グラフは, T 頂点の両側のエレメントの輝度値の連続性が保持されないことを示している．

ラジオシティ法の算出結果を用いた画像生成には, グーロー・シェーディング (Gouraud Shading) [23] という輝度算出手法を適用したレイ・トレーシング法や Z バッファ法が用いられることが多い．グーロー・シェーディングとは, エレメント内部の輝度分布をなめらかに表現するために, ノードに与えられた輝度値や法線ベクタを線形補間して, エレメント内部の輝度値を算出する手法である．

ラジオシティ法の算出結果をグーロー・シェーディングを用いて画像生成するためには, まずラジオシティ法によってエレメントごとに算出された光エネルギー値を用いて, ノードにおける輝度値 I を下記の式により算出する．

$$I = \frac{1}{n} \sum_{i=1}^n I_{E_i} \quad (2.8)$$

ただし, n はノードに隣接するエレメント数, I_{E_i} は隣接エレメント E_i の光エネルギー値である．

Z バッファ法におけるグーロー・シェーディングでは, まずエレメントの 3 頂点を 2 次元スクリーン座標系に投影し, 続いてスクリーン座標系において 3 頂点を結ぶ三角形の内部にある各画素における輝度値を算出する．この時, 3 頂点のスクリーン座標系での座標値を $(x_1, y_1), (x_2, y_2), (x_3, y_3)$ (ただし $y_1 > y_2 > y_3$) とし, 3 頂点の輝度値を I_1, I_2, I_3 とすると, 三角形の内部にある画素 (x, y) における輝度値 I は, $y_1 > y > y_2$ の時には

$$I = \frac{(x_{12} - x)I_{13} + (x - x_{13})I_{12}}{x_{13} - x_{12}} \quad (2.9)$$

ただし,

$$x_{12} = \frac{(y_1 - y)x_2 + (y - y_2)x_1}{y_1 - y_2}, \quad I_{12} = \frac{(y_1 - y)I_2 + (y - y_2)I_1}{y_1 - y_2},$$

$$x_{13} = \frac{(y_1 - y)x_3 + (y - y_3)x_1}{y_1 - y_3}, \quad I_{13} = \frac{(y_1 - y)I_3 + (y - y_3)I_1}{y_1 - y_3},$$

である．同様に, $y_2 > y > y_3$ の時には

$$I = \frac{(x_{23} - x)I_{13} + (x - x_{13})I_{23}}{x_{13} - x_{23}} \quad (2.10)$$

ただし,

$$x_{13} = \frac{(y_1 - y)x_3 + (y - y_3)x_1}{y_1 - y_3}, I_{13} = \frac{(y_1 - y)I_3 + (y - y_3)I_1}{y_1 - y_3},$$

$$x_{23} = \frac{(y_2 - y)x_3 + (y - y_3)x_2}{y_2 - y_3}, I_{23} = \frac{(y_2 - y)I_3 + (y - y_3)I_2}{y_2 - y_3},$$

である。

2.2.4 ラジオシティ法におけるパッチ/エレメント生成技術の要件

ラジオシティ法では、パッチやエレメントの生成技術が、生成される画像の写実性だけでなく、ラジオシティ法の処理時間、ラジオシティ法の出力結果のデータ量、画像生成時の処理時間などに、大きな影響を与えることが多い。以下、従来の研究報告で指摘されてきた、ラジオシティ法のためのパッチやエレメントの生成手法に求められる要件 [54] を列挙する。

要件 2.1 本論文で定義された入力形状は、定義 2.1 に示される通り、凹部や穴を含む多角形である。そこで、これらの特徴を含む多角形に対しても、的確にかつ自動的にパッチやエレメントを生成する必要がある。

要件 2.2 ラジオシティ法の写実性と処理時間、データ量は、ともにパッチやエレメントの数に大きく依存する。計算機資源の許される範囲内でラジオシティ法の処理時間や出力結果のデータ量、フォーム・ファクタ値の算出精度、画像生成の処理量を制御するためには、パッチやエレメントの大きさを自在に制御できることが望ましい。また、処理時間やデータ量の増加を抑えながら写実性を向上するためには、輝度変化の大きい部位（例えば陰影の境界部分など）だけを適応的に細かく分割できることが望ましい。

要件 2.3 一般的にラジオシティ法では、パッチやエレメントを細かく生成することによって、写実性を向上することが出来る。しかし、パッチやエレメントを細かく生成するには、同時にフォーム・ファクタの算出精度を高める必要がある。さもないと、極端に小さいエレメントや、極端に細長く歪んだエレメントが生成された際に、フォーム・ファクタのサンプリング誤差の要因となりやすい。そのため、指定されたエレメントの大きさよりも極端に小さいエレメントや、極端に細長く歪んだエレメントは、極力生成されないことが望ましい。

ところで、ラジオシティ法のパッチやエレメント生成技術と同様な、多角形を三角形や四角形の小片の集合に分割する技術として、有限要素法や境界要素法などの数値解析に

用いられる，2次元メッシュ生成手法が知られている．有限要素法や境界要素法におけるメッシュに求められる要件は，上記のラジオシティ法における要件に類似している．そこで，ラジオシティ法のパッチやエレメント生成には，数値解析のための2次元メッシュ生成手法が適用されることが多い．次節では，2次元メッシュ生成手法について，従来の研究成果を概観する．

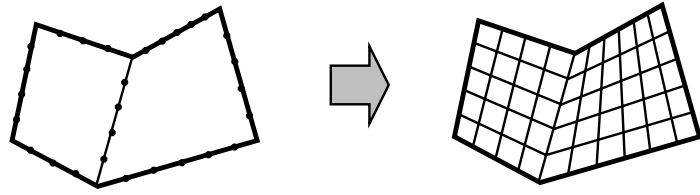
2.3 2次元メッシュ生成手法

本節では，有限要素法などの数値解析手法に用いられている，2次元メッシュ生成手法について，従来の研究成果を概観する．

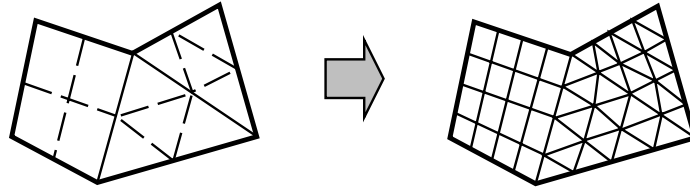
2.3.1 2次元メッシュ生成手法の概観

有限要素法などの数値解析技術が発展するにつれて，メッシュ生成技術が数値解析の処理時間および精度に大きな影響を与えることが知られるようになった．そこで近年では，多くのメッシュ生成手法が研究報告されてきた．過去に提案されている2次元メッシュ生成手法には，下記のような手法があげられる（図2.4参照）．

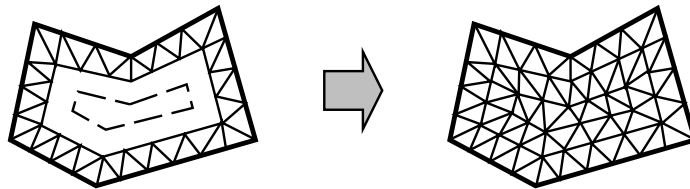
- (a) 格子パターンのマッピング：領域を正三角形に近い三角形や長方形に近い四角形に対角線分割した後，三角形や四角形の格子パターンを領域内部にマッピングする手法 [31]．凹部や穴を含む多角形領域の対角線分割の自動化が難しいので，要件 2.1 を満たすことが困難である．また，局部的に要素の大きさを制御することが難しいので，要件 2.2 を満たすことが困難である．
- (b) 再帰的分割：領域を正三角形に近い三角形や長方形に近い四角形に対角線分割した後，長い辺を再帰的に分割して小さい要素を生成する手法 [70]．(a) と同様の理由により，要件 2.1 および要件 2.2 を満たすことが困難である．
- (c) 外枠からの要素生成：領域境界に接触する帯状領域に三角形要素を生成し，以後領域の内側へ順に三角形要素を生成する手法 [20]．領域中心付近の要素に歪みが生じやすいので，要件 2.3 を満たすことが困難である．
- (d) 格子点を優先した手法：領域の境界上および内部に要素頂点を多数生成し，それを最適に結んで三角形要素を生成する手法．有名なものにデローニー三角メッシュ生



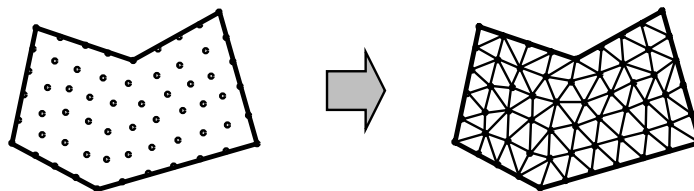
(a) Pattern mapping method



(b) Recursive subdivision method



(c) Advancing front method



(d) Node-first method

Figure 2.4: 代表的な 2 次元メッシュ生成手法

成法 (Delaunay Triangulation) [42] がある。凹部や穴をもつ多角形のメッシュ生成が容易である。また、要素の大きさの制御性や、要素の歪みについては、要素頂点の生成手法に大きく依存する。

2.3.2 デローニー三角メッシュ生成法

デローニー三角メッシュ生成法 (Delaunay triangulation) は、領域内に与えられた多数の点を連結して、三角メッシュを生成する手法である。デローニー三角メッシュの本来の定義は、領域内に与えられた多数の点から得られるボロノイ図の相対である [50] (図 2.5 参照)。近年では、ボロノイ図を用いずにデローニー三角メッシュを生成する、さまざまな高速化手法が報告されている [42] [58]。

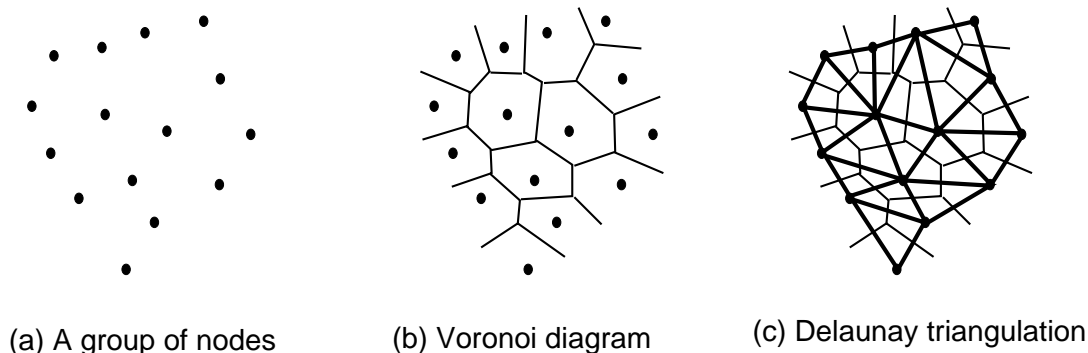


Figure 2.5: デローニー三角メッシュの定義

デローニー三角メッシュ生成法は、凹部や穴をもつ多角形においても、自動的にメッシュを生成することが出来るので、要件 2.1 を満たすことができる。また、デローニー三角メッシュ生成法では、適切に与えられた要素頂点に対して、最も歪みの少ない三角メッシュを生成することができる。よって、要素頂点が適切に与えられれば、要件 2.2 および要件 2.3 を満たすことが出来る。そのため近年では、デローニー三角メッシュ生成法をラジオシティ法のパッチ/エレメント生成に適用した報告が増えている。

2.3.3 バブル・メッシュ法

バブル・メッシュ法 (Bubble mesh) [53] は、デローニー三角メッシュ生成法の実行前に必要な要素頂点を生成する手法であり、下記の直観的仮定に基づいて構築された手法である。

仮定 2.1 領域内部、および領域境界上に球を最密充填し、その中心を連結して出来る三角メッシュは、要素の形状に歪みが少ない。

本論文の第3章および第4章にて提案する手法にも，バブル・メッシュ法が適用されている．

バブル・メッシュ法は，仮定 2.1 に基づいて，質量と分子間力をもった仮想的な球状物体を領域内に多数生成し，その球状物体が領域内で最密充填となるように力学シミュレーションを解き，その収束解における球状物体の位置を要素頂点の位置として出力する．図 2.6 (a)(b) に示すように，バブル・メッシュ法では力学シミュレーションは入力形状の次元の低い順に実行する．つまり，頂点や辺上に最初に球状物体を配置し，続いて領域内部に球状物体を配置する．本論文の適用範囲ではないが，バブル・メッシュ法の原理は，3次元入力形状の四面体メッシュ生成にも適用することができる．3次元入力形状の場合には，入力形状の境界面である2次元領域内部に球状物体の充填した後に，3次元領域内部に球状物体を充填する．

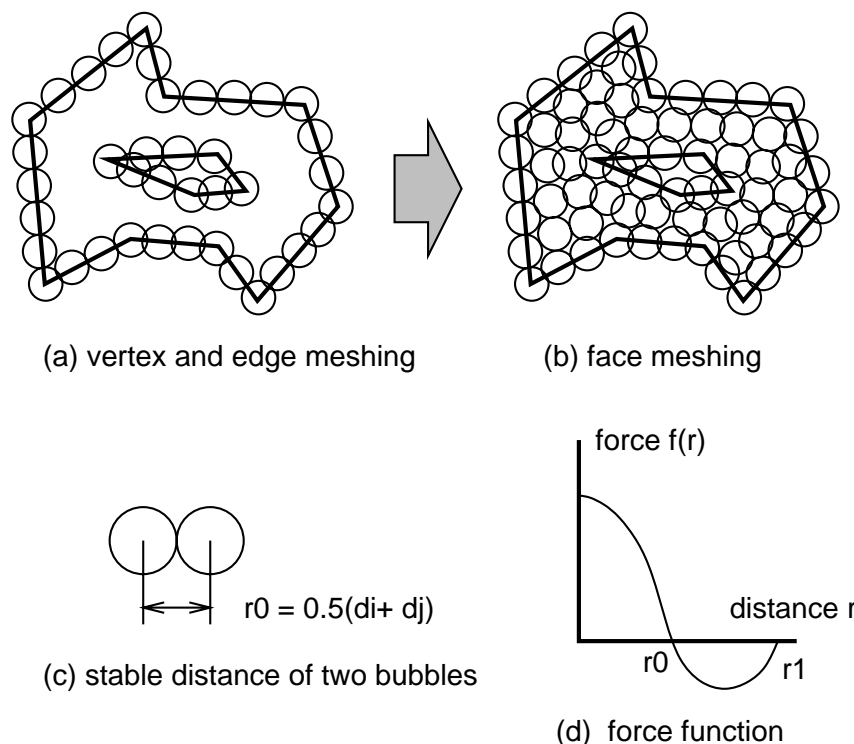


Figure 2.6: バブル・メッシュ法の原理

ここで，隣接する2個の球状物体 B_i および B_j の直径を d_i, d_j とすると，球状物体

の安定距離 r_0 は,

$$r_0 = 0.5(d_i + d_j) \quad (2.11)$$

で表現される．力学シミュレーションを解くために， $B_i B_j$ 間の距離が r_0 以下である時には斥力， r_0 以上である時には引力となるような分子間力を，下記の条件を満たす 3 次関数 (図 2.6 (d) 参照) で近似する．

$$\begin{aligned} f(r_0) &= 0, f'(r_0) = 0, f''(r_0) = 0, \\ f(r_1) &= 0, f(r) = 0 (r > r_1) \end{aligned} \quad (2.12)$$

この 3 次関数を用いることで，球状物体の運動方程式を

$$m \frac{d^2 x(t)}{dt^2} + c \frac{dx(t)}{dt} = f(t) \quad (2.13)$$

のように表すことが出来る．この式は 2 次常微分方程式であるので，Runge-Kutta 法などの手法を用いることで，反復処理により要素頂点の位置を決定することが出来る．ここで， m は球状物体の質量， c は粘性抵抗係数， $x(t)$ は球状物体の位置を表す．

図 2.7 は，バブル・メッシュ法によって得られる三角メッシュの例である．左上は球状物体の初期位置，左下は力学シミュレーションの反復処理によって求められた球状物体の位置である．右上および右下の三角メッシュは，各々の球状物体の中心を要素頂点として，デローニー三角メッシュ生成法により生成したものである．右上の三角メッシュには細長い三角形要素が多く見られるが，右下の三角メッシュにはほとんど見られない．この結果から，バブル・メッシュ法は要素の歪みを低減する有効な手法であることがわかる．

2.4 適応的メッシュ生成手法

2.2.4 節の要件 2.2 に示した通り，ラジオシティ法で処理時間やデータ量の増加を抑えながら写実性を向上するためには，輝度変化の大きい部位だけを適応的に細かく分割できることが望ましい．有限要素法などの数値解析の分野においては，処理時間の増加を抑えながら解の精度を高めるために，数値分布を考慮してメッシュを生成する適応的メッシュ生成手法 (Adaptive meshing) が，数多く提案されてきた．近年では，適応的メッシュ生成手法をラジオシティ法に適用する研究もいくつか報告されている．本節では，適応的メッシュ生成手法と，そのラジオシティ法への適用について，従来の研究成果を概観する．

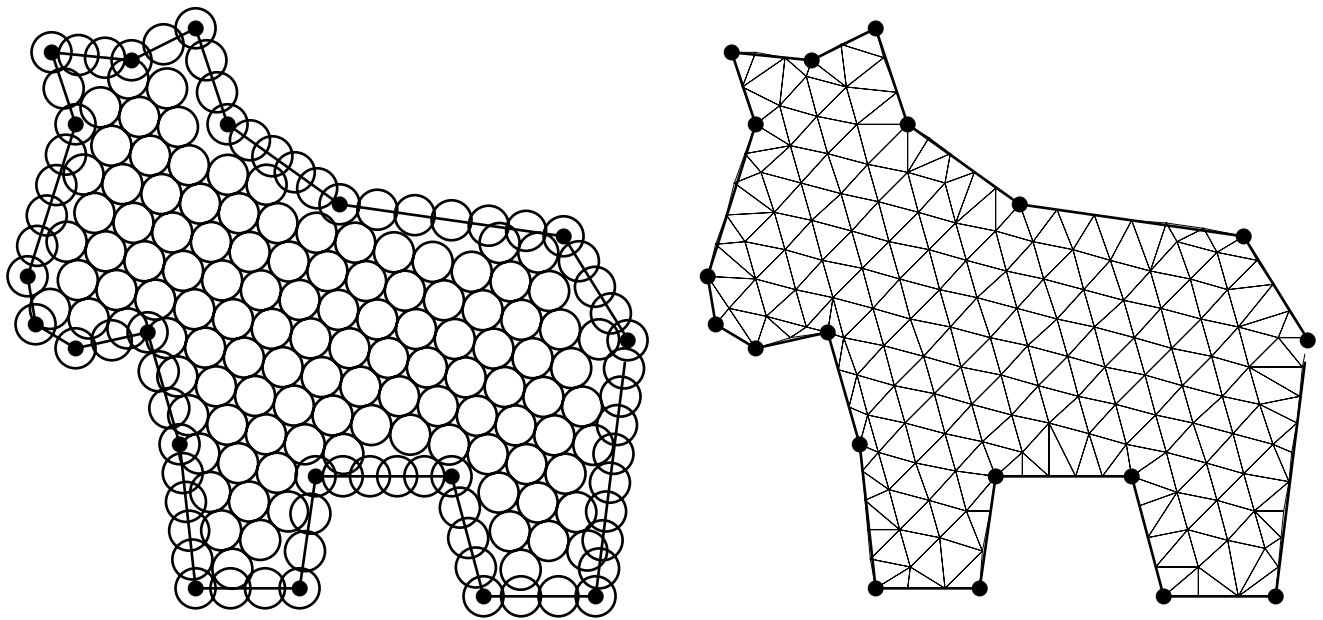


Figure 2.7: バブル・メッシュ法によって得られる三角メッシュ

2.4.1 適応的メッシュ生成手法の概観

適応的メッシュ生成手法は、数値分析に適応した条件をメッシュ生成前の入力形状に加える前処理型 (Prior method) 適応的メッシュ生成と、数値解析結果を参照してメッシュを修正する後処理型 (Posterior method) 適応的メッシュ生成に大別される [52]。数値解析の分野で用いられている後処理型適応的メッシュ生成手法は、以下の3種類の手法に大

別される。

h 法: 数値変化の大きい部位の要素を再分割する [44]。実装が単純である。

p 法: 要素の形状や、数値の補間関数の次数をあげる [74]。要素数の増加を抑えることができる。

r 法: 要素の接続関係を保持しながら、要素頂点の位置を修正する [45]。

2.4.2 適応的メッシュ生成手法を用いたラジオシティ法

2.4.1 節で述べた適応的メッシュ生成手法のうち、p 法および h 法については、ラジオシティ法への適用例が報告されている。

p 法は文献 [62] [72] などでラジオシティ法に適用されている。要素の再分割などを行わずに、要素内部の輝度値を高次式で補間するので、データ量の増加を防ぐと同時に、隣接エレメント間のマッハバンドを減少出来るなどの利点がある。しかし、実装が難しい上に、多くのグラフィックスハードウェアやソフトウェアが輝度値の高次補間機能を持っていないので、実用的とは言えない。そのためこの手法は、文献 [62] [72] の報告後には活発な議論は進んでいない。

h 法は文献 [4] [8] などでラジオシティ法に適用されている。文献 [4] では、この手法を 適応的再分割法 (Adaptive subdivision) と呼んでいる。本論文では、下記の通り適応的再分割法を定義する。

定義 2.6 適応的再分割法では、1 回目のラジオシティ法の反復処理の後、隣接エレメント間の輝度差を算出し、輝度差の大きいエレメントをさらに細かく分割する。この時、T 頂点が生じないように隣接エレメントを同時に分割する必要がある。

適応的再分割法では、実装が単純であり、再分割自体の処理時間が小さいという利点がある。一方、処理のしきい値を隣接頂点の輝度差で与えるので、利用者にとって直観的でなく、熟練を要するという問題点がある。また、適応的再分割法では、再分割によって 2 回目以降のラジオシティ法では要素数が増加するが、この増加数の見積りが難しい。特に、しきい値の設定が悪いと、膨大な数の要素が分割され、ラジオシティ法の処理時間やメモリ使用量、ラジオシティ法実行後の画像生成の処理時間に大きな影響を与える可能性があ

る．そのため，適応的再分割法では，1 回目のラジオシティ法ではあまり細かいメッシュを用いることも多い．しかしその場合，要素サイズよりも小さな陰影を表現することが困難である (図 2.8 参照) ．

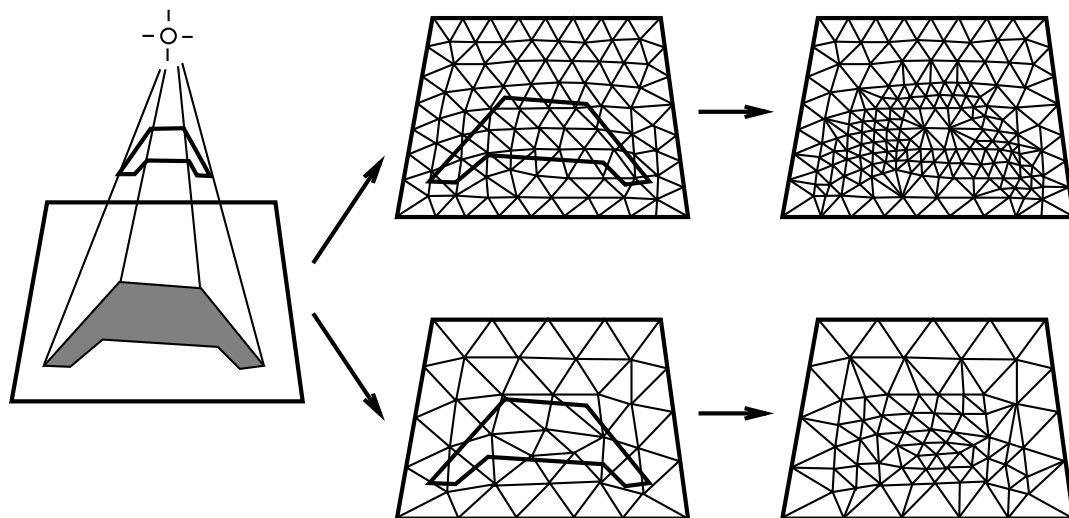


Figure 2.8: 適応的再分割法の例

2.4.3 輝度不連続線を用いたラジオシティ法

Lischinski らは，適応的再分割法を用いずに輝度変化の大きい部位を写實的に表現する手法として，輝度不連続線法 (Discontinuity Meshing) [40] を提案した．ここで，輝度不連続線法のために文献 [40] に示された定義は，下記の通りである．

定義 2.7 輝度不連続線法を適用する際には，光源を凸多角形に限定する．

定義 2.8 光源から直接放射される光線を受光する面を 受光面 (Receiver)，光源から直接放射される光線を遮蔽する面を 遮蔽面 (Occluder) と呼ぶ．

定義 2.9 光源の頂点 S_n を出発点として，遮蔽面の一辺の両端点 O_i, O_j を通過する半無限直線を生成し，半無限直線上にある無限遠点をそれぞれ I_i, I_j とする．この時，半無限面 $O_i I_i I_j O_j$ を，輝度不連続面 (Critical surface) と定義する．同様に，光源の辺の両端点 S_m, S_n を出発点として，遮蔽面の頂点 O_i を通過する半無限直線を生成し，半無限

直線上にある無限遠点をそれぞれ I_m, I_n とする．この時，半無限な面 $O_i I_m I_n$ も，輝度不連続面 (Critical surface) と定義する．

定義 2.10 物体と輝度不連続面との交線を，輝度不連続線 (Discontinuity edge) と定義する．

定義 2.11 輝度不連続線法では，光源からの直接光を受光する物体上に輝度不連続線を生成し，これを交差しないようにパッチやエレメントを生成し，ラジオシティ法を実行する．

文献 [40] では，輝度不連続線の両側の輝度値の特性についても，下記の通り定義している．

定義 2.12 平行光源や点光源がつくる輝度不連続面および輝度不連続線の両側の輝度値の特性を， D_0 不連続 (D_0 discontinuity) と定義する．線光源や面光源がつくる輝度不連続面および輝度不連続線の両側の輝度値の特性を， D_1 不連続 (D_1 discontinuity) と定義する．

図 2.9 (a) は，点光源と遮蔽面がつくる輝度不連続面の例である．輝度不連続面は，点光源 s と遮蔽面の辺 $O_1 O_2, O_2 O_3, \dots$ のうちの 1 辺を含む平面である．

図 2.9 (b) は，面光源と遮蔽面がつくる輝度不連続面の例である．輝度不連続面は，面光源の頂点 S_1, S_2, \dots のうちの 1 頂点と，遮蔽面の辺 $O_1 O_2, O_2 O_3, \dots$ のうちの 1 辺を含む平面，あるいは面光源の辺 $S_1 S_2, S_2 S_3, \dots$ のうちの 1 辺と，遮蔽面の頂点 O_1, O_2, \dots のうちの 1 頂点を含む平面である．

図 2.9 (b) の受光面上には，輝度不連続線が密集している部位がいくつか存在する．この周辺では，輝度不連続線によって受光面が多くの部分に分割される．この時，受光面分割のための輝度不連続線どうしの交点計算に多くの処理時間を要するとともに，受光面分割後のデータ量が大きく増加する．また，この部位では受光面が極端に小さい領域や細長い領域に分割される可能性が高く，その結果極端に小さいエレメントや細長いエレメントが生成される可能性が高い．よってこの手法は，要件 2.3 を満たさなくなることが多い．

文献 [40] に示した輝度不連続線の生成手法は，ラジオシティ法に適用してはいないものの，文献 [9] にてすでに報告されていた．文献 [9] では，光源を点光源等に限定し，レンダリング実行前に輝度不連続線で物体を分割し，分割された各部分ごとに陰影判定を実行して記録する．この手法では，通常のレンダリング実行中の，各画素ごとの陰影判定を

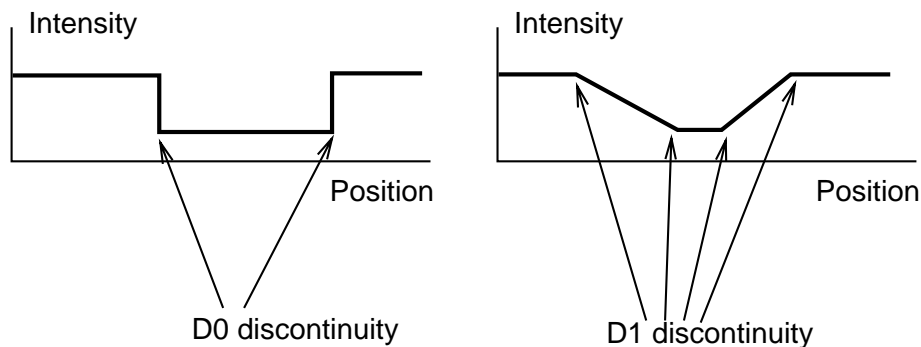
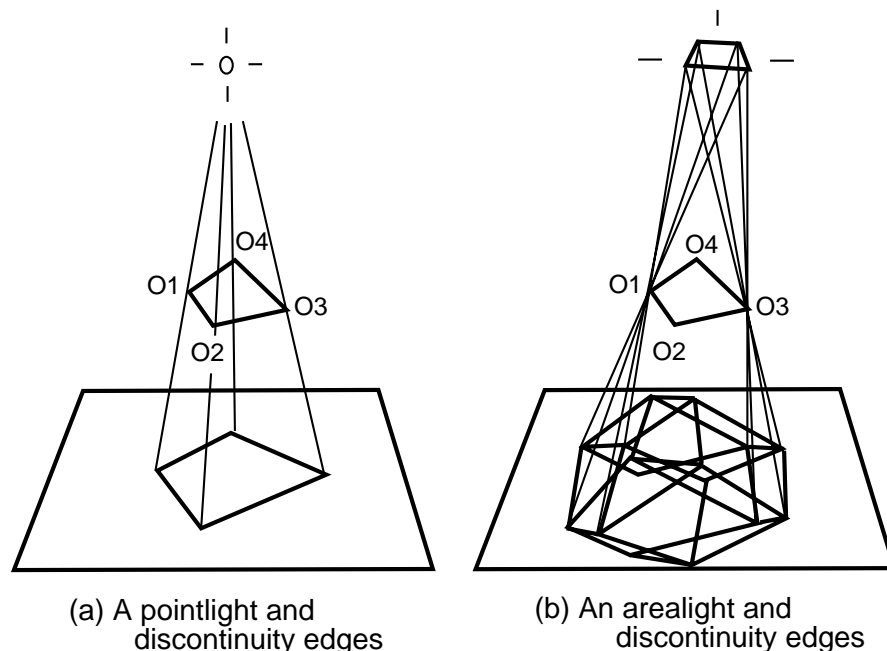


Figure 2.9: 輝度不連続線の生成例

省略することが出来るので、アニメーション等に用いるための高速レンダリングを実現することが出来る。

受光面を輝度不連続線で分割するために、文献 [9] [40] では BSP 法 (図 2.10 (a) 参照) を用いている。BSP 法は文献 [60] 以来、コンピュータ・グラフィックスで頻繁に適用されるようになった、二分木のデータ構造を定義した手法である。BSP 法は、複雑に分割された面を階層的に処理することで、レンダリング時の交点計算などの高速化に用いることが出来る。しかし、BSP 法は交線が 1 本生じるたびに受光面を 2 分するので、メモリ使用

量が増大しやすい。また，受光面の分割数が多くなるので，極端に小さいエレメントや細長いエレメントが生成される可能性が高くなり，要件 2.3 を満たさなくなることが多い。

図 2.10(b) は，図 2.9 (b) の受光面を BSP 法により分割した例である。非常に小さい領域や細長い領域が生成されていることがわかる。

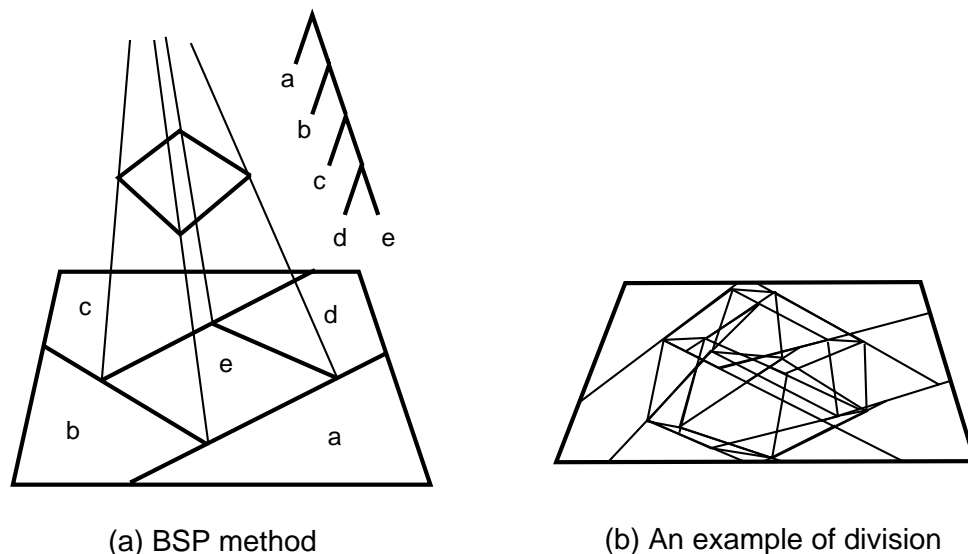


Figure 2.10: BSP 法を用いた輝度不連続線の管理

2.5 むすび

本章では，ラジオシティ法の概要，有限要素法等の数値解析で用いられている 2 次元メッシュ生成技術，および数値分布を考慮した適応的メッシュ生成手法について，従来の研究成果を概観した。ラジオシティ法において，処理時間やデータ量の増加を抑えながら，画像の写実性を高める手法として，適応的メッシュ生成手法が有用である。適応的メッシュ生成手法は，前処理型適応的メッシュ生成手法と，後処理型適応的メッシュ生成手法に大別される。

ラジオシティ法に適用されている代表的な前処理型適応的メッシュ生成手法に，輝度不連続線法があげられる。しかし，輝度不連続線法には，

- 大量の輝度不連続線が生成されたときに，輝度不連続線の交点計算に多くの計算量

を要する。

- 多数の輝度不連続線が複雑に交差する部位では、非常に小さいエレメントや細長いエレメントが生成され、ラジオシティ法のフォーム・ファクタ算出時のサンプリング誤差を生じる可能性が高くなる。

などの問題点がある。

ラジオシティ法に適用されている代表的な後処理型適応的メッシュ生成手法に、適応的再分割法があげられる。しかし、適応的再分割法には、

- 要素の再分割の条件として、隣接エレメントや隣接ノードの輝度差をパラメータ入力するケースが多い。輝度差をパラメータとする場合、不慣れな利用者には試行錯誤を強いられるケースが多い。
- 1回目のラジオシティ法に用いる要素が小さいと、2回目のラジオシティ法における要素数が非常に多くなる可能性が高くなる。
- 逆に1回目のラジオシティ法に用いる要素が大きいと、要素数の増加を抑えることができる反面、小さな陰影の表現を欠落する可能性が高くなる。

などの問題点がある。

第 3 章

簡略な輝度不連続線法

3.1 はじめに

第2章では、ラジオシティ法のための前処理型適応的メッシュ生成、および後処理型適応的メッシュ生成について、従来の研究成果を概観した。

ラジオシティ法のための前処理型適応的メッシュ生成の代表的な手法に、輝度不連続線法 [40] があげられる。2.4.3 節で述べた通り、従来の輝度不連続線法では、多くの輝度不連続線が特定の部位に集中することが多い。このため、従来の輝度不連続線法には、

- 大量の輝度不連続線が生成されたときに、輝度不連続線の交点計算に多くの計算量を要する。
- 多数の輝度不連続線が複雑に交差する部位では、非常に小さい要素や細長い要素が生成され、フォーム・ファクタの算出時のサンプリング誤差を生じる可能性が高くなる。

などの問題点がある。本章では、上記の問題点を解決するために、輝度不連続線の生成数を減少した、簡略な輝度不連続線法 [27] を提案する。

3.2 節では、簡略な輝度不連続線法の概要について述べる。本手法では、光源からの直射光が一部照射される半影領域 (Penumbra) と、直射光が完全に遮蔽される本影領域 (Umbra) の境界線、および直射光が完全に照射される照射領域と半影領域の境界線を、輝度不連続線として生成する。本手法は輝度不連続線の生成数を減少するので、輝度不連続線どうしの交点計算の処理量を低減するとともに、極端に小さいエレメントや極端に細長いエレメントの生成を低減する。また、本手法では、輝度変化の大きい半影領域のみにおいて要素を細かく生成し、他の領域では半影領域よりも要素を大きく生成することができるので、エレメント数の増加を抑えながら、写実的な結果を得ることができる。

3.3 節では、簡略な輝度不連続線法の各処理部について、詳しく示す。具体的には、影多面体 [46] の生成部、影多面体と受光面の交線計算、輝度不連続線や受光面の境界によって囲まれる各領域の定義、各領域と影多面体との内外判定、メッシュ生成、の各処理について示す。

3.2 ラジオシティ法のための簡略な輝度不連続線法の概要

まず、従来の輝度不連続線法におけるメッシュ生成結果について考察する。図 3.1(a) は、面光源および遮蔽面が 1 個ずつ存在する場合の、輝度不連続線の生成例である。図 3.1(b) は、図 3.1(a) に示す輝度不連続線を交差しないように、パッチおよびエレメントを生成した例である。図 3.1(b) において、輝度不連続線が集中している部位の周辺では、輝度不連続線によって極端に小さいエレメントおよび細長いエレメントがいくつか生成されている。そのため、従来の輝度不連続線法では、パッチやエレメントの生成手法の要件 2.3 を満たさないことが多い。

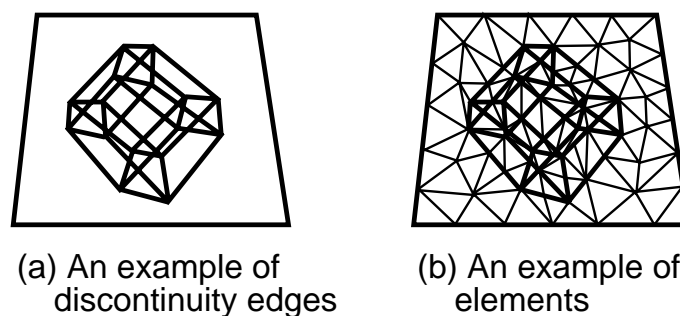


Figure 3.1: 従来の輝度不連続線法によるメッシュ生成

ここで、輝度不連続線法によって生じる極端に小さいエレメントや細長いエレメントの生成について、下記のような仮定を設ける。

仮定 3.1 受光面上に生成される輝度不連続線が増加し、受光面の分割数が増加するほど、極端に小さい領域や細長い領域が発生する可能性が高くなり、極端に小さいエレメントや細長いエレメントが生成される可能性が高くなる。この状況を抑える一手段として、輝度不連続線の生成数を減らすことが考えられる。

また、光源と受光面の照射状態について、以下のように定義する。

定義 3.1 光源から照射される直射光と受光面の関係は、光源からの直射光が全部到達する 照射領域 (Bright)，直射光が全部遮蔽される 本影領域 (Umbra)，および直射光が一部到達する 半影領域 (Penumbra) の 3 種類に分類される。

この時、下記の定理 (証明は本章の付録を参照) が成立する。

定理 3.1 平行光源または点光源がつくる D_0 輝度不連続線は、照射領域と本影領域の境界上に存在する。

定理 3.2 面光源がつくる D_1 輝度不連続線は、照射領域と半影領域の境界、本影領域と半影領域の境界、および半影領域内部のいずれかの領域に存在する。

定理 3.2 により、図 3.1(a) において輝度不連続線が集中している領域は、半影領域であることがわかる。ここで、輝度不連続線が存在する受光面上における輝度変化について、下記の通り仮定する。

仮定 3.2 半影領域内部には多数の輝度不連続線が生成されるので、他の領域よりも輝度変化が大きいと推測することが出来る。よって、半影領域のみにパッチおよびエレメントを細かく生成し、他の領域では半影領域よりもパッチおよびエレメントを大きく生成することで、パッチ数およびエレメント数の増加を抑えながら、写實的に陰影を表現することができる。

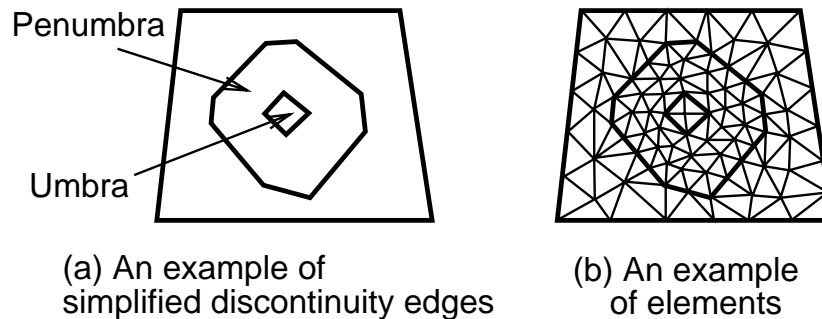


Figure 3.2: 簡略な輝度不連続線法によるメッシュ生成

図 3.2(a) は、図 3.1(a) と同一の受光面を、照射領域、半影領域、本影領域の境界線で分割した例である。図 3.2(b) は、仮定 3.2 で輝度変化が大きいと仮定された半影領域で、パッチやエレメントを細かく生成した例である。図 3.1(b) に見られたような、極端に小さいエレメントおよび細長いエレメントが減少されていることがわかる。

ここで，照射領域または本影領域と半影領域の境界線は， D_1 輝度不連続線であると考えることが出来る．また，照射領域と本影領域の境界線は， D_0 輝度不連続線であると考えることが出来る．つまり，図 3.2(a) に示した受光面の分割手法は，輝度不連続線の生成数を減少した，簡略な輝度不連続線法と位置づけることができる．

本章では，下記の定義による簡略な輝度不連続線法を提案する．

定義 3.2 平行光源，点光源，面光源の各光源からの直射光が，多角形の遮蔽面によって遮蔽される領域は，半無限な多面体となる．この半無限な多面体を，影多面体 (Shadow polyhedron) と定義する．

定義 3.3 本章では，影多面体と受光面の交線で受光面を分割して，パッチやエレメントを生成する手法を，簡略な輝度不連続線法 (Simplified discontinuity meshing) と定義する．

定義 3.4 平行光源や点光源が形成する影多面体は 1 個であり，その内部では直射光が完全に遮蔽されている．この内部の空間を 本影空間 (Umbral space) と定義する．面光源が形成する影多面体は 2 個であり，その一方は本影空間を形成する．他方の影多面体の内部では，直射光の一部が遮蔽されている．この内部の空間を 半影空間 (Penumbral space) と定義する．簡略な輝度不連続線法では，このそれぞれの影多面体との交線を，輝度不連続線として扱う．

簡略な輝度不連続線法は，極端に小さいエレメントや細長いエレメントの生成を低減するとともに，輝度不連続線どうしの交点計算の処理量を減少することが期待できる．図 3.3 (a) は，従来の輝度不連続線法によって生成された輝度不連続線の例であり，図 3.3 (b) は本手法によって生成された輝度不連続線の例である．光源の頂点数を N_s ，遮蔽面の頂点数を N_o とすると，輝度不連続線の総数は，図 3.3 (a) では $2N_s N_o$ ，図 3.3 (b) では $N_s + 2N_o$ である．よって本手法では，輝度不連続線の生成数が減少するので，輝度不連続線どうしの交点計算の処理量が減少すると期待される．

本手法では，輝度不連続線が生成された受光面を，BSP 法を用いずに，輝度不連続線 (および受光面の境界線) によって一周されるループを単位として分割する．パッチやエレメントの生成にはデローニー三角メッシュ生成法を用いる．図 3.4 (b) は，図 3.4 (a) に示された簡略な輝度不連続線を用いて，受光面を BSP 法によって分割した例である．図

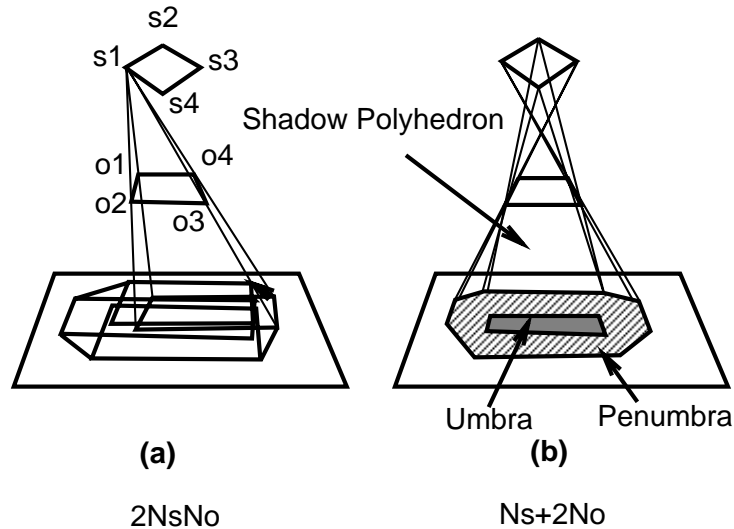


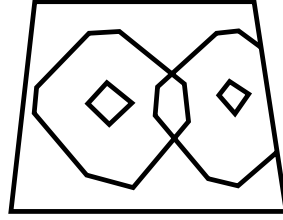
Figure 3.3: 影多面体との交線として得られる輝度不連続線

3.4 (c) は受光面をループ単位で分割した例である．受光面の分割数は，図 3.4(b) では 29 であるのに対し，図 3.4 (c) では 6 と，大幅に減少している．図 3.4 (d)(e) はそれぞれ，図 3.4 (b)(c) に示された受光面からエレメントを生成した例である．図 3.4(e) では，図 3.4 (d) と比較して，歪みの少ないエレメントが生成されていることが示されている．

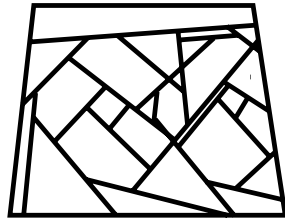
本手法では，半影領域のみにおいてメッシュを細かく生成するために，受光面を分割したそれぞれの領域が半影領域に属するか否かを判定する必要がある．本手法では，下記に定義される照射状態を，それぞれの領域ごとに判別し，その結果に基づいて各領域のパッチおよびエレメントの大きさを決定する．

定義 3.5 本章では，輝度不連続線によって分割された各領域が，光源からの直射光が完全に照射されている照射領域，直射光線の一部が遮蔽されている半影領域，直射光線の全部が遮蔽されている本影領域のいずれに属するかを判定した結果を，照射状態 と呼ぶ．照射状態は，それぞれの光源がつくる影多面体と，受光面が分割されてできる領域との内外判定によって判別される．

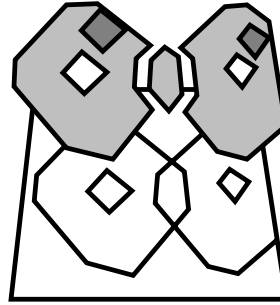
図 3.5 は，本章で提案する，影多面体を用いた輝度不連続線法のアルゴリズムを，疑似コードで表したものである．各処理の単純化のため，本章では，入力形状を凸多角形 (Convex polygon) に限定する．そのため，凹部や穴を含む多角形は，複数の凸多角形に



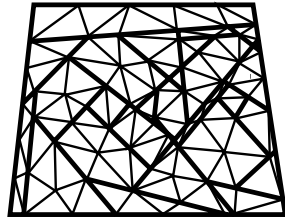
(a) An example of simplified discontinuity edges



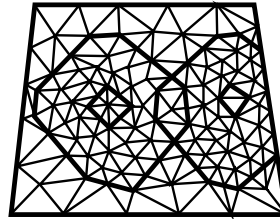
(b) Division by BSP method



(c) Division by loop operation



(d) An example of elements



(e) An example of elements

Figure 3.4: ループ単位による受光面の分割

分割されているものとする。

3.3 ラジオシティ法のための簡略な輝度不連続線法の各処理

本節では、影多面体を用いた輝度不連続線法の主要技術である、影多面体の生成、輝度不連続線の生成、受光面の分割、照射状態の判定、パッチやエレメントの生成について

```

void DiscontinuityMeshing() {
    for( each polygon ) {
        Divide into convex polygons;
    }
    for( each light source ) {
        for( each polygon ) {
            Generate a shadow polyhedron;
            for( each polygon ) {
                Generate discontinuity edges;
                Construct discontinuity loops;
                Check the relationship between loops and the light source;
            }
        }
    }
    for( each polygon ) {
        Divide the polygon into faces
        enclosed by discontinuity loops;
        Specify the size of patches and thats of elements;
        Genarete patches and elements in the faces;
    }
}

```

Figure 3.5: 影多面体を用いた輝度不連続線法の擬似コード

て，その実装方法を述べる．

3.3.1 影多面体の生成

簡略な輝度不連続線法では，光源および物体を凸多角形に限定し，文献 [46] に類似した下記の手法で，影多面体を生成する．以下，面光源がつくる本影空間および半影空間の影多面体の生成について述べる．

まず，本影空間を生成する手法について述べる．本手法では，遮蔽面の一辺 O_1O_2 と，光源の頂点 S_1, S_2, \dots, S_m のうちの1頂点を含む各平面 $O_1O_2S_1, O_1O_2S_2, \dots, O_1O_2S_m$ を生成し，その中で遮蔽面とのなす角が最小である面を保存する．この処理を，遮蔽面の各辺 $O_1O_2, O_2O_3, \dots, O_nO_1$ について同様に実行し，各辺ごとに1面ずつ平面を保存する．これらの平面を，平面どうしの交線で切りとることにより，本影空間を表す影多面体が生

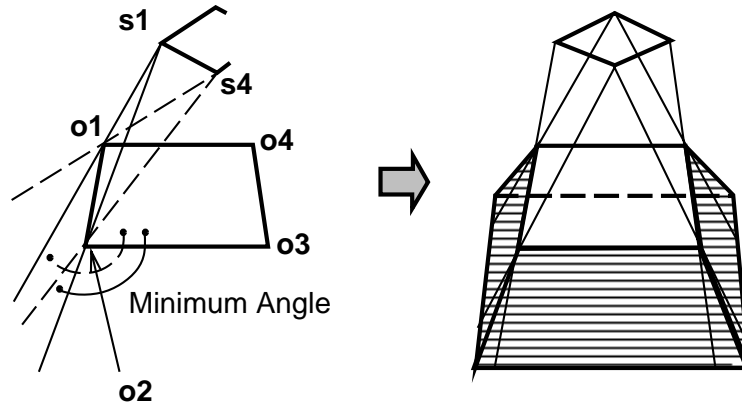


Figure 3.6: 本影空間の生成

成される (図 3.6 参照) .

続いて、半影空間を生成する手法について述べる．本手法では、光源の頂点 S_1, S_2, \dots, S_m のうちの 1 頂点と、遮蔽面の辺 $O_1O_2, O_2O_3, \dots, O_nO_1$ のうちの 1 辺を含む平面を、すべて生成する、同様に、光源の辺 $S_1S_2, S_2S_3, \dots, S_mS_1$ のうちの 1 辺と、遮蔽面の頂点 O_1, O_2, \dots, O_n のうちの 1 頂点を含む平面を、すべて生成する．このようにして生成された平面のうち、遮蔽面とのなす角が最大である平面 F_0 を抽出する．続いて、生成された残りの平面のうち、 F_0 と光源の一頂点および遮蔽面の一頂点を共有しているものを抽出し、抽出された平面の中で F_0 とのなす角が最大である平面 F_1 を選択する．例えば、図 3.7 では、光源の頂点 S_1 と遮蔽面の辺 O_2O_3 を含む面を F_0 とし、続いて選択された面上にある二頂点 S_1 および O_3 を共有する面の中で、 F_0 とのなす角が最大である面として、光源の辺 S_1S_4 と遮蔽面の頂点 O_3 を含む平面が抽出される．以下同様に、抽出された F_i と光源の一頂点および遮蔽面の一頂点を共有しているものを抽出し、抽出された平面の中で F_i とのなす角が最大である平面 F_{i+1} を選択する．この処理を、平面 F_0 が選択されるまで反復することにより、生成された平面群の凸包 (convex hull) を形成することができる．この凸包は、半影空間を表す影多面体である．

3.3.2 受光面のデータ構造

簡略な輝度不連続線法では、辺 (Wireframe) や面 (Surface) を統一的に処理する、非多様体形状モデリング [64] の考え方をういたデータ構造を用いて、入力形状および輝度

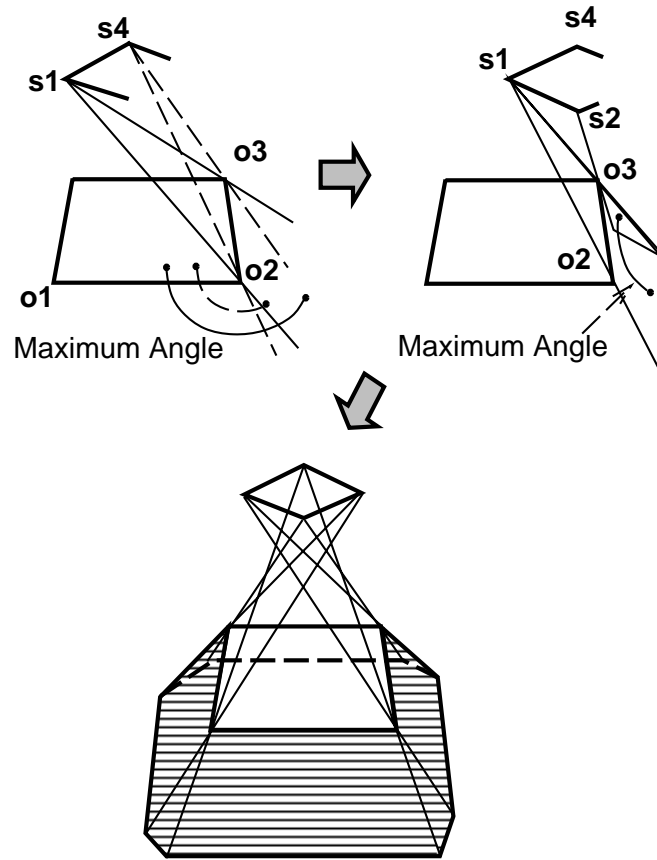


Figure 3.7: 半影空間の生成

不連続線生成後の受光面を表現する。

本手法では、下記の通り定義されたデータ構造を用いて、物体を表現する (図 3.8 参照)。

定義 3.6 入力形状の頂点、および輝度不連続線の両端点を、頂点 (Vertex) と定義する。頂点 V は、座標値 $\{x, y, z\}$ と、 V を端点にもつ辺の集合 $\{E_1, E_2, \dots\}$ より構成される。

定義 3.7 本章では、入力形状の外枠および穴を構成する線分、および輝度不連続線を、辺 (Edge) と定義する。辺 E は、両端の頂点 $\{V_1, V_2\}$ と、 E が所属するループ $\{L_1, L_2\}$ (1 個または 2 個) から構成される。

定義 3.8 本章では、面積を有する領域を囲い込む、頂点によって連結された辺の集合を、ループ (Loop) と定義する。ループ L は、辺の集合 $\{E_1, E_2, \dots\}$ と、各々の辺の方向 (V_1

V_2 , または $V_2 - V_1$ のいずれか) を表す符号 $\{d_1, d_2, \dots\}$ と, L が所属する面 $\{F\}$ より構成される.

定義 3.9 本章では, ループによって囲まれた領域を, 面 (Face) と定義する. 面 F は, 外枠となるループ $\{L_o\}$ と, 穴となるループ $\{L_1, L_2, \dots\}$ より構成される. この時, 外枠となるループは, 面の法線方向からみて半時計回りに定義されている必要がある. 穴となるループは, 面の法線方向からみて時計回りに定義されている必要がある.

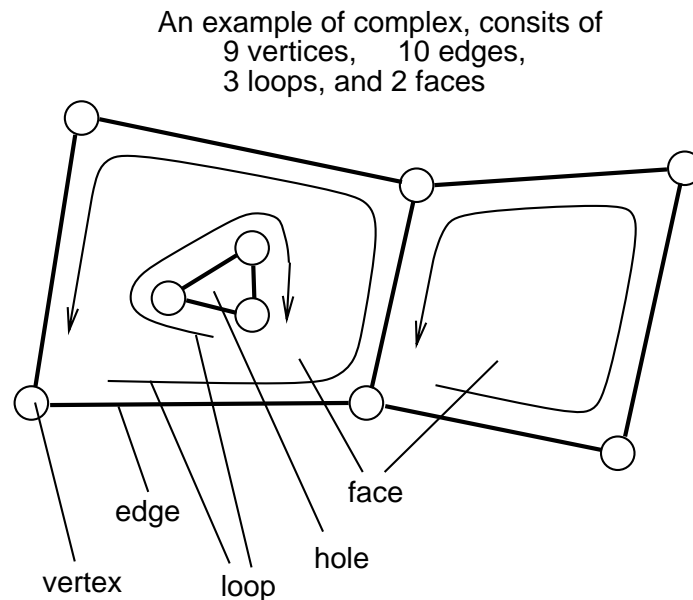


Figure 3.8: 複体 (complex) を用いたデータ構造

本手法では, 輝度不連続線を生成する前に, 物体の形状を複体を用いたデータ構造に変換する. 本手法では, 光源と物体を 1 個ずつ選択して影多面体を生成し, 他の物体に対して下記のような処理を行なって, 輝度不連続線を生成する. 輝度不連続線の生成処理は, 複体を用いたデータ構造上で実行される.

1. 影多面体の各辺と受光面との交点を算出し, その交点を受光面の頂点として追加する.
2. 追加した頂点を結んで辺をつくる. この辺が輝度不連続線に相当する.

3. 新しく生成された辺と，すでに生成されている辺の交点に対して，頂点を追加する．
さらに，追加した頂点で2辺を分割する．
4. 受光面上の，辺によって囲まれる各領域を，ループで表現する．

3.3.3 影多面体との交点計算による輝度不連続線の生成

多くの遮蔽物体によって多くの輝度不連続線が生成される受光面では，輝度不連続線の交点計算に多くの処理量を要する．本手法では，この交点計算を軽減するために，影多面体と受光面との交線計算に，以下のようなアルゴリズムを用いている．

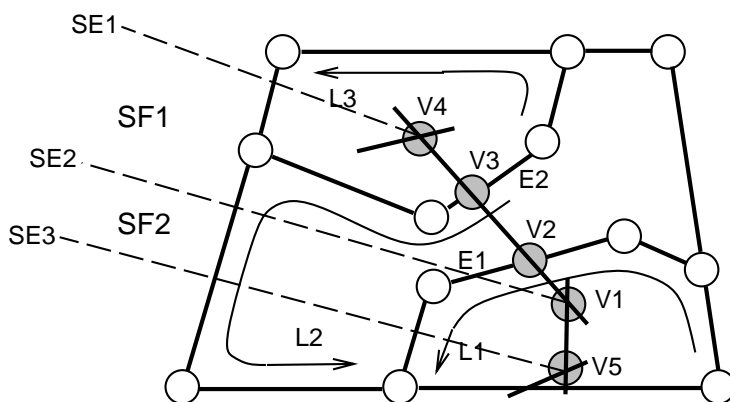


Figure 3.9: 影多面体と受光面の交点計算

1. 影多面体の辺と受光面の交点を1点抽出して，頂点を生成する．頂点が生成できたら，それがどのループ内部に属するかを検出する．図3.9では，影多面体の辺 SE_2 と受光面との交点として，頂点 V_1 が最初に生成される．この頂点は，ループ L_1 の内部に存在する．
2. 当該影多面体の辺を共有する2面について，他辺も同一ループ内部で受光面と交差するか判定する．図3.9では，辺 SE_2 を共有する2面 SF_1 および SF_2 がもつ辺 SE_1 および SE_3 と受光面との交点 V_4 および V_5 が，同一ループ L_1 内部にあるか判定される．

3. 2. で同一ループ内部で受光面と交差しない場合には，影多面体の辺を共有する面と受光面の交線を求め，ループを構成する辺との交差判定を実行する．図 3.9 では，影多面体の面 SF_1 と受光面の交線が求められる．続いて，交線とループ L_1 を構成する各辺との交点計算を実行する．その結果，辺 E_1 との交点 V_2 が生成され，辺 E_1 が V_2 で分割される．
4. 交点をもつ辺を共有するループ上で，2. および 3. を反復する．図 3.9 では，辺 E_1 を共有するループ L_2 について，影多面体の辺 SE_1 と受光面の交点 V_4 の内外判定を実行する．交点 V_4 はループ L_2 の内部にはないので，ループ L_2 を構成する各辺との交点計算を実行する．その結果，辺 E_2 との交点 V_3 が生成され，辺 E_2 が V_3 で分割される．
5. 2. で同一ループ内部で受光面と交差する場合には，頂点を生成し，2 点を結ぶ辺を生成する．その後，新しく生成された頂点に対し，2. および 3. を反復する．図 3.9 では，影多面体の辺 SE_3 と受光面との交差判定が実行され，交点 V_5 が同一ループ L_1 上にあるので，そのまま V_1 および V_5 を結ぶ辺を生成する．

上記のアルゴリズムは，影多面体の辺と受光面との交点計算，および輝度不連続線どうしの交点計算の処理量を低減することが出来る．図 3.9 では，影多面体の辺と受光面との交点 V_1, V_4, V_5 と，受光面上のループとの内外判定の処理回数を低減している．また，新しく生成された輝度不連続線 V_1V_4, V_1V_5 と，既存の辺との交点計算も，ループ L_1 や L_2 に属する辺に制限しているので，交点計算の処理回数を低減している．

3.3.4 ループ処理による受光面の領域分割

本手法では，3.3.3 節で示したアルゴリズムによって，1 個の影多面体との交線が生成されるたびに，受光面上のループを更新している．

ここで，複体における辺とループの関係において，以下のように定義する．

定義 3.10 入力形状の外枠および穴を構成する辺は，1 個のループに所属する．一方，輝度不連続線として受光面の内部に生成された辺は，2 個のループに所属する．

本手法では上記の定義に基づいて，3.3.3 節に示すアルゴリズムで新しく生成されたそれぞれの辺が，新しい 2 個のループに所属するようにループを生成し，不必要になった

ループを消去する．本手法では1個のループを生成するために，まず新しく生成された1辺を選択し，頂点によって連結される隣接辺を左回りまたは右回りに探索し，出発した辺に探索が戻るまで探索を反復する．

以下，ループを生成するアルゴリズムを，図 3.13 を用いて示す．図 3.13(a) は，2個の遮蔽面 O_1 および O_2 によってすでに輝度不連続線が生成された受光面の例である．図 3.13(1) に示すように，受光面には $L_1 \sim L_8$ の8個のループがある．この受光面に，3.3.3節で示したアルゴリズムによって，図 3.13(b) に示すように，遮蔽面 O_3 による輝度不連続線を新しく生成する．この結果，受光面のループは図 3.13(4) の $L_1 \sim L_9$ の9個に修正される．

1. 影多面体との交線計算により，輝度不連続線を生成する．図 3.13(2) では，影多面体の辺との交点が $v_1 - v_6$ の6頂点として生成され，新しい輝度不連続線と既存の輝度不連続線の交点が $v_a - v_d$ の4頂点として生成されている．
2. 新しく生成された辺について，「左側」「右側」の2個の欄をもつ一覧表を用意する．この一覧表は最初は空欄であるが，以後の処理において，新しく生成されたループの番号が記入される．図 3.13(2) では，新しく生成された辺 $v_1v_2, v_2v_3, v_3v_a, \dots$ に対して一覧表を作成している．
3. 一覧表に登録された辺の中で，一覧表に空欄をもつ辺を選択する．図 3.13(2) では，辺 v_1v_2 を選択している．
4. 辺の1頂点を共有する隣接辺を，順に探索する．3辺以上に接する頂点に探索が到達したら，左回りあるいは右回りに，探索済の辺とのなす角が最小である辺を選択して，探索を続行する．また，既存の辺が探索された時には，既存の辺が属する同一方向のループを消去する．図 3.13(2) では， v_1v_2 を出発して， v_2 を共有する辺 v_2v_a を探索している． v_a は4辺に接しているので， v_a を共有する辺の中で，左回りにみて v_2v_a とのなす角が最小である辺を選択している．選択された既存の辺は，図 3.13(1) で L_6 に所属しているので，このループを消去する．
5. 4. を反復し，探索が出発点に戻ったら探索を停止し，生成されたループの番号を，一覧表中の探索された辺の欄に記録する．図 3.13(2) では，探索が v_1 に到達した時

点で，生成されたループ L_4 の番号を，探索された辺 v_1v_2, v_2v_a, v_dv_6 の各辺の欄に記入している．

6. 3. および 4. を反復し，一覧表のすべての空欄が埋まるまでループを生成する．図 3.13(3) では， v_av_2 を選択してループ L_5 が生成され， v_av_3 を選択してループ L_6 が生成されている．
7. 完成したループが左回りであるか右回りであるかを確認する．本手法では，法線方向に対して左回りであるループは領域の外枠を形成し，右回りであるループは領域の穴を形成するものとみなす．図 3.13(4) において，右回りの L_1 は，その外側の L_3 に囲まれる領域の穴として扱われる．左回りの L_2 は， L_2 によって囲まれる領域の外枠として扱われる．

3.3.5 照射状態の判定

3.2 節で述べた通り，本手法では影多面体 1 個と受光面の交線計算を実行し，受光面上のループを更新した時に，ループが囲む領域と影多面体の内外判定を実行する．この内外判定は，受光面上で新しく生成されたループのうち，外枠として扱われるものに対して実行される．内外判定の結果から，ループが囲む領域における，影多面体を形成する光源との照射状態が記録される．

本手法では，各々のループに，各光源との照射状態を表す一覧表を保持する．光源とループの照射状態は，本影領域=3，半影領域=2，照射領域=1，未処理=0 の 4 数で表現される．図 3.10 は，点光源と面光源が 1 個ずつ存在する情景における，照射状態の記録例である．分割された受光面ごとに，光源の数に等しい数の欄を持つ表を用意し，照射領域，半影領域，本影領域のいずれであるかが記録されている．この記録は，受光面の分割後のパッチおよびエレメントの生成時に，パッチおよびエレメントの大きさを決定するために参照される．

3.3.6 受光面の分割とメッシュ生成

本手法では，すべての光源に対して，影多面体と受光面との交線計算によって輝度不連続線を生成した後，ループによって分割された領域ごとにメッシュを生成する．本手法

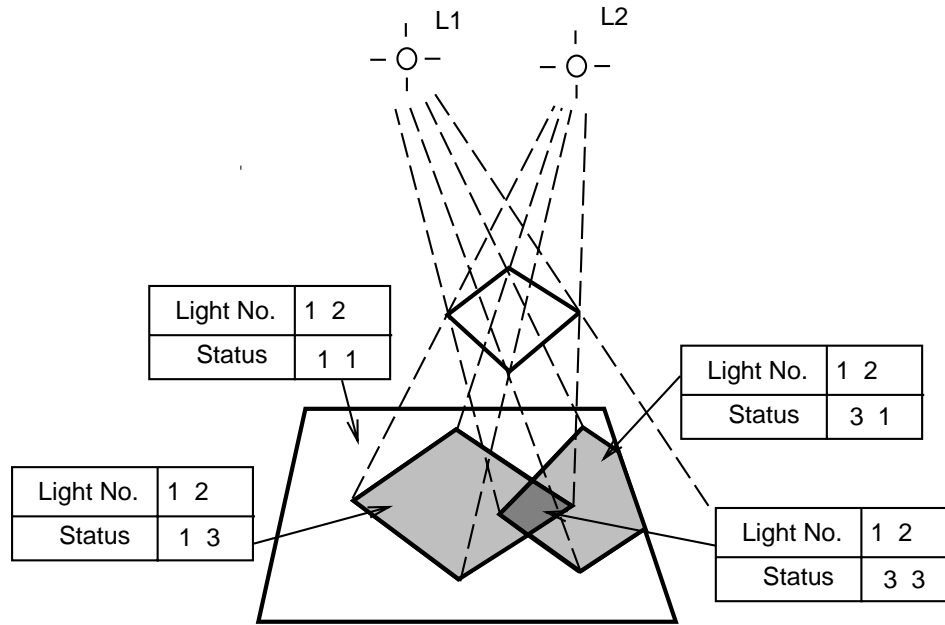


Figure 3.10: 照射状態の判定と記録

では、分割された面ごとにメッシュを生成するので、輝度不連続線を交差するパッチやエレメントは生成されない。

点光源や平行光源によって生成される輝度不連続線は、 D_0 輝度不連続線であり、輝度値が不連続である鮮明な陰影が見られる。このような鮮明な陰影を表現するためには、輝度不連続線の両側のパッチやエレメントには、ノードを個別に与え、独立な輝度値が算出されるようにすることが望ましい。逆に、面光源がつくる輝度不連続線は、 D_1 輝度不連続線であり、なめらかに輝度に変化する陰影が見られる。このような陰影を表現するためには、輝度不連続線の両側のパッチやエレメントには、ノードを共有させ、輝度不連続線上の輝度値を共有させることが望ましい。本手法では、点光源や平行光源によって生成される輝度不連続線の両側の領域には、受光面の分割時に個別の辺と頂点を与え、独立にノードを発生する。面光源によって生成される輝度不連続線の両側の領域には、隣接する辺と頂点を共有させて、輝度不連続線上のノードを共有させる (図 3.11 参照)。

本手法では、受光面を分割した後、以下のアルゴリズムでパッチおよびエレメントを生成する (図 3.14 参照)。

1. パッチおよびエレメントの大きさ (一辺の長さ) L_p および L_e を決定する。各領域

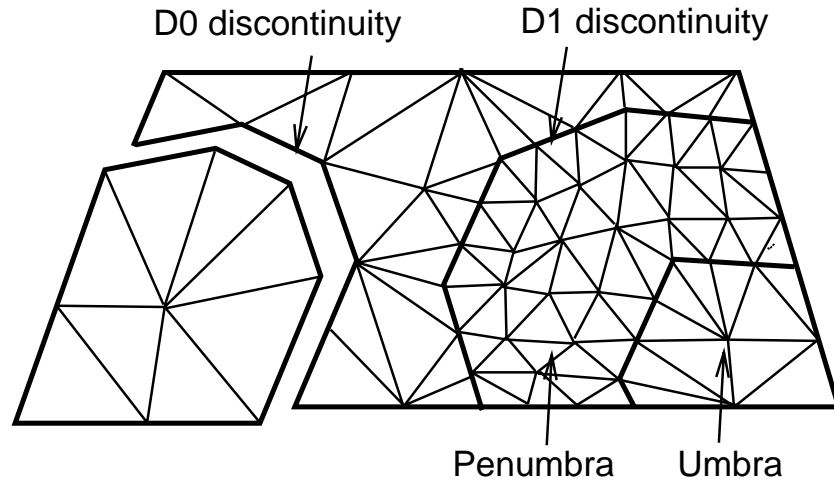


Figure 3.11: 輝度値の連続性を考慮した受光面の分割

ごとに，光源との照射状態の一覧表を参照し，少なくとも1個の光源に対して半影領域である場合には，パッチおよびエレメントをそれ以外の領域よりも小さく設定する．

2. 分割された各領域の頂点に，ノードを与える．
3. 分割された各領域の辺上に，ノードを与える．ノードの間隔は，パッチの大きさ L_p を用いる．辺が2領域に共有されている場合には，両領域に与えられたパッチの大きさのうち，小さい方の値を用いる．
4. 分割された各領域の内部に，ノードを与える．ノードの間隔は，パッチの大きさ L_p を用いる．
5. デローニー三角メッシュ生成法を用いてノードを連結し，三角メッシュを生成する．三角メッシュはパッチの集合として保存される．
6. パッチをさらに分割して，エレメントを生成する．この再分割は，デローニー三角メッシュ生成法を用いてもよいが，2.3節に示した，パターンマッピングや再帰的分割のような単純な手法で十分な場合も多い．処理速度を重視するならば，それらの単純な手法の方が有利である．

本手法では、パッチの歪みを低減するために、上記アルゴリズムの 3. および 4. において、バブル・メッシュ法 [53] などのノード配置手法を用いることが望ましい。

3.4 実行例

本節では、本手法を実装して実行した結果を示す。なお、計算機は IBM PowerStation RS/6000 (Model 560) を用いた。

図 3.15 は、直方体の内部に長方形の面光源 1 個と浮遊する長方形 2 個をもつ単純な入力形状に対して、輝度不連続線を生成し、受光面を分割した例である。図 3.16 は、図 3.15 に示される輝度不連続線生成結果を用いて、パッチおよびエレメントを生成し、ラジオシティ法を実行して生成された画像である。

図 3.17 は、図 3.15 に示される輝度不連続線生成結果のうち的一面、底面に位置する面に対してメッシュを生成した例である。エレメントの重心につけられたマークは、赤、黄、青が、それぞれ照射領域、半影領域、本影領域に属することを示している。このマークより、半影領域内部においては、エレメントが小さく生成されていることがわかる。

図 3.19 は、180 枚の多角形で近似された球状物体と、非常に小さい面光源をもつ入力形状に対して、輝度不連続線を生成し、パッチおよびエレメントを生成し、ラジオシティ法を実行して生成された画像である。

続いて、本手法を適用したラジオシティ法の計算時間等について考察する。表 3.1 は、本手法を用いたメッシュ生成結果と、輝度不連続線、メッシュ生成、ラジオシティ法の各処理における処理時間をまとめたものである。ここで、実行例 1 および実行例 2 は、それぞれ図 3.16、図 3.19 の生成に用いたデータである。

ここで、表中のそれぞれの変数は、

N_{f1} : ループ分割前の受光面の数

N_{f2} : ループ分割後の受光面の数

N_d : 輝度不連続線の数

N_p : パッチの総数

N_{e1} : エレメントの総数 (半影領域のみ細かく生成)

Table 3.1: 輝度不連続線生成，メッシュ生成，ラジオシティ法の処理時間

	実行例 1	実行例 2
N_{f1}	8	721
N_{f2}	20	2495
N_d	40	3116
N_p	486	6030
N_{e1}	3485	11230
N_{e2}	6090	17150
$T_d(\text{sec.})$	0.12	166.58
$T_m(\text{sec.})$	1.77	15.28
$T_{rs}(\text{sec.})$	156.63	289.02

N_{e2} : エレメントの総数 (均一に生成)

T_d : 輝度不連続生成に要した処理時間

T_m : メッシュ生成に要した処理時間

T_{rs} : ラジオシティ法実行に要した処理時間

を示している。

表 3.1 に示された輝度不連続線の総数から，本手法は BSP 法に比べ，受光面の分割数が少ないことが推定される．なぜならば，BSP 法による受光面の分割数は，少なくとも $N_{f1} + N_d$ 以上であり，本手法による受光面の分割数 N_{f2} はそれよりも少ないからである．また，表 3.1 において， N_{e1} は N_{e2} よりも非常に小さくなっていることから，本手法はエレメント数の増加を抑えていることがわかる．

続いて，本手法による輝度不連続線生成の処理時間について考察する．表 3.2 は，輝度不連続線生成の各処理の処理時間をまとめたものである．

ここで，表中のそれぞれの変数は

T_{sdw} : 影多面体生成部の処理時間

T_{isr} : 影多面体と受光面の交点計算部の処理時間

T_{idd} : 輝度不連続線同士の交点計算部の処理時間

Table 3.2: 輝度不連続線生成部の計算時間

	実行例 1	実行例 2
T_{sdw} (sec.)	0.03	0.21
T_{isr} (sec.)	0.04	154.48
T_{idd} (sec.)	0.02	3.76
T_{lop} (sec.)	0.01	2.13
T_{lit} (sec.)	0.01	5.50
T_{div} (sec.)	0.01	0.50

T_{lop} : ループ生成部の処理時間

T_{lit} : 光源との照射状態の記録部の処理時間

T_{div} : 受光面の分割部の処理時間

を示している。

入力形状である多角形数を n とすると、 T_{isr} は影多面体数と受光面数の双方に比例するため、その処理量は $O(n^2)$ であると推測される。表 3.2 の結果から見ても、データ量の増大とともに、 T_{isr} が非常に大きくなっている。本手法は従来の輝度不連続線法よりも輝度不連続線数を低減しているため、従来の手法よりも T_{isr} を低減しているが、さらにこの処理量を低減することが今後の課題につながると思われる。

また、BSP 法では必要以上に長い輝度不連続線を生成するため、物体が多くなった時に、輝度不連続線同士の交点計算に要する処理量 T_{idd} が飛躍的に大きくなることが予想される。本論文では、BSP 法を用いた従来の手法との処理速度の測定比較をしていないので、これを測定して本手法における T_{idd} の値がどの程度減少したのかを評価することも、今後の課題につながるものと思われる。

最後に、2.4.3 節で述べた通り、点光源や非常に小さい面光源がつくる急激な輝度変化の表現において、輝度不連続線法が適応的再分割法よりも有利である例を示す。図 3.20 は、多角形の集合で表現された椅子のような形状と、非常に小さい面光源をもつ入力形状に対して、輝度不連続線を生成し、パッチおよびエレメントを生成した結果である。図 3.21 は、同様な入力形状に対して、適応的再分割法を適用してパッチおよびエレメントを生成した結果である。図 3.22 および図 3.23 は、それぞれ 図 3.20 および図 3.21 に示さ

れるメッシュを用いてラジオシティ法を実行し、画像を生成した例である。図 3.22 の方が、椅子のような物体の影を鮮明に表現しており、輝度不連続線法の優位性が見られる。しかし、図 3.22 では細長いエレメントがいくつか生成されており、フォーム・ファクタの算出の精度をあげる必要があった。

3.5 むすび

本章では、影多面体と受光面との交線を求めることで、半影領域と本影領域の境界線、および半影領域と照射領域の境界線のみを輝度不連続線として生成する、簡略な輝度不連続線法を提案した。本手法は、輝度不連続線の生成数を減少し、また BSP 法を用いないので、輝度不連続線の交点計算の計算量を減少すると同時に、非常に小さいエレメントや細長いエレメントの生成を低減することが出来る。また、半影領域のみにパッチおよびエレメントを細かく生成し、他の領域では半影領域よりもパッチおよびエレメントを大きく生成することで、パッチ数およびエレメント数の増加を抑えながら、写實的に陰影を表現することができる。

付録

(定理 3.1 の証明)

平行光源がつくる輝度不連続線について考える。輝度不連続線上の任意の点 P_d と、輝度不連続線上にある遮蔽面上の 2 点 O_i, O_j 、および P_d から平行光源の光線方向と反対方向にのばした直線上の無限遠点 S は、同一な輝度不連続面上にある。この時、 P_d から見て辺 $O_i O_j$ と点 S は重なって見える。 P_d を輝度不連続線上でない方向に微小距離だけ移動すると、 S が遮蔽面に遮蔽されるか、 S がまったく遮蔽面に遮蔽されない状態になる (図 3.12 (a) 参照)。よって P_d は、微小距離だけ移動することで照射領域または本影領域に属する位置に存在する。これは、照射領域と本影領域の境界上を意味する。

点光源についても、点光源の位置を S とすることで、同様に説明できる。

(定理 3.2 の証明)

光源の頂点 S_m と遮蔽面の辺 $O_i O_j$ がつくる輝度不連続線について考える。輝度不連続線上の任意の点 P_d と、遮蔽面上の 2 点 O_i, O_j 、および光源の頂点 S_m は、同一な輝度不連

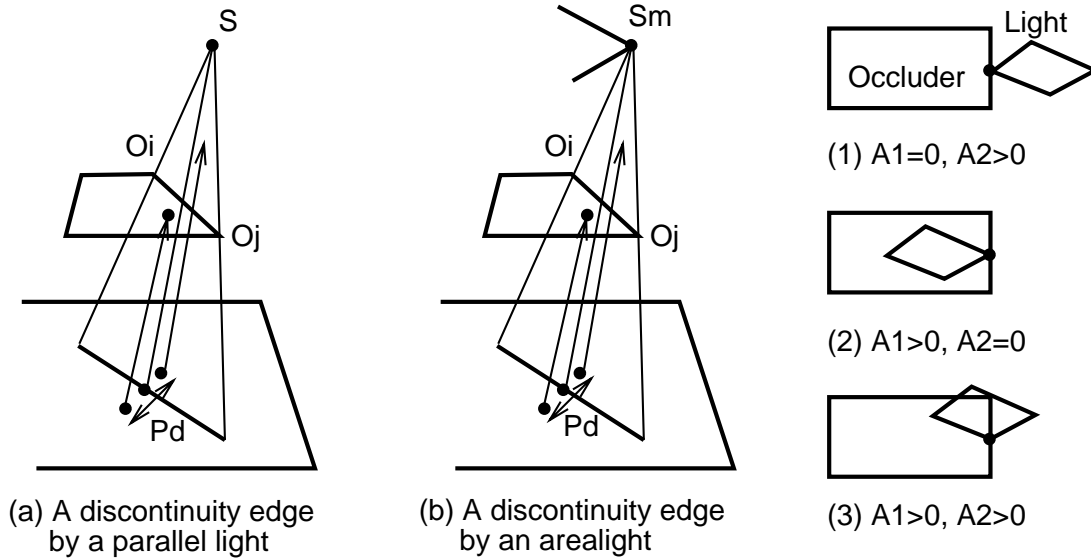


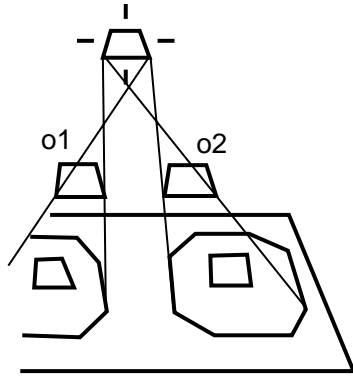
Figure 3.12: 輝度不連続線の位置に関する証明

続面上にある．この時， P_d から見て辺 O_iO_j と点 S_m は重なって見える．光源が遮蔽面によって遮蔽されている領域の面積を A_1 ，遮蔽されていない領域の面積を A_2 とすると，

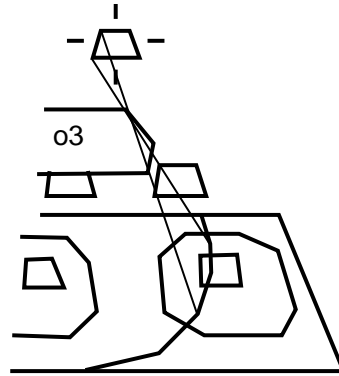
- (1) $A_1 = 0, A_2 > 0$
- (2) $A_1 > 0, A_2 = 0$
- (3) $A_1 > 0, A_2 > 0$

の3通りに場合わけをすることが出来る (図 3.12 (b) 参照) . (1) の場合， P_d を輝度不連続線上でない方向に微小距離だけ移動すると， S_m が遮蔽されて $A_1 > 0$ になるか， S_m が遮蔽されなくなって光源がまったく遮蔽面に遮蔽されない状態になる．この時， P_d は微小距離だけ移動することで照射領域または半影領域に属する位置に存在する．これは， P_d が照射領域と半影領域の境界上に存在することを意味する．同様にして，(2) の場合， P_d は微小距離だけ移動することで本影領域または半影領域に属する位置に存在する．これは， P_d が本影領域と半影領域の境界上に存在することを意味する．(3) の場合， P_d を輝度不連続線上でない方向に微小距離だけ移動しても， $A_1 > 0$ かつ $A_2 > 0$ であることに変わりはない．この時， P_d は微小距離だけ移動しても半影領域に属する位置に存在する．これは， P_d が半影領域の内部に存在することを意味する．

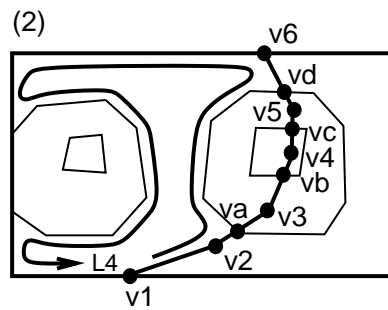
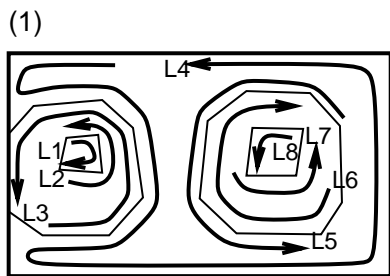
光源の辺 $S_m S_n$ と遮蔽面の頂点 O_i が作る輝度不連続線についても，同様に説明できる．



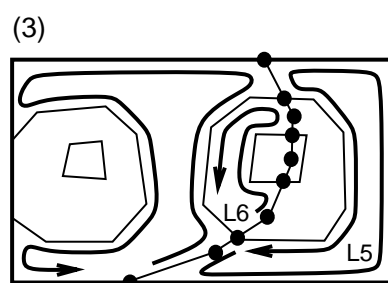
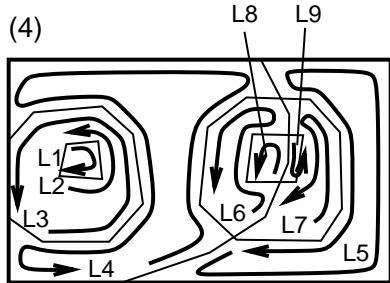
(a) Before the discontinuity edges generation



(b) After the discontinuity edges generation



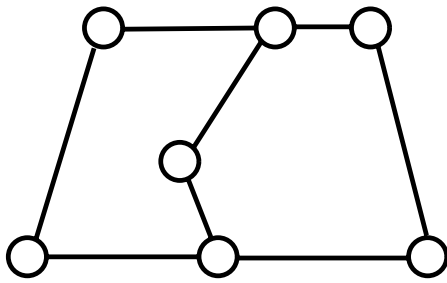
	v1v2	v2va	vav3	...	vdv6
left	L4	L4			L4
right					



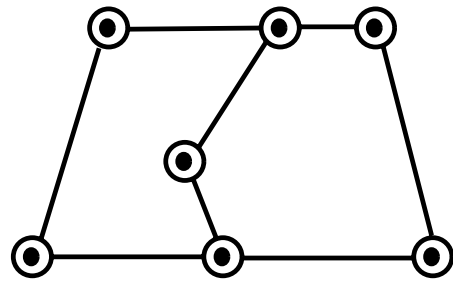
	v1v2	v2va	vav3	...	vdv6
left	L4	L4	L6	...	L4
right	L5	L5	L7	...	L5

	v1v2	v2va	vav3	...	vdv6
left	L4	L4	L6		L4
right	L5	L5			L5

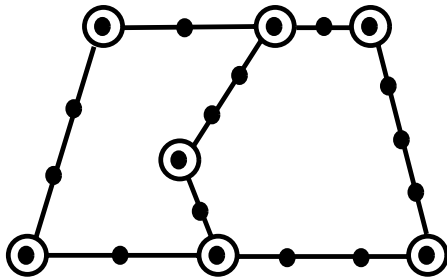
Figure 3.13: 輝度不連続線ループの生成



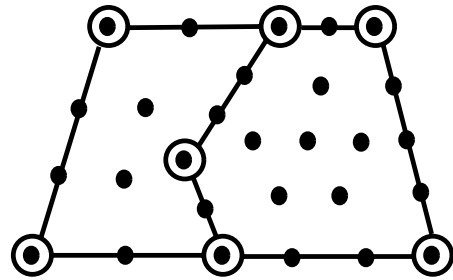
(1) A Polygon with discontinuity edges



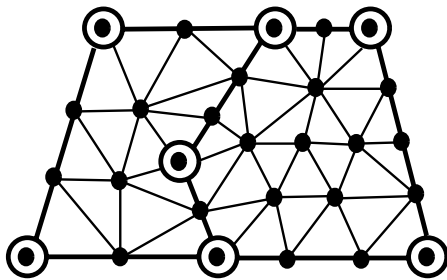
(2) Node location on vertices



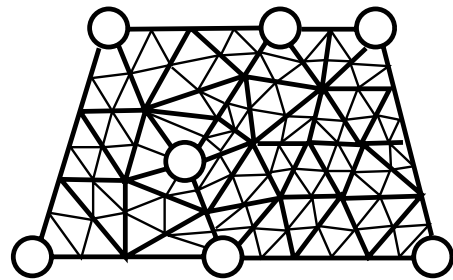
(3) Node location on edges



(4) Node location inside faces



(5) Patch generation by Delaunay triangulation



(6) Element generation

Figure 3.14: 本手法におけるパッチ/エレメントの生成

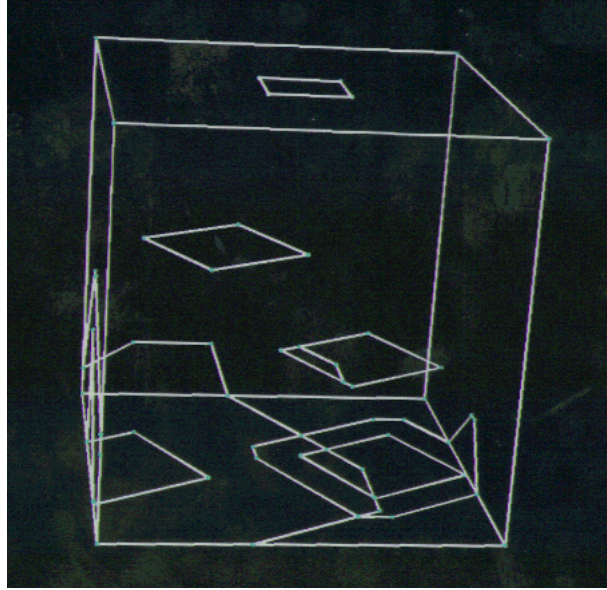


Figure 3.15: 輝度不連続線生成結果 (a)

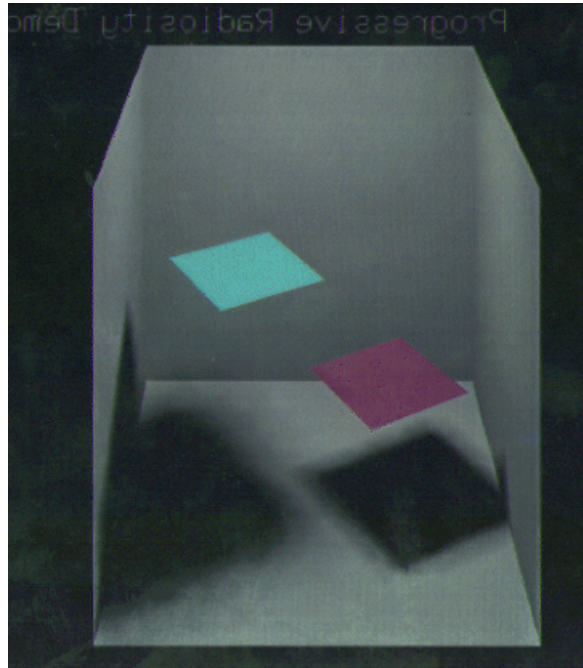


Figure 3.16: レンダリング結果 (a)

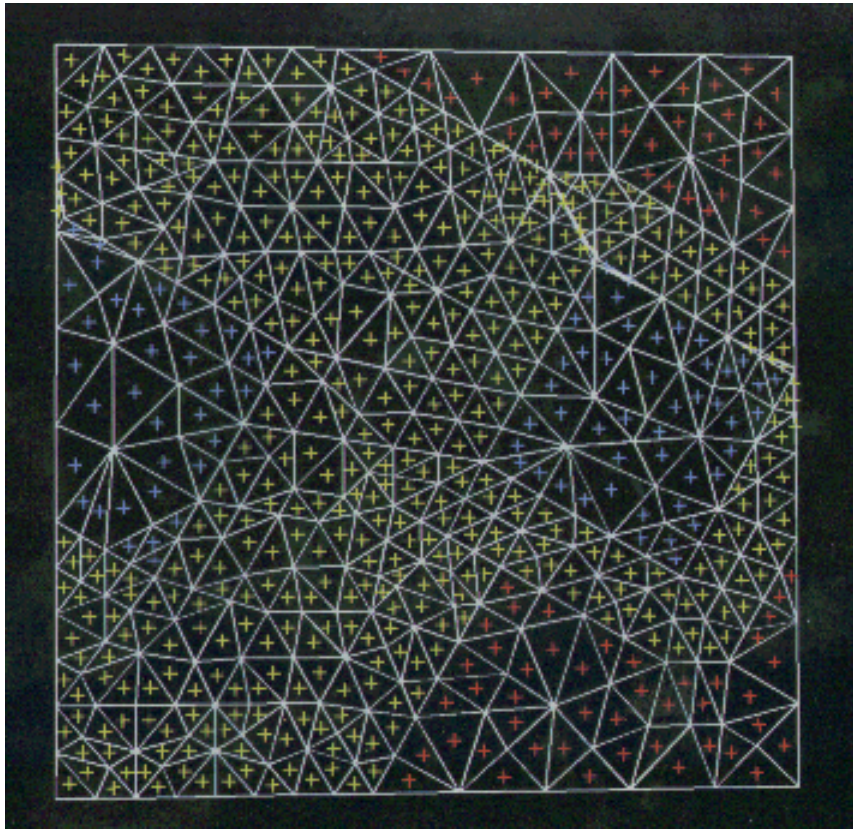


Figure 3.17: メッシュ生成結果 (a)

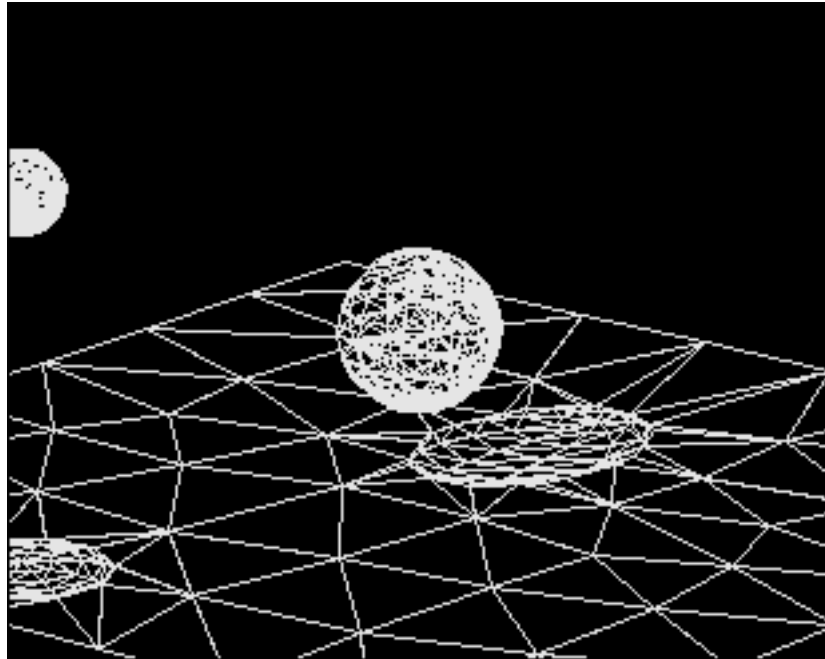


Figure 3.18: メッシュ生成結果 (b)

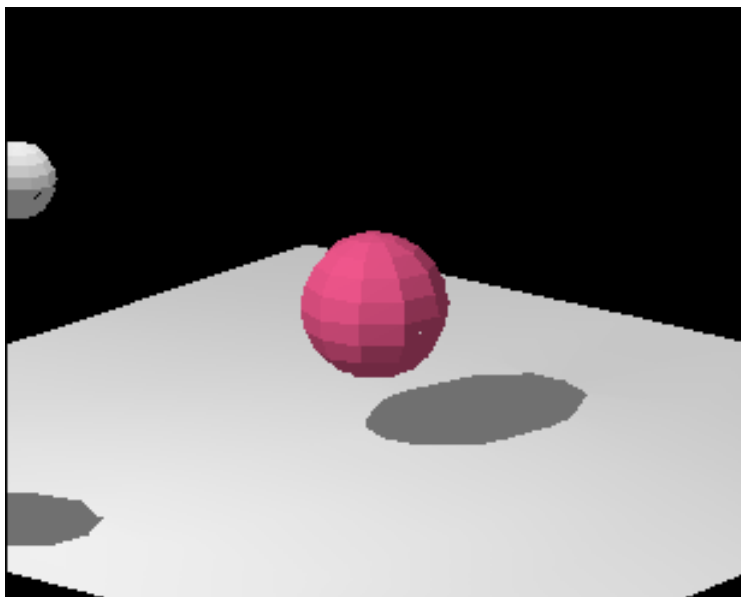


Figure 3.19: レンダリング結果 (b)

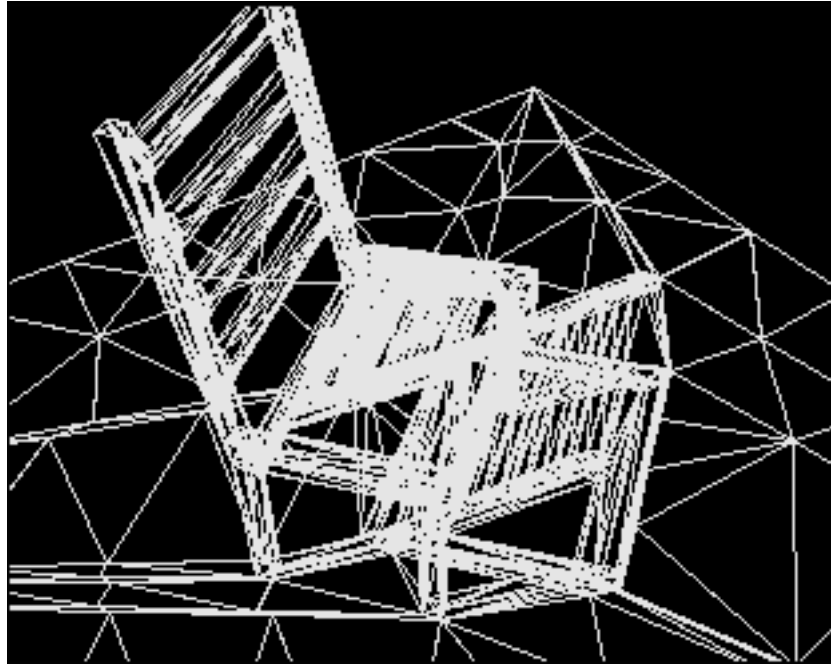


Figure 3.20: メッシュ生成結果 (c)

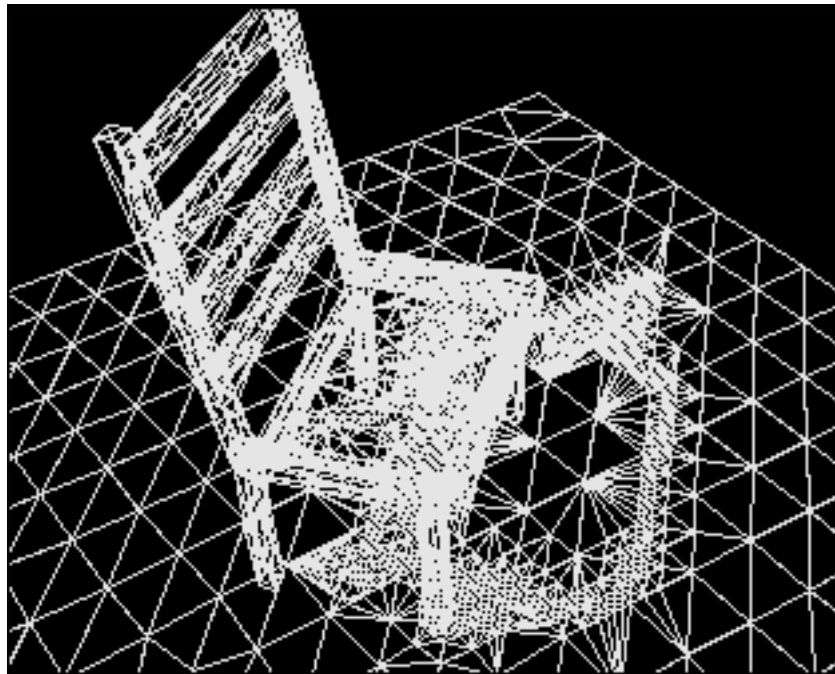


Figure 3.21: メッシュ生成結果 (d)



Figure 3.22: レンダリング結果 (c)



Figure 3.23: レンダリング結果 (d)

第 4 章

輝度分布参照型バブル・メッシュ法

4.1 はじめに

第2章では、ラジオシティ法のための前処理型適応的メッシュ生成、および後処理型適応的メッシュ生成について、従来の研究成果を概観した。

ラジオシティ法のための後処理型適応的メッシュ生成の代表的な手法に、適応的再分割法 [4] がある。2.4.2 節で述べた通り、従来の適応的再分割法には、

- 要素の再分割の条件として、隣接エレメントや隣接ノードの輝度差をパラメータ入力するケースが多い。輝度差をパラメータとする場合、不慣れな利用者には試行錯誤を強いられるケースが多い。
- 1回目のラジオシティ法に用いる要素が小さいと、2回目のラジオシティ法における要素数が非常に多くなる可能性が高くなる。
- 逆に1回目のラジオシティ法に用いる要素が大きいと、要素数の増加を抑えることができる反面、小さな陰影の表現を欠落する可能性が高くなる。

などの問題点がある。

本章では、上記の問題点を解決するために、1回目のラジオシティ法の算出結果から、輝度変化の大きい部位では要素サイズを小さく、輝度変化の小さい部位では要素サイズを大きく設定してメッシュを再生成する、輝度分布参照型バブル・メッシュ法 [28] を提案する。

4.2 節では、本手法の概要について述べる。本手法では、まず一様な要素サイズでメッシュを生成し、1回目のラジオシティ法を実行し、算出結果をノードごとに保持する。続いて、輝度差が大きい部位においては要素サイズ値を小さく、輝度差が小さい部位においては要素サイズ値を大きくするような、要素サイズを算出する関数を定義し、ノードごとに要素サイズ値を算出する。続いて、入力形状領域を覆う直交格子形状を生成し、要素サイズ値の関数を用いて、直交格子形状の格子点における要素サイズ値を算出する。この直交格子形状を入力形状領域と一緒に入力して、バブル・メッシュ法 [53] を実行することで、適応的にメッシュを生成することが出来る。

4.3 節では、本手法の各処理部について示す。具体的には、ノードの位置における輝度変化の大きさから要素サイズ値を算出する関数の定義、直交格子形状の格子点における要素サイズ値の算出、直交格子形状を入力したバブル・メッシュ法の各処理について示す。

4.2 ラジオシティ法のための輝度分布参照型バブル・メッシュ法の概要

2.4.2 節では、適応的再分割法 [4] の最も大きな問題点として、要素数の増加と、小さな陰影の欠落の問題点を示した。これらの問題を解決する一手段として、下記の要件を満たす手法を実現する必要がある。しかし、この要件を満たすような手法は、まだ十分に議論されていない。

要件 4.1 小さな陰影を欠落せずに写實的に表現するためには、メッシュを細かく生成して1回目のラジオシティ法を実行する必要がある。この時、1回目のラジオシティ法の実行結果を参照して、輝度変化の小さい部位の要素を大きく再生成することができれば、小さな陰影の欠落を防ぎ、かつ要素数の増加を抑えることができる。

本章では、2.3.3 節で示したバブル・メッシュ法 [53] を用いて、下記のアルゴリズムにより要件 4.1 を満たすような適応的メッシュ生成を実現する手法を提案する。

1. 一様な要素サイズでパッチ/エレメントを生成する。
2. 1回目のラジオシティ法を実行し、輝度値をノードごとに算出する。
3. ノードごとに、輝度変化の大きさを算出し、要素サイズ値を算出する。
4. 入力形状内部の多数のサンプリング点において、要素サイズ値を算出する。
5. 要素サイズ値を入力したバブル・メッシュ法により、パッチおよびエレメントを生成する。
6. 2回目のラジオシティ法を実行する。

本手法では、輝度変化の大きい部位において要素を細かく、輝度変化の小さい部位において要素を大きく生成するように、要素サイズ値の関数を算出する。続いて、入力形状ごとに、形状領域を覆う直交格子形状を生成し、その格子点における要素サイズ値のサンプリング値を算出する。直交格子形状と入力形状をあわせて入力したバブル・メッシュ法により、輝度変化の大きい部位では小さい要素を、輝度変化の小さい部位では大きい要素を生成する。本論文では、輝度変化の値から算出された要素サイズ値をもつ直交格子形状用い

るバブル・メッシュ法を，輝度分布参照型バブル・メッシュ法 (Intensity-based bubble mesh) と呼ぶ．

本手法では，下記の定義にしたがって，輝度変化の大きい部位において要素を細かく，輝度変化の小さい部位において要素を大きく生成するように，要素サイズ値の関数を定義する．

定義 4.1 本章では，ノード N とエレメントの辺によって連結されるノード N_a を，隣接ノードと呼ぶ．

定義 4.2 本章では，下記の式 4.1 を用いて，ノードの位置における輝度変化 D を表すものとする．ここで， n は隣接ノード数， I および I_i は当該ノードおよび i 番目の隣接ノードの輝度値， $dist_i$ は当該ノードと i 番目の隣接ノードの距離である．

$$D = \max\left(\frac{|I_i - I|}{dist_i}\right), (i = 1, 2, \dots, n) \quad (4.1)$$

定義 4.3 本章では，ノードの位置における要素サイズ値を算出するために， D が大きい位置においては小さい値を， D が小さい部位においては大きい値をもつような， D の関数 $f(D)$ を定義する，(図 4.1 参照)．

本手法では， D が小さい部位における要素サイズ値を，1 回目の一様なメッシュ生成における要素サイズ値よりも大きく設定することで，2 回目のメッシュ生成においてエレメント数を減少することが可能になる．

続いて，ノードごとに与えられた要素サイズ値を用いて，入力形状領域内部の多数のサンプリング点における要素サイズ値を算出する．本論文では，下記の定義にしたがってサンプリング点を設定する (図 4.2 (a) 参照)．

定義 4.4 本論文では，入力形状である各々の多角形に対し，多角形が xy 平面に含まれるような直交座標系を設定し，その入力形状を覆うように， x 軸方向に m 個， y 軸方向に n 個の，合計 mn 個の格子点からなる直交格子形状を生成する (図 4.2 (a) 参照)．要素サイズのサンプリング値は，その各々の格子点に代入されるものとする．

要素サイズ値の特性は，要素形状の歪みに大きな影響を与える．本論文では，要素形状の歪みが少ないメッシュを生成するために，以下の定理を参考にして，直交格子形状の格子点における要素サイズ値を算出するアルゴリズムを構築する (定理の証明は本章の付録を参照)．

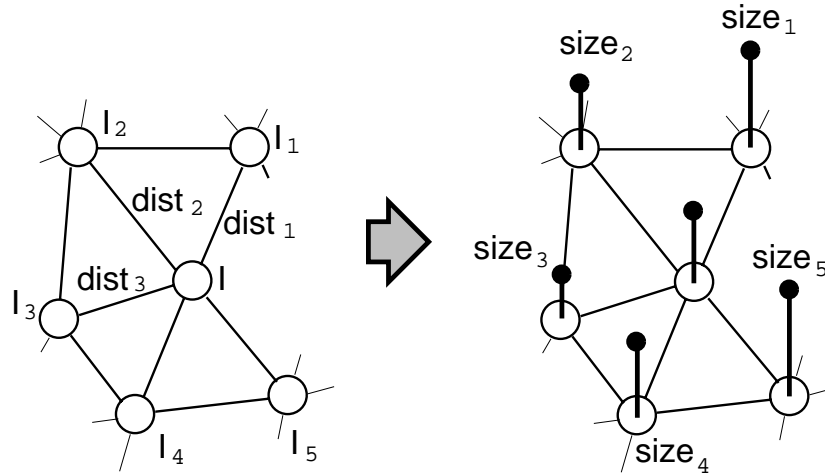


Figure 4.1: ノードにおける要素サイズ値の算出

定理 4.1 要素サイズ値の変化が急激であるほど、要素形状が細長く歪む可能性が高い。

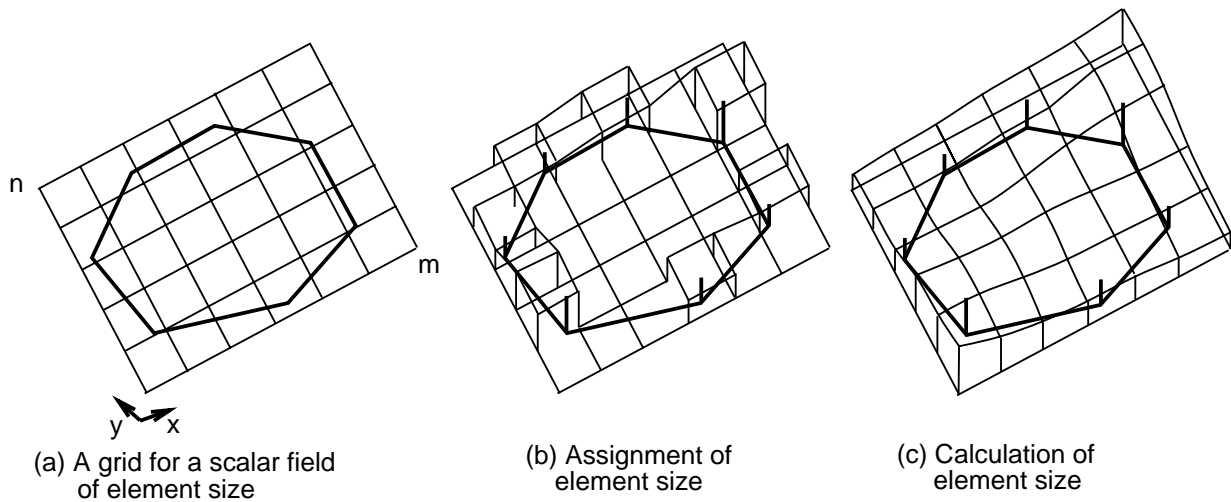


Figure 4.2: 要素サイズのスカラ場を表す直交格子形状

本手法では、直交格子形状のいくつかの格子点に要素サイズ値を与え、その要素サイズ値を格子点の z 座標値と仮想する (図 4.2 (b) 参照)。ここで、定理 4.1 により、要素サイズ値はなめらかに変化することが望ましいといえる。この時、なめらかに数値が変化する要素サイズ値を算出することは、図 4.2 (c) に示すような、 z 座標値の勾配の緩やかな

直交格子形状を生成することと同等である．ここで，直交格子形状の z 座標値を与えるために，下記の仮定を用いる．

仮定 4.1 z 座標値の勾配が最も緩やかな直交格子形状を生成する一手段は，直交格子形状を金属等の薄板と仮定し，それを変形する過程の歪みエネルギーが最小となる状態の z 座標値を算出することである．

本論文では上記の仮定に基づいて，薄板の歪みエネルギーが最小となる状態を有限要素法を用いて求めることにより，格子点における要素サイズ値を算出する．この手法ではまず， xy 平面上に直交格子形状を配置し，ノードにおける要素サイズ値を周囲の格子点に割りつける．続いて，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，割りつけられた要素サイズ値を z 座標値と仮想する．続いて有限要素法を用いて，要素サイズ値が割りつけられなかった格子点における z 座標値を算出する．格子点に算出された z 座標値を要素サイズ値とみなすことで，なめらかに変化する要素サイズのサンプリング値が得られる．

2.3.3 節で示したバブル・メッシュ法 [53] では，要素サイズのサンプリング値を入力して，適応的にメッシュを生成することが出来る．要素サイズのサンプリング値を入力するバブル・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位置における要素サイズ値を参照し，その値を移動位置における球状物体の直径とする．球状物体の半径が一定でなくなる場合にも，バブル・メッシュ法では球状物体が最密充填になる状態を求め，デローニー三角メッシュ生成法によって三角メッシュを生成する (図 4.7 参照)．球状物体の最密充填時における三角メッシュの要素の辺の長さは，要素頂点における球状物体の半径の和となり，この値は連続的に変化する．この手法は，1 回目のメッシュのトポロジーをまったく参照せずに新しいメッシュを生成するという点で， h 法， p 法， r 法のいずれとも異なる手法である．

本手法では，以上の処理をすべての入力形状領域に対して実行した後に，ラジオシティ法を再実行する．図 4.3 は，本手法のアルゴリズムを，擬似コードで示したものである．なお，本手法ではバブル・メッシュ法を用いてパッチを生成し，そのパッチを再分割してエレメントを生成している．

```

void RadiosityWithAdaptiveMeshing() {
    Generate patches and elements by uniform meshing;
    Radiosity calculation;
    for( each node ) {
        Calculate the difference of intensity;
    }
    for( each node ) {
        Calculate the element size;
    }
    for( each polygon ) {
        Define a rectangular grid;
        Calculate element size at each grid-point by sheet-forming analysis;
        Generate patches by bubble mesh method;
        Generate elements by dividing patches;
    }
    Radiosity calculation;
}

```

Figure 4.3: 輝度分布参照型バブル・メッシュ法の擬似コード

4.3 ラジオシティ法のための輝度分布参照型バブル・メッシュ法の各処理

本節では、要素サイズの算出、バブル・メッシュを用いたパッチやエレメントの生成の各処理について述べる。

4.3.1 輝度変化から要素サイズ値への変換式

本論文では、2.2.3 節で示したデータ構造を用いているので、ラジオシティ法の算出結果である輝度値はノードごとに保持される。そこで、それぞれのノードに与えられた輝度値から、ノードごとに要素サイズ値を算出する。まず、式 4.1 を用いて、ノードごとに輝度変化 D の値を求める。通常では輝度値は RGB の 3 成分で表現されるので、本手法では D の値を RGB 別に算出し、その最大値を D として扱う。すべてのノードに対して D が算出されたら、その最小値 D_{min} および最大値 D_{max} を求める。

続いて, D_{min} および D_{max} を用いて,

$$\begin{aligned} size_{min} &= f(D_{min}), \\ size_{max} &= f(D_{max}) \end{aligned} \quad (4.2)$$

を満たす関数 $f(D)$ を定義する．ここで, $size_{min}$ および $size_{max}$ は, 単位距離あたり輝度差の最大値 D 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として用いられる．この時, $size_{min}$ および $size_{max}$ の値の大小関係は,

$$size_{max} < size_{initial} < size_{min} \quad (4.3)$$

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要素サイズ値である．本手法では, 上記の要素サイズの最小値 $size_{min}$ および最大値 $size_{max}$ をパラメータとして用い, 従来の適応的再分割法に用いている輝度差などのパラメータを必要としない．

本章では, $f(D)$ を一次関数として実装した．この場合には $f(D)$ は

$$\begin{aligned} f(D) &= aD + b, \\ a &= \frac{size_{max} - size_{min}}{D_{max} - D_{min}}, \\ b &= \frac{size_{min}D_{max} - size_{max}D_{min}}{D_{max} - D_{min}} \end{aligned} \quad (4.4)$$

と表される．

4.3.2 要素サイズのサンプリング値の算出

本手法では, 関数 $f(D)$ が定義されたら, 各ノードの位置における要素サイズ値を算出する．続いて, 入力領域 (ポリゴン) を覆う直交格子形状を生成し, 各ノードにおける要素サイズ値を, ノードの近傍にある格子点に割り付ける．

図 4.4 は, 要素サイズ値の格子点への割り付けを示したものである．ここで, 格子点 G_{mn} に割り付けられる要素サイズ値 f_{mn} の算出式を導出する．格子点 G_{mn} には, 4 個の長方形格子 $C_{(m-1)(n-1)}, C_{(m-1)n}, C_{m(n-1)}, C_{mn}$ に隣接している．この 4 個の長方形格子の内部に存在するノードを N_i ($i = 1, 2, \dots$), ノード N_i における要素サイズ値 $f(D)$ を F_i ($i = 1, 2, \dots$) とする．格子点 G_{mn} に隣接している 4 個の長方形格子の内部に, ノードが

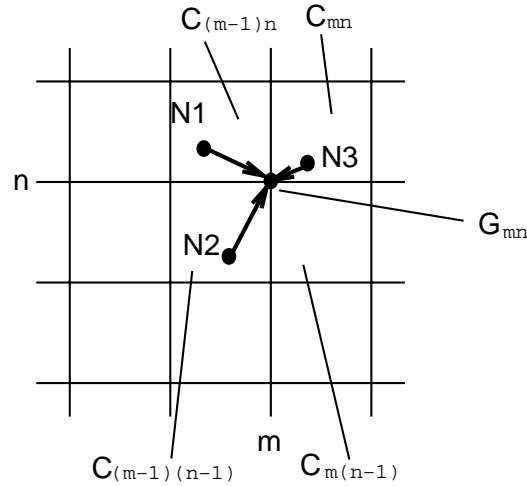


Figure 4.4: 指定された要素サイズの格子点への割り付け

存在すれば，以下の式を使って要素サイズ値 f_{mn} を求める．この式は，格子点に近いサンプリング点における要素サイズ値に重みをつけている．

$$f_{mn} = \frac{\sum_i w_i F_i}{\sum_i w_i}, w_i = \frac{1}{dist_i} \quad (4.5)$$

ただし $dist_i$ は，2点 G_{mn} , N_i 間の距離を表す． $dist_i = 0$ である場合には，十分小さな正数で代用する．

格子点に隣接する4個の長方形格子内にノードが存在しない場合，格子点 G_{mn} には要素サイズ値を割り付けられない．本手法では，前節で述べた通り，金属等の薄板の成形過程の歪みエネルギーが最小となる状態を算出する定式を適用して，要素サイズ値が割り付けられなかった格子点における要素サイズ値を算出する．

本手法では，直交格子形状の辺の方向をそれぞれ x 軸方向， y 軸方向とする2次元座標系を用いる．一般的に，板状物体の変形によって生じる歪みエネルギー U は，下記の式で表されることが知られている [73]．

$$U = \frac{1}{2} \int_{\Omega} (\varepsilon D \varepsilon^T) dx dy \quad (4.6)$$

ここで， \int_{Ω} は，領域全体にわたる積分を表している．また， ε は歪みを表すベクタであり， D は板状物体の特性を表す弾性マトリクスという定数であり，それぞれ以下の式で

与えられる．

$$\varepsilon = \left[-\frac{\partial^2 w}{\partial x^2}, -\frac{\partial^2 w}{\partial y^2}, 2\frac{\partial^2 w}{\partial x \partial y} \right],$$

$$D = \frac{Eh^3}{12(1-\nu^2)} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & (1-\nu)/2 \end{bmatrix}.$$

ここで， h, ν, E はそれぞれ，領域を覆う平面形状の板厚，ポアソン比，ヤング率を表す．また， w は板状物体のたわみ（=荷重によって生じる変位）の大きさを表す．ここで，長方形格子の格子点における x 座標値および y 座標値は変化しないと仮定すると，長方形格子の頂点 (x_{mn}, y_{mn}, z_{mn}) における変位 w_{mn} は， $w_{mn} = z_{mn}$ である．

本手法では，いくつかの格子点に割り付けた要素サイズ値を境界条件として，歪みエネルギー U を最小にするように，有限要素法を用いて格子点の z 方向の変位 w を求める．

ここで，歪みエネルギー U に関する式 4.6 では，領域全体の積分を要するので，長方形格子内部における変位の大きさを表現する数式が必要である．本手法では，それぞれの長方形格子の内部について，任意の位置 (x, y) における変位 w を，以下に示す形状関数を用いて表現する．この式は， x ＝一定，または y ＝一定の直線上で形状が 3 次曲線状に変形することを示している．

$$\begin{aligned} w &= \alpha_1 + \alpha_2 x + \alpha_3 y + \alpha_4 x^2 + \alpha_5 xy \\ &+ \alpha_6 y^2 + \alpha_7 x^3 + \alpha_8 x^2 y + \alpha_9 xy^2 \\ &+ \alpha_{10} y^3 + \alpha_{11} x^3 y + \alpha_{12} xy^3. \end{aligned} \quad (4.7)$$

式 4.8 の定数 $\alpha_1 \sim \alpha_{12}$ を求めるには，12 元連立方程式を解く必要がある．本手法では，有限要素法を用いて w を解く過程で， w の他に，格子点の変位にともなう x, y 軸方向の回転 θ_x および θ_y を同時に求める．長方形格子の 4 格子点において， w, θ_x, θ_y の各値を求めることで，下記に示すような 12 元連立方程式が得られ，これを用いて $\alpha_1 \sim \alpha_{12}$ の各値を求めることにより， U を算出するための積分が可能になる．

$$\begin{aligned} w_i &= \alpha_1 + \alpha_2 x_i + \alpha_3 y_i + \alpha_4 x_i^2 + \alpha_5 x_i y_i \\ &+ \alpha_6 y_i^2 + \alpha_7 x_i^3 + \alpha_8 x_i^2 y_i + \alpha_9 x_i y_i^2 \\ &+ \alpha_{10} y_i^3 + \alpha_{11} x_i^3 y_i + \alpha_{12} x_i y_i^3 \end{aligned} \quad (4.8)$$

$$\begin{aligned}
-\left(\frac{\partial w}{\partial y}\right)_i = \theta_{xi} &= -\alpha_3 - \alpha_5 x_i - 2\alpha_6 y_i - \alpha_8 x_i^2 - 2\alpha_9 x_i y_i \\
&\quad - 3\alpha_{10} y_i^2 - \alpha_{11} x_i^3 - 3\alpha_{12} x_i y_i^2
\end{aligned} \tag{4.9}$$

$$\begin{aligned}
-\left(\frac{\partial w}{\partial x}\right)_i = \theta_{yi} &= \alpha_2 + 2\alpha_4 x_i + \alpha_5 y_i + 3\alpha_7 x_i^2 + 2\alpha_8 x_i y_i \\
&\quad + \alpha_9 y_i^2 + 3\alpha_{11} x_i^2 y_i + \alpha_{12} y_i^3
\end{aligned} \tag{4.10}$$

本手法では、以上の手法によって求められた、格子点の z 方向の変位 w を、格子点の位置における要素サイズ値として扱う。

4.3.3 要素サイズ関数を参照するバブル・メッシュ法

バブル・メッシュ法では、力学シミュレーションの反復処理の過程において、球状物体が移動するたびに、その球状物体の位置における要素サイズ値を求め、球状物体の直径とする。ここで、形状領域および、要素サイズ関数である直交格子形状の座標値は、直交格子形状の辺に平行な方向に x 軸および y 軸をもつ 2 次元座標系に変換されているものとする。

要素サイズ値のサンプリング値を入力するバブル・メッシュ法では、下記のアルゴリズムにより要素頂点を生成する (図 4.7 参照)。

- 入力形状の頂点上に、球状物体を配置する。
- 入力形状の辺上において、以下の処理を実行する。
 1. 2 分木を用いた再帰処理により、要素サイズのサンプリング値を参照しながら球状物体の初期位置を決定する。
 2. 辺上の球状物体において、隣接物体間の分子間力を算出する。
 3. 辺上のすべての球状物体の変位が十分小さくなるまで、2. を反復する。反復が終わった時点で隣接球状物体間の距離を算出し、距離が小さい部位では球状物体を間引き、距離が大きい部位では球状物体を追加して 2. を反復する。間引きや追加がなかったら反復処理を終了する。
- 入力形状の領域内部において、以下の処理を実行する。

1. 4分木を用いた再帰処理により，要素サイズのサンプリング値を参照しながら球状物体の初期位置を決定する．
2. デローニー三角メッシュ生成法を用いて，三角メッシュを生成する．以後，三角形要素の辺で連結された球状物体を，隣接球状物体とみなす．
3. 領域内部の球状物体において，隣接物体間の分子間力を算出する．
4. 領域内部のすべての球状物体の変位が十分小さくなるまで，3. を反復する．反復が終った時点で隣接球状物体間の距離を算出し，距離が小さい部位では球状物体を間引き，距離が大きい部位では球状物体を追加して，デローニー三角メッシュ生成法によって三角メッシュを生成し，3. を反復する．間引きや追加がなかったら，反復処理を終了する．

ここで，分子間力の算出，および球状物体の変位の算出のアルゴリズムは，下記の通りである．

1. 隣接する球状物体 A および B の直径 r_A, r_B ，および AB 間の距離 d を算出する．
2. 直径の和に対する距離の比 $R = d/(r_A + r_B)$ を算出する．
3. 式 2.12 を満たす 3 次式を用いて，AB 間の分子間力 f を算出する．
4. A, B の座標値を P_{A0}, P_{B0} として，新しい座標値 P_{A1}, P_{B1} を下記の式により算出する．

$$\begin{aligned}
 P_{A1} &= P_{A0} - f(P_{B0} - P_{A0}), \\
 P_{B1} &= P_{B0} - f(P_{A0} - P_{B0})
 \end{aligned}
 \tag{4.11}$$

上記のアルゴリズムにおいて，球状物体の直径 r_A および r_B は，要素サイズ値をサンプリングした直交格子形状を用いて算出される．球状物体の中心を内部に含む長方形格子を特定し，その 4 格子点における要素サイズ値を補間することで，球状物体の直径が算出される．要素サイズ値の補間には，式 4.7 に示す 3 次関数を用いることが望ましい．ただし実用上は，もっと簡単な補間を用いることで十分である．本章では，処理速度の観点から，要素サイズ値を線形補間によって算出する．球状物体の中心の位置を (x, y) ，直交格

子形状中の球状物体を含む長方形格子の4点の座標を $(x_1, y_1), (x_1, y_2), (x_2, y_1), (x_2, y_2)$, 長方形格子の4点における要素サイズ値を $r_{11}, r_{12}, r_{21}, r_{22}$ とする (図 4.5 参照) . ただし , $x_1 < x < x_2, y_1 < y < y_2$ である . この時 , 球状物体の中心の位置における要素サイズ値は ,

$$(1.0 - t)((1.0 - s)r_{11} + sr_{21}) + t((1.0 - s)r_{12} + sr_{22}) \quad (4.12)$$

によって算出される . ただし ,

$$s = \frac{x - x_1}{x_2 - x_1}, t = \frac{y - y_1}{y_2 - y_1} \quad (4.13)$$

である .

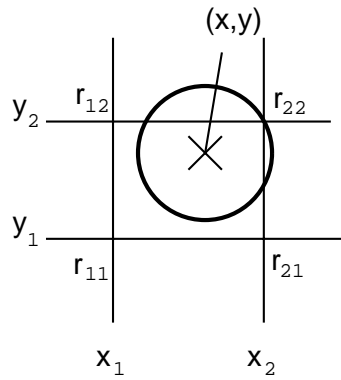


Figure 4.5: 球状物体の中心位置における要素サイズ値の算出

前述した通り , 本手法ではバブル・メッシュを用いてパッチを生成し , そのパッチを再分割してエレメントを生成している . 本手法における , パッチおよびエレメントの生成のアルゴリズムを , 下記の通りまとめる (図 4.6 参照) .

1. 分割された各領域の頂点 , 边上 , 領域内部において , 要素サイズのサンプリング値を参照しながらバブル・メッシュ法を実行し , ノードを配置する .
2. デローニー三角メッシュ生成法を用いてノードを連結し , 三角メッシュを生成する . 三角メッシュはパッチの集合として保存される .
3. パッチを再分割してエレメントを生成する . この再分割は , デローニー三角メッシュ生成法を用いてもよいが , 2.3 節に示した , パターンマッピングや再帰的分割のよう

な単純な手法で十分な場合も多い．処理速度を重視するならば，それらの単純な手法の方が有利である．

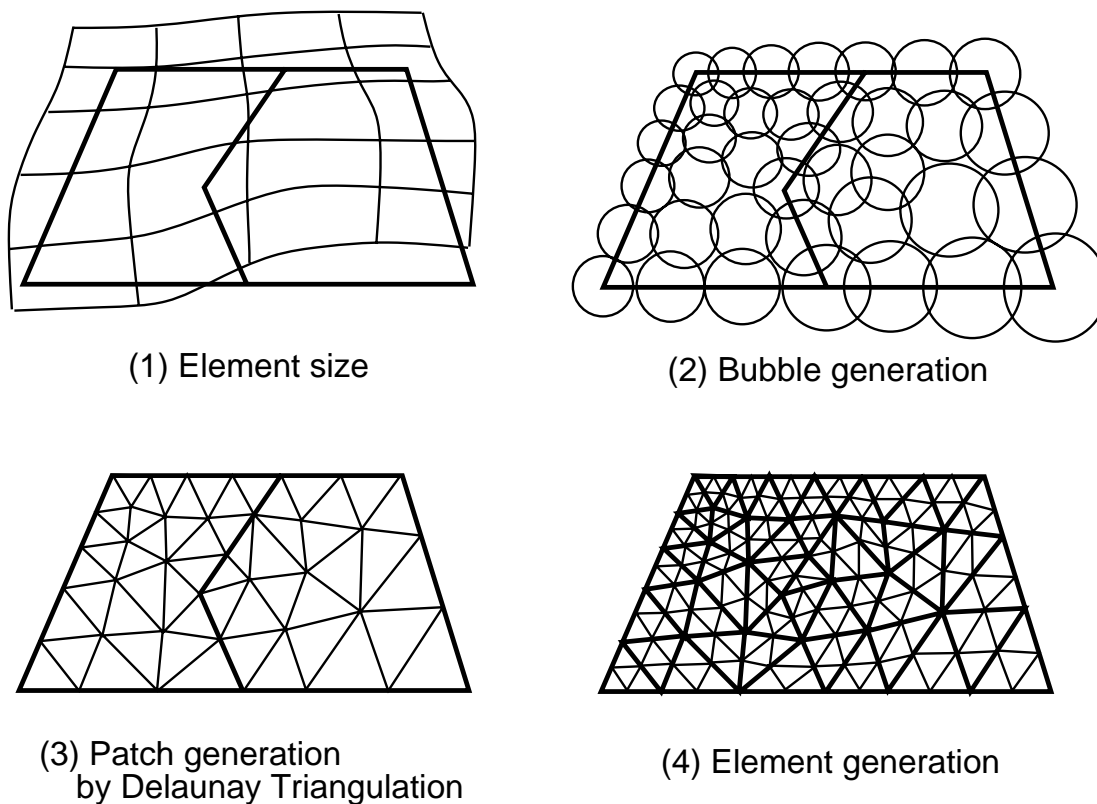


Figure 4.6: 本手法におけるパッチ/エレメント生成

4.4 実行例

本節では，本手法を実装して実行した結果を示す．なお，計算機は IBM PowerStation RS/6000 (Model 560) を用いた．

図 4.8 は，直方体の内部に 3 枚の長方形が浮遊する単純な入力形状に対して，パッチおよびエレメントを生成した結果である．図 4.11 は，図 4.8 に示されるメッシュを用いて，ラジオシティ法を実行して生成した画像である．

図 4.9 は，図 4.11 に示すラジオシティ法の算出結果を用いて，従来の適応的メッシュ生成手法により再生成されたメッシュである．図 4.10 は，図 4.11 に示すラジオシティ法

の算出結果を用いて、本手法により再生成されたメッシュである。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 4.12 および図 4.13 は、それぞれ図 4.9 および図 4.10 に示されるメッシュを用いて、ラジオシティ法を再実行して生成した画像である。図 4.11 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 4.1 は、図 4.8、図 4.9、および図 4.10 に示すメッシュのエレメント数である。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エレメント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エレメント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。

Table 4.1: メッシュ生成結果におけるエレメント数とノード数

	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 4.2 は、図 4.13 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をまとめたものである。

Table 4.2: 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、要素サイズのサンプリング値の算出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、要素サイズの算出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。

4.5 むすび

本章では、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生成し、その要素サイズ関数によって算出された要素サイズのサンプリング値を参照しながらメッシュを再生成する、輝度分布参照型バブル・メッシュ法を提案した。本手法では、1回目のラジオシティ法でメッシュを細かく生成し、輝度変化の小さい部位の要素を大きく修正することが出来る。よって、1回目のラジオシティ法では細かい陰影を検出することが出来る上に、2回目のラジオシティ法の終了後には要素数の増加を抑えた出力結果が得られる。また本手法は、なめらかに数値が変化する要素サイズ関数を用いているので、歪みの小さいメッシュを生成することが出来る。また、利用者が与えるパラメータは要素サイズ値であり、これは輝度差のしきい値を与える従来の手法のパラメータよりも操作性が高いと思われる。

付録

(定理 4.1 の証明)

バブル・メッシュ法によって生成される三角形要素の3頂点 A, B, C における要素サイズ値を、それぞれ r_a, r_b, r_c とする。この時、三角形要素の3辺の長さは、

$$AB = 0.5(r_a + r_b), \quad BC = 0.5(r_b + r_c), \quad CA = 0.5(r_c + r_a)$$

と表される。この3辺の長さの比が大きいほど、三角形要素は細長くなる。

ここで、要素サイズ値の大小関係を $r_a < r_b < r_c$ と仮定し、

$$r_b = sr_a, \quad r_c = tr_a, \quad 1 < s < t$$

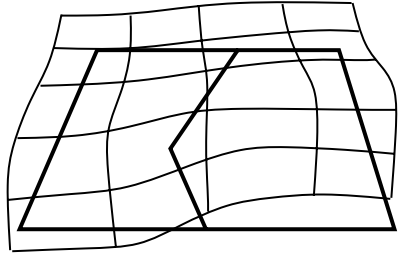
とすると、3 辺の長さは

$$AB = 0.5r_a(1 + s), \quad BC = 0.5r_a(s + t), \quad CA = 0.5r_a(1 + t)$$

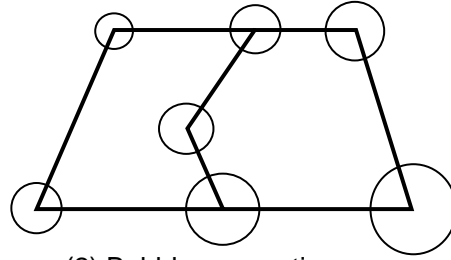
であるので、最も短い辺 AB と、最も長い辺 BC の長さの比は、

$$BC/AB = \frac{s+t}{1+s}$$

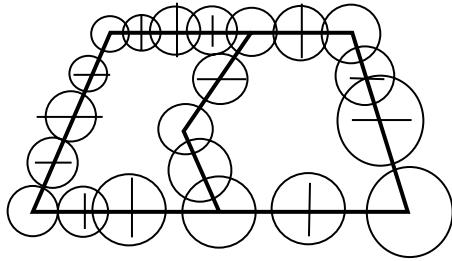
と表現され、この値は t の増加に対して単調増加である。よって、要素サイズの変化が急激である時に、要素サイズ値 r_a と r_c の比が大きくなるので、三角形要素の辺の長さの比が大きくなり、要素の形状は細長く歪む。



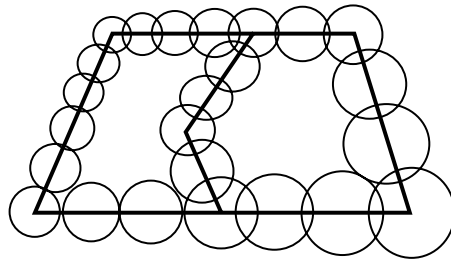
(1) Element size control function



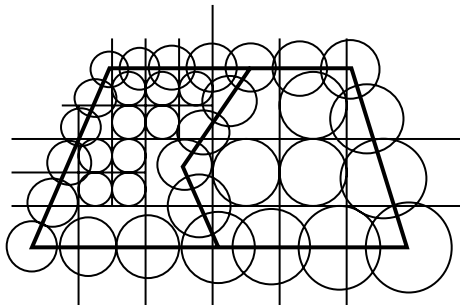
(2) Bubbles on vertices



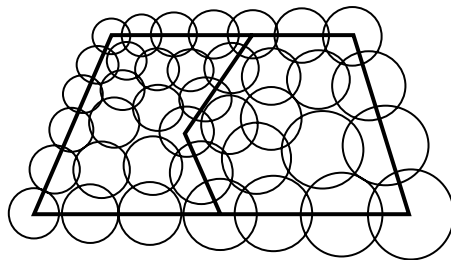
(3) Bubble initial location on edges



(4) Fixed bubbles on edges



(5) Bubble initial location on faces



(6) Fixed bubbles on faces

Figure 4.7: 輝度分布参照型バブル・メッシュ法

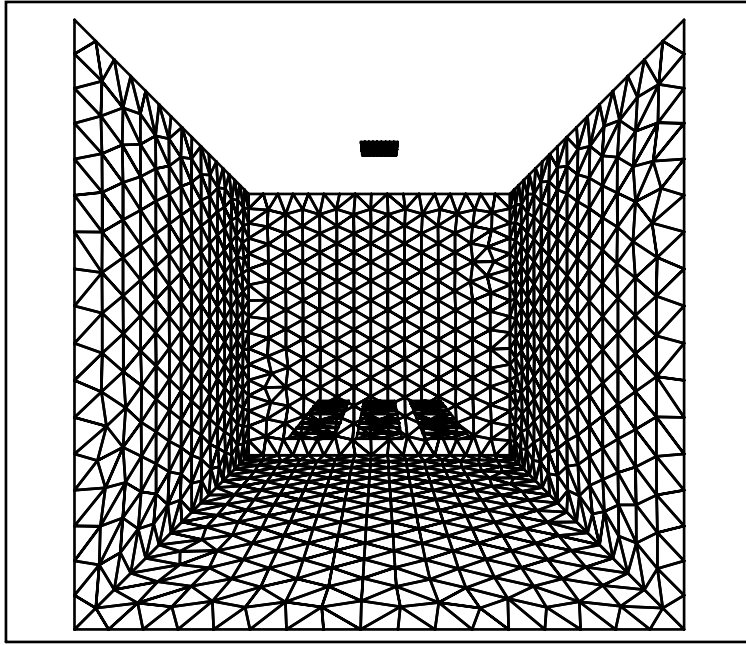


Figure 4.8: 1回目のメッシュ生成

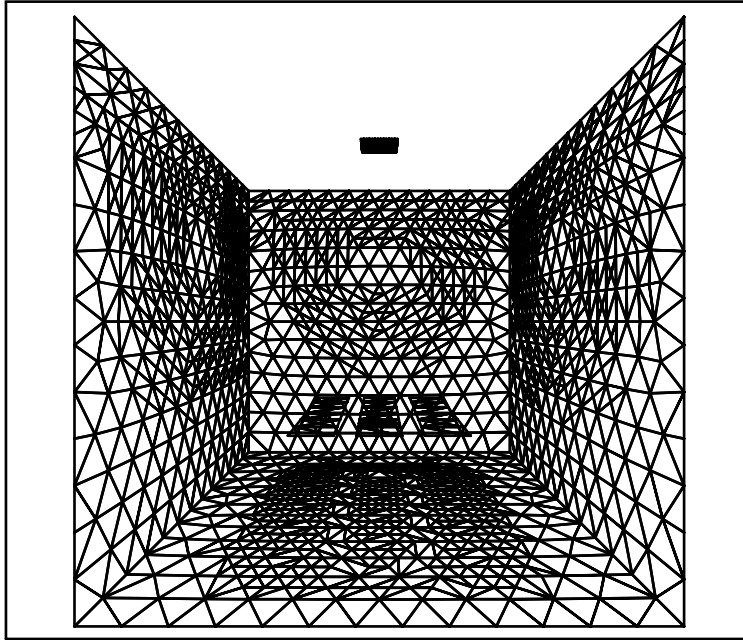


Figure 4.9: 従来手法の適応的メッシュ生成

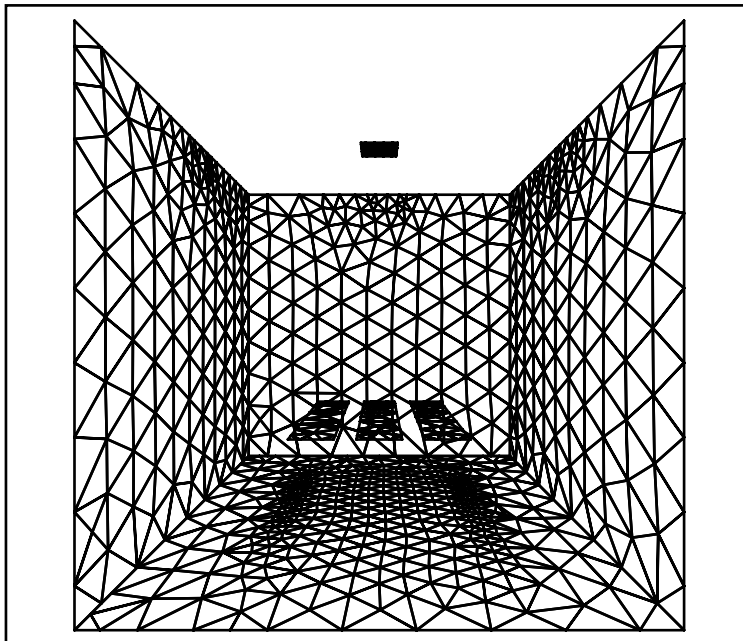


Figure 4.10: 本手法の適応的メッシュ生成

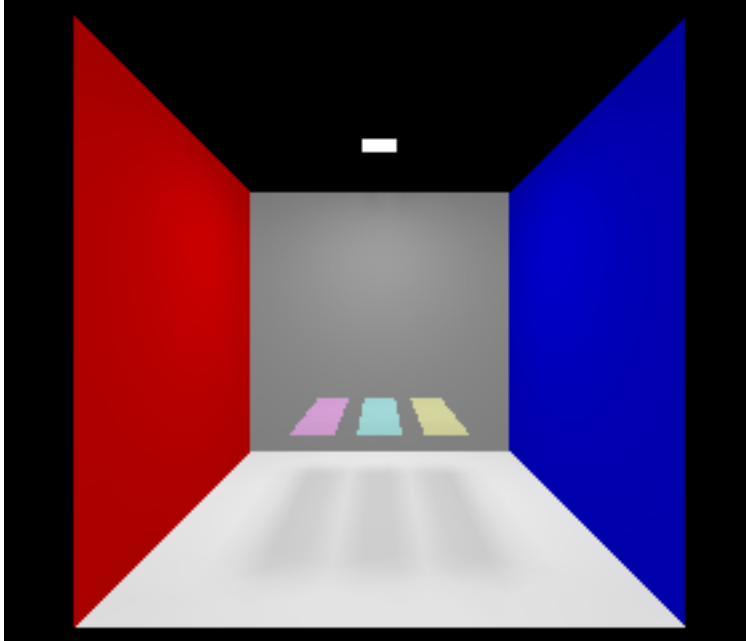


Figure 4.11: 1 回目のラジオシティ法算出結果による画像

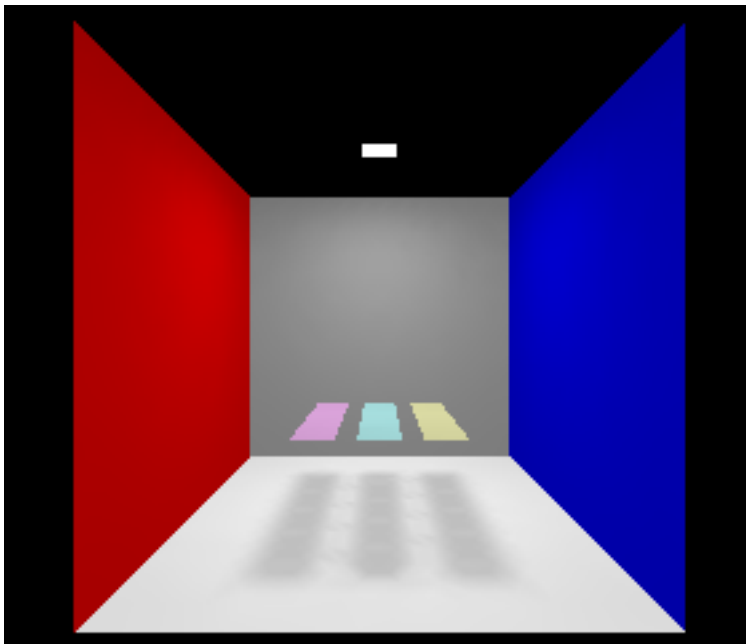


Figure 4.12: 従来手法の適応的メッシュ生成後の画像

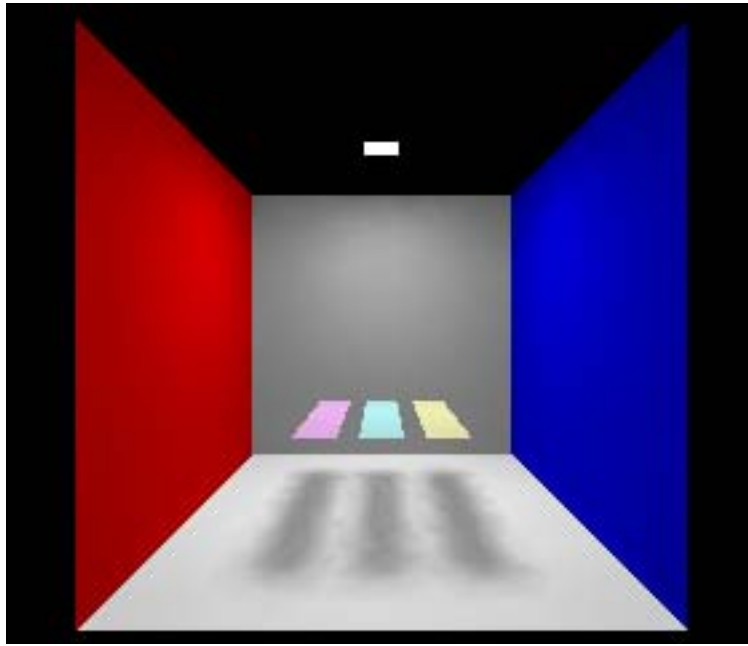


Figure 4.13: 本手法の適応的メッシュ生成後の画像

第 5 章

3次元数値解析結果のグラフィックス表示

5.1 はじめに

本章では、3次元数値解析結果の代表的なグラフィックス表示手法について、従来の研究成果を概観する。

5.2節では、3次元数値解析結果を格納するデータ構造であるボリューム・データを定義する。また、その代表的なグラフィックス表示手法に、インダイレクト・アプローチと、ダイレクト・アプローチの二者があることを示す。

5.3節では、インダイレクト・アプローチの代表例である等値面生成について、従来の研究成果を概観する。

5.4節では、等値面生成の高速化手法を列挙し、その処理量や問題点について述べる。

5.2 ボリューム・データのグラフィックス表示手法

本節では、3次元数値解析結果を格納するボリューム・データのデータ構造と、そのグラフィックス表示手法について概観する。

5.2.1 ボリューム・データのデータ構造

有限要素法などによる3次元数値解析結果は、ボリューム・データとよばれるデータ構造で表現されることが多い。ボリューム・データは、有限要素法などの数値解析結果の他に、CT スキャンなどを用いた医療測定結果、気象・天文・地学などの各分野の計算機シミュレーション結果などの表現に用いられる。

本論文では、ボリューム・データのデータ構造を、下記の通り定義する [34]。また、このデータ構造を図解したものを図 5.1 に示す。

定義 5.1 ボリューム・データの対象となる3次元空間領域を、四面体や六面体の小片に分割した単位を、要素 (Element) と定義する。要素 E は、後述する要素頂点 $\{N_1, N_2, \dots\}$ 、および要素の面を共有する隣接要素 $\{E_1, E_2, \dots\}$ から構成されるものとする。本論文では、異なる要素頂点をもつ面で接している要素は、幾何学的に隣接していても、隣接要素として扱わない。

定義 5.2 要素の頂点を、要素頂点 (Node) と定義する。要素頂点 N は、3次元座標値 $\{x, y, z\}$ と、その位置における数値解析結果のスカラー値 $\{S\}$ より構成されるものとする。

要素頂点は通常，隣接する複数の要素に共有されている．数値解析の種類によっては，要素頂点にベクタ値が与えられている場合もあるが，本論文では扱わない．

定義 5.3 要素の面 (四面体要素では三角形，六面体要素では四角形) のうち，ただ 1 個の要素に接する面を，境界面 (Boundary Face) と定義する．境界面 B は，その面をもつ要素 $\{E\}$ と，要素中の面の番号 $\{i\}$ より構成される．

定義 5.4 有限な 3 次元空間領域を，多数の四面体や六面体の要素に分割し，各々の要素頂点にスカラ値を与えたデータを，ボリューム・データ (Volume Data) と定義する．本論文では，四面体要素のみで構成されるボリューム・データ，および六面体要素のみで構成されるボリューム・データのみを対象にする．また，ボリューム・データの対象となる空間領域を，ボリュームと呼ぶ．

なお，第 2 章で扱う要素を 2 次元要素，本章で扱う要素を 3 次元要素と呼ぶこともある．

ボリューム・データをグラフィックス表示する際には，ボリューム・データ中の任意の点 P におけるスカラ値を求める処理が必要がある．本論文では，以下の定義を用いて，ボリューム・データ中の任意の点 P におけるスカラ値を求める．

定義 5.5 ボリューム・データ中の任意の点 P におけるスカラ値は， P を内部に含む要素 E の，要素頂点 $\{N_1, N_2, \dots\}$ におけるスカラ値を線形補間して求める．よって，ボリューム・データ中のスカラ値は，任意の要素内部では連続かつ微分可能である．ただし，要素の面上や辺上においては，スカラ値は連続であるが微分可能であるとは限らない．

5.2.2 ボリューム・データの種類

ボリューム・データの種類は，要素の配列状態から，下記の通り分類される．また，この各形態を図解したものを，図 5.2 に示す．

直交 (Rectangular): 直方体の要素が 3 次元直交座標系の各軸方向に整列し，要素の位相的な位置が 3 次元配列で表わされる．要素の各辺は，座標系の各軸に平行である．

構造 (Structured): 直交ボリュームと同様に，要素の位相的な位置が 3 次元配列で表わされるが，要素の各辺は，座標系の各軸に平行ではない．要素は直方体に限らない

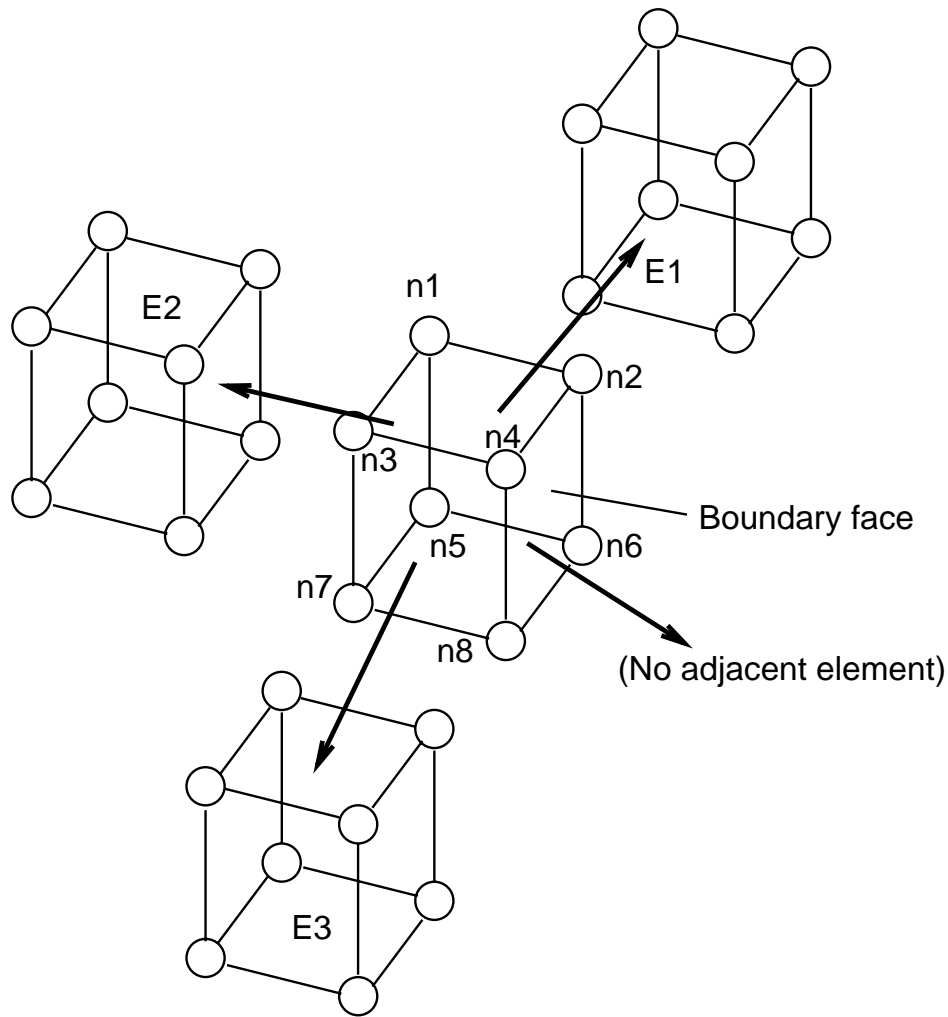


Figure 5.1: 本論文で扱うボリューム・データのデータ構造

六面体である．構造ボリュームの特殊なものが直交ボリュームであるとも考えることもできる．

非構造 (Unstructured): 要素の位相的な位置が3次元配列で表わされない．要素は六面体に限らない．本論文では六面体のみで構成されるボリュームと，四面体のみで構成されるボリュームを対象とする．

ここで，本論文で対象とする非構造ボリュームの形態を制限するために，下記の定義を示す．

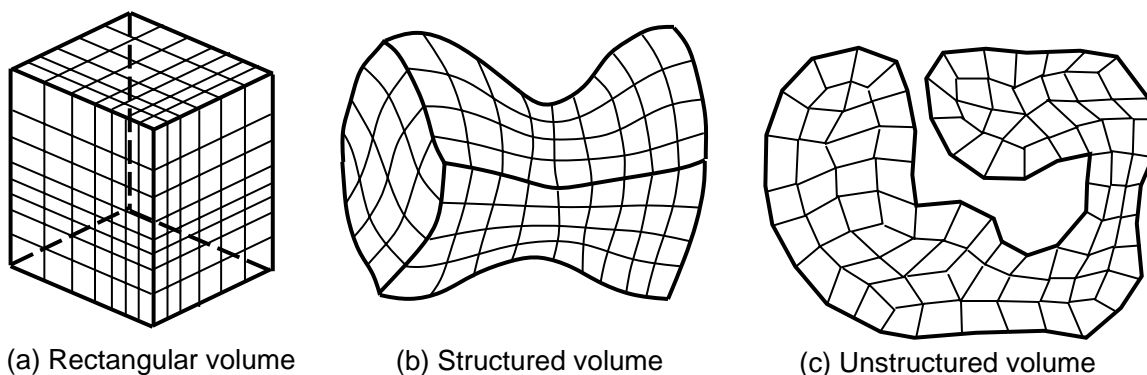


Figure 5.2: 本論文で扱うボリューム・データの形態

定義 5.6 ボリューム・データ中の境界面が非連結な n 個の境界面群にわかれる時、境界面群がボリューム内部に要素のっていない空虚な領域を囲い込むことがある。本論文では、この領域を 空洞 (Void) と定義する。

定義 5.7 ボリューム・データ中の連結な境界面群のトポロジーを表す指標として本論文では、

(オイラー数) = (境界面に接する要素頂点数) + (境界面数) - (境界面に接する辺の数) を用いる。

定義 5.8 ボリューム・データ中の連結な境界面群のオイラー数を E とした時に、

$$G = 1 - E/2$$

で与えられる G を、種数 (Genus) と定義する。この時、ボリュームには G 個のドーナツ状の穴が見られる。本論文では、この穴を 貫通穴 (Through-hole) と定義する。

これらの定義を用いて、本論文では、以下のように対象とする非構造ボリュームの形態を制限する。

定義 5.9 本論文では、境界面が非連結な複数の境界面群にわかれるような非構造ボリュームも対象とする。

定義 5.10 本論文では、貫通穴や空洞をもつ非構造ボリュームも対象とする。ただし、空洞に接する境界面群が貫通穴をつくるような非構造ボリュームは対象としない。

5.2.3 ボリューム・データのグラフィックス表示手法

Zバッファ法，レイ・トレーシング法，ラジオシティ法などの，従来のレンダリングアルゴリズムは，物体の面（サーフェス）や線（ワイヤフレーム）を表現することを前提としていた．よって，従来のレンダリングアルゴリズムで，ボリューム・データを直接入力して画像を生成することは困難である．そこで，ボリューム・データを表現するアルゴリズムを構築するために，以下の2通りの考え方が適用される（図 5.3 参照）．

インダイレクト・アプローチ: ボリューム・データの一部の特徴を，面や線で抽出し，従来のレンダリングアルゴリズムに入力して画像を生成する手法の総称．スカラ場を扱う代表的な手法に，ボリューム・データ中の同一スカラ値をもつ点の集合を曲面で表現する，等値面 (Isosurface) がある．

ダイレクト・アプローチ: ボリューム・データを直接レンダリングアルゴリズムに入力し，ボリューム・データのための特殊なレンダリングアルゴリズムを用いて，ボリューム・データの数値分布を1枚の画像で表現する手法の総称．代表的な手法に，ボリューム・レンダリング (Volume Rendering) がある．ボリューム・レンダリングは，ボリューム中に半透明の微小物体が浮遊しているモデルを仮定し，微小物体の密度を数値の大小，微小物体の面法線方向をベクタ場 (スカラ値の勾配方向など) から設定することで，ボリューム全体の数値分布を表現する手法である．

数値解析結果のグラフィックス表示では，上記の2種類のアプローチを使い分けることが多い．例えば，数値解析結果を評価するためにグラフィックス表示する場合に，評価の初期段階では高速なインダイレクト・アプローチを用いて試行錯誤的に画像を連続生成し，評価の最終段階ではボリューム全体を1枚の画像で表現できるダイレクト・アプローチを用いる，というような使い分けをすることが多い．つまり，数値解析結果のグラフィックス表示においては，インダイレクト・アプローチと，ダイレクト・アプローチの両者が重要な技術であるといえる．また，インダイレクト・アプローチでは，その画像表示の対話性が非常に重要である．

インダイレクト・アプローチ，およびダイレクト・アプローチによって，ボリューム・データから画像を生成することを，総じてボリューム・ビジュアライゼーション (Volume Visualization) と呼ぶことが多い．

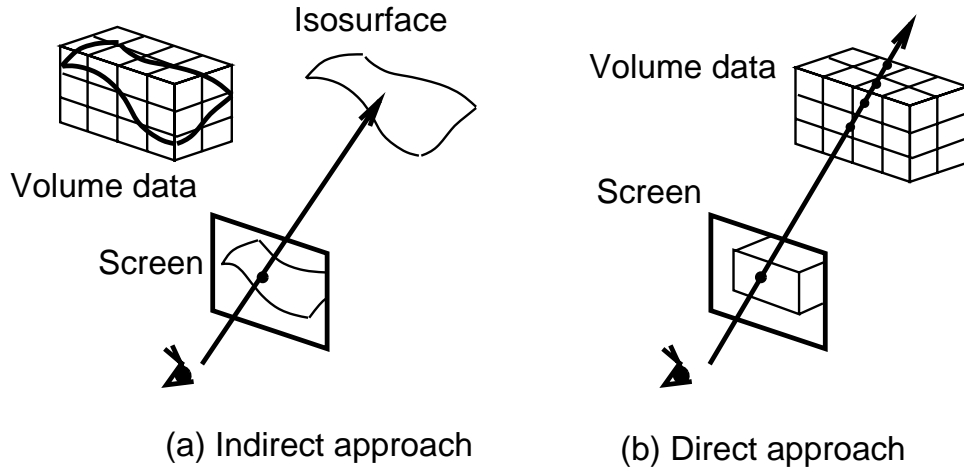


Figure 5.3: インダイレクト・アプローチとダイレクト・アプローチ

5.3 等値面生成

本節では，スカラ場をグラフィックス表示するインダイレクト・アプローチの代表的な手法である，等値面生成手法について概観する．

5.3.1 等値面生成

等値面の本来の定義は，同一のスカラ値をもつ点の集合，つまり

$$S(x, y, z) - C = 0 \tag{5.1}$$

を満たす点の集合である．ここで， $S(x, y, z)$ はボリューム中の任意の点の座標値より得られるスカラ値である．また， C は定数である．しかし，上記の式を満たす曲面を生成することは難しい．そこで等値面は，ポリゴンの集合として近似されるのが一般的である．

等値面を生成する手法として旧来から知られている手法に，ボリューム中に多数の断面を生成し，断面中に生成した等高線を多角形で接続する手法がある [7] [10] [55]．しかしこの手法は，2 つ以上に非連結な部分に分断される等値面の生成処理を的確に行うことが難しい．近年ではこの手法にかわって，ボリュームの各要素と等値面との交差部分を多角形で近似する手法 [14] [43] [71] が用いられることが多い．この手法は，文献 [43] のタイトルから，マーチング・キューブ法 (Marching Cube) と呼ばれることが多い．

マーチング・キューブ法では，等値面の定数 C が与えられた時に，ボリューム・データ中の要素頂点におけるスカラ値との大小を比較し，比較結果(大，小，または等)を要素頂点ごとに記録する(図 5.4 (a) 参照)．続いて，各要素の辺の両端点のスカラ値 C_1 および C_2 に対して， $C_1 < C < C_2$ である辺を抽出する．要素の辺上におけるスカラ値は，両端の要素頂点におけるスカラ値の線形補間で求められるので，抽出した辺には等値面との交点が1点存在する．この時，等値面の交点の座標値 P_c は，両端の要素頂点の座標値を P_1 および P_2 として，

$$P_c = \frac{(C - C_1)P_2 - (C - C_2)P_1}{C_2 - C_1} \quad (5.2)$$

で算出される(図 5.4 (b) 参照)．

等値面と交点をもつ辺が抽出された要素では，それを連結して多角形を構成することで，等値面との交差部分を表現することが出来る(図 5.4 (c) 参照)．本論文では，等値面の交差部分の多角形を，すべて対角線で三角形面素に分割して処理するものとする(図 5.4 (d) 参照)．

本論文では，等値面を以下の通り定義する．

定義 5.11 マーチング・キューブ法によって得られる三角形面素の集合を，等値面 (Iso-surface) と定義する．等値面を形成する三角形面素 T は，後述する3頂点 $\{V_1, V_2, V_3\}$ より構成される．等値面の三角形面素の頂点 V は，3次元座標値 $\{x, y, z\}$ と，法線ベクトル $\{n_x, n_y, n_z\}$ より構成される．等値面の三角形面素のすべての頂点は，要素の辺上に存在する．等値面の三角形面素の辺は，境界面上にあるものを除いて，すべて2枚の三角形面素に共有される．また，等値面は非連結な2個以上の部分に分離することもある．

マーチング・キューブ法で等値面との交差部分を多角形近似する時，四面体要素では等値面の交点の連結に曖昧性は生じないが，六面体では曖昧性が生じる．この曖昧性を解消する方法として，要素の面の重心におけるスカラ値を参照する手法 [71]，スカラ値の勾配方向を参照する手法 [19]，隣接要素における交差部分の多角形近似を参照する手法 [25] などが提案されている．また，六面体をあらかじめ5個の四面体に分割して交差部分を多角形近似する手法 [14] も提案されている．

第2章で示されたグーロー・シェーディングは，等値面の表示にも多く用いられる．この時，多角形の頂点における等値面の法線方向が算出される必要がある．初期の手法で

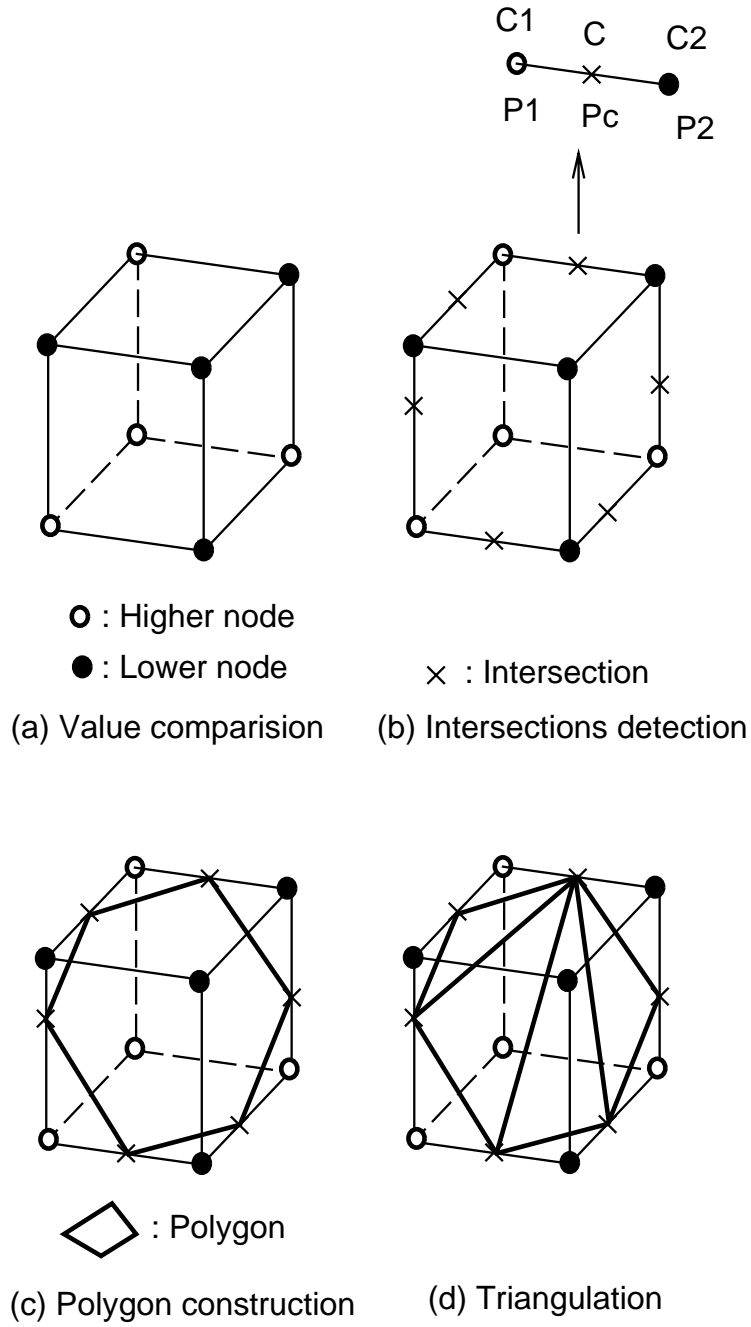


Figure 5.4: マーチング・キューブ法による等値面生成

は、等値面の法線方向はスカラ場の勾配方向から算出されていた [16] [43]。本論文では、スカラ場の勾配方向を用いるかわりに、隣接多角形の法線方向の平均値を求める手法 [14]

[63] を用いている．この手法は，隣接多角形に共有される頂点を，ハッシュ・テーブルなどの検索手法を用いて同一化しているので，メモリ使用量の低減や，要素辺上における等値面との交点の座標値算出の重複の回避，という意味でも有用である．

5.3.2 等値面生成手法の問題点

数値解析結果をグラフィックス表示するソフトウェアには，式 5.1 における定数 C を連続的に変化させて，複数の等値面を生成して連続表示する機能 (図 5.5 参照) を実装したものが多く見られる．このアニメーション表示は，数値分布の理解に重要な役割を果たすことが多い．

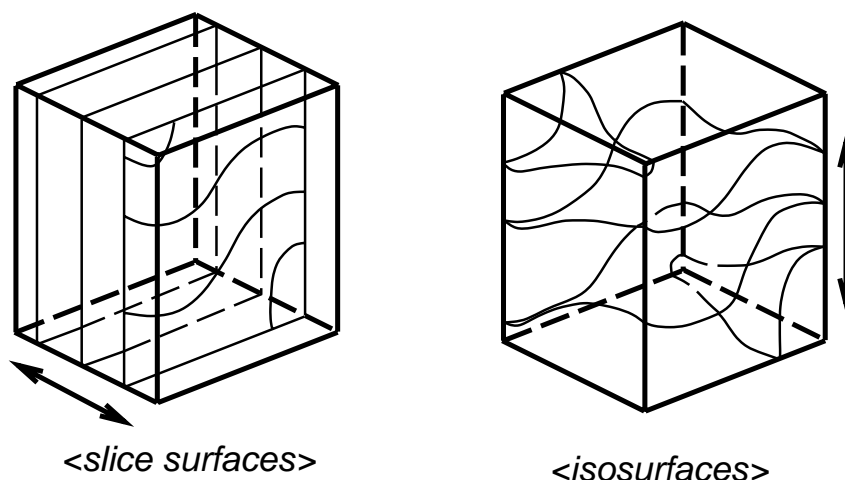


Figure 5.5: 等値面や断面のアニメーション表示

要素数の非常に多い大規模なボリュームデータでは，等値面生成の処理時間が大きくなる．等値面生成に大きな処理時間を要すると，等値面の連続生成表示によって数値分布を理解することが難しいことも多い．

最も単純な等値面生成アルゴリズム [14] [43] では，すべての要素に対して等値面との交差判定が実行される．しかし，一般的に 1 枚の等値面と交差する要素は，ボリューム・データ中のわずか一部である場合が多い．表 5.1 は，定数 C を連続して 20 枚の等値面を生成した時の，交差要素および非交差要素の総数を示したものである．この測定結果では，有限要素法によって得られた 4 つの非構造ボリューム・データが用いられている．

Table 5.1: 交差要素と非交差要素の総数

データ番号	1	2	3	4
N_i	51357	62377	148836	277457
N_n	363363	1171223	6784044	10879903
$20N_c$	414720	1233600	6932880	11157360
$N_i/20N_c$	0.1238	0.05057	0.02147	0.02487

表 5.1 において，それぞれの変数は，

N_i : 20 枚の等値面との交差要素の総数．

N_n : 20 枚の等値面との非交差要素の総数．

$20N_c$: ボリューム中の要素数の 20 倍 ($= N_i + N_n$)

を示している．

上記の結果は，大規模なボリュームになるほど非交差要素の割合が高くなる傾向を示している．よって，全要素を処理する単純な手法では，ボリュームの要素数が大きくなるほど，不必要な処理を多く費やすことがわかる．

5.4 等値面生成の高速化手法

5.3 章で述べた通り，全要素を処理する初期の等値面生成アルゴリズムは，等値面と交差しない要素との交差判定に，不必要な多くの処理量を要することが知られている．本章では，等値面を高速に生成する手法として，非交差要素との交差判定処理を減らす高速化手法について，従来の研究成果を概観する．なお，表 5.2 に，本節で示す諸手法の処理量がまとめられている．表 5.2 において， n および k はそれぞれ，ボリューム・データの要素の総数，および等値面と交差する要素の総数である．また，本論文では，処理量の評価のために，下記の仮定を用いている．

仮定 5.1 ボリューム中の要素数が n である時，ボリューム内部または境界上の直線に交差する要素数は $O(n^{1/3})$ である．また，ボリューム内部または境界上の平面に交差する要素数は $O(n^{2/3})$ である．

5.4.1 等値面生成の高速化手法

Giles と Haimes は、要素を要素頂点の数値でソートする手法を報告している [21] . この手法では、前処理として、各要素 E がもつ要素頂点 $\{N_1, N_2, \dots\}$ の数値の最小値 S_{min} および最大値 S_{max} を算出し、最小値で全要素をソートしたリスト、および最大値で全要素をソートしたリストをつくる . 等値面を生成する時には、式 5.1 における定数 C が与えられると、前処理で作成されたリストを参照し、 $S_{min} > C$ である要素、および $S_{max} < C$ である要素を除外したリストをつくる . 等値面との交差判定は、このリスト中の要素のみに実行される . このリストには、ボリューム中の一部の要素のみが登録されるので、多くの非交差要素は処理から除外される .

Gallagher は、要素を要素頂点の数値で階級化する手法を報告している [17] . この手法では、まず全要素頂点の数値の最大値および最小値を等分する階級 $\{Group_1, Group_2, \dots, Group_n\}$ をつくる . 前処理では、各要素 E がもつ要素頂点 $\{N_1, N_2, \dots\}$ の数値の最小値 S_{min} および最大値 S_{max} を算出し、 S_{min} が該当する階級 $Group_{S_{min}}$ から S_{max} が該当する階級 $Group_{S_{max}}$ までのすべての階級に E を登録する . 等値面を生成する時には、定数 C が与えられると、 C が該当する階級 $Group_C$ に登録された要素だけが交差判定の対象となり、多くの非交差要素は処理から除外される .

上記の 2 手法は、いずれも処理される要素数が $O(n)$ である . 一方、等値面と交差する要素数は、一般的には $O(n)$ より小さい . よって、要素数の多い大規模なボリュームにおいても高速に等値面を生成するためには、処理する要素数が $O(n)$ より小さくなるような等値面生成手法が確立されることが望ましい .

Wilhelms と Gelder は、ボリュームを空間分割し、分割された領域を管理する木構造に要素を保持する手法を報告している [67] . それぞれの領域には、領域中の要素の頂点の最大値 S_{max} および最小値 S_{min} が記録される . 等値面を生成する時には、定数 C が与えられると、 $S_{min} > C$ である領域、および $S_{max} < C$ である領域に登録されている要素を処理から除外する . よって、非交差要素の多くは処理から除外される . この手法における等値面生成の処理量は $O(k + k \log n/k)$ であることが、文献 [41] により示されている . 文献 [67] では、この手法は非構造ボリュームには適用されていない . 原理的にはこの手法は非構造ボリュームに適用が可能であるが、その効率性は薄れるものと思われる .

Speray と Kennon は、自己増殖的に等値面を生成する手法を提案している [59] . この手法は、等値面と交差する 1 要素を出発して、等値面と交差する隣接要素を再帰的に処理

する．このアルゴリズムは，等値面との交差要素のみを処理するが，等値面との交差要素を最低1個指定する必要がある．Wilhelms と Gelder の手法と同様な空間分割によって，等値面との交差要素を高速抽出して，自己増殖的に等値面を生成する手法も提案されている [57] が，非構造ポリウムには適用されていない．なお，隣接要素を再帰的に処理する自己増殖的なアルゴリズムは，Speray らの提案より以前に，陰関数表現された曲面をポリゴンの集合で近似表現する目的で提案されている [6] [71] ．

要素の隣接順にポリゴンを生成する他の手法として，文献 [25] に示される手法がある．この手法は，等値面との交差要素からポリゴンを生成し，その後ただ1個の隣接未処理要素を探索し，それを反復することでポリゴンの列を生成する手法である．この手法で等値面を生成するためには，多数の出発点要素が必要であるので，結局すべての要素との交差判定が必要になり，処理量は $O(n)$ となる．ただし，市販の多くのグラフィックスライブラリは，一列に配置したポリゴン列を高速表示する機能を持っているので，この手法によって表示を高速化することができる．

非構造ポリウムにおいても $O(n)$ 未満の処理量で等値面を生成する手法として，Livnat らは各要素のもつスカラ値の最大値 S_{max} と最小値 S_{min} の2次元座標系を用いた手法を提案している [41] [51] ．この手法は，まず要素ごとに S_{max} と S_{min} を記録し，それを2次元座標系に配置する．等値面のスカラ値 C が指定されたら，2次元座標系で $S_{min} < C < S_{max}$ である要素を抽出し，等値面との交差部分をポリゴンで近似する．図 5.6 (a) は， S_{min} および S_{max} の値を座標軸とする2次元座標系である．座標中の点は要素を表している．点線で囲まれた領域に属する要素が，等値面に交差する要素である．

文献 [41] では，k-d 木構造を用いて要素を管理している．この手法では，全要素を S_{min} でソートして2グループに大別し，2分された各々の要素グループを S_{max} でソートして2分し，以後， S_{min} ， S_{max} ，... の順に交互に同様な処理を反復し，すべての要素を2分木に格納する．図 5.6 (b) は，k-d 木構造を用いて要素を管理した例である．k-d 木構造はデータ処理の研究成果として以前から知られている手法であり，Livnat らの適用例では目的とする範囲内の数値をもつ要素の抽出に $O(n^{1/2})$ の処理量を要することが知られている．よって，等値面との交差要素数を k とすると，この手法は $O(n^{1/2} + k)$ の処理量で等値面を生成することが出来る．一方，k-d 木構造の構築にはクイックソート法を用いるので，この手法は前処理に平均 $O(n \log n)$ ，最大 $O(n^2)$ の処理量を要する．

文献 [51] では，2次元配列を用いて要素を管理している．この手法では，全要素を

S_{min} および S_{max} でソートした 2 つのリストをつくり，それぞれのリスト中で各階級に属する要素数が等しくなるように階級値を設定し，2次元配列に要素を格納する．図 5.6 (c) は，2次元配列を用いて要素を管理した例である．文献 [51] では，2次元配列の階級数を目的とする範囲内の数値をもつ要素を抽出するために， $O(\log \frac{n}{L} + \frac{n^{1/2}}{L})$ の処理量を要することが示されている．よって，等値面との交差要素数を k とすると，この手法は $O(\log \frac{n}{L} + \frac{n^{1/2}}{L} + K)$ の処理量で等値面を生成することが出来る．一方，2次元座標系の階級値の設定にクイックソート法を用いるので，この手法は前処理に平均 $O(n \log n)$ ，最大 $O(n^2)$ の処理量を要する．

Table 5.2: 高速な等値面生成手法の処理量

著者名	手法名	前処理量	本処理量	備考
Giles	順列	$O(n \log n)$	$O(n)$	
Gallagher	階級化	$O(n)$	$O(n)$	
Welhelms	空間分割	$O(n)$	$O(k + k \log n/k)$	非構造ボリュームに未適用
Silver	自己増殖+空間分割	$O(n)$	$O(k + k \log n/k)$	非構造ボリュームに未適用
Livnat	k-d 木構造	$O(n \log n)$	$O(n^{1/2} + k)$	
Shen	2次元配列	$O(n \log n)$	$O(\log \frac{n}{L} + \frac{n^{1/2}}{L} + K)$	

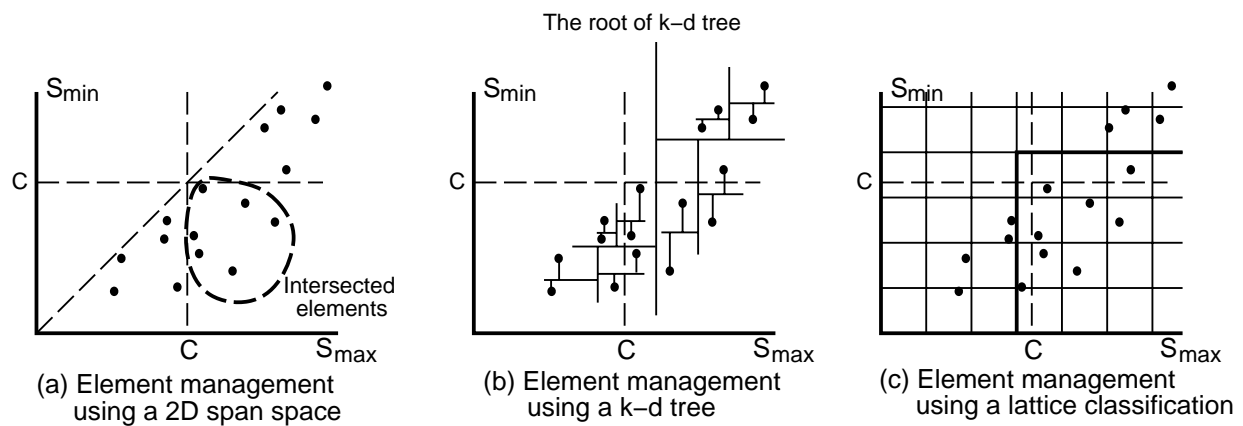


Figure 5.6: 2次元座標系による要素の格納

5.4.2 自己増殖的な等値面生成手法

本論文の第 6 章および第 7 章で提案する手法は，前節で示した Speray らの自己増殖的な等値面生成手法を用いている．本節では，自己増殖的な等値面生成手法のアルゴリズムを示す．

自己増殖的な等値面生成手法では，要素の面を共有する隣接要素との接続性を表わすデータ構造が必要である．よってこの手法は，構造ボリュームか，非構造ボリュームの場合には 5.2 節で示すような，それぞれの要素が隣接要素へのポインタを持つようなデータ構造で表現されるボリュームを前提とする．

等値面を形成するポリゴンの辺は，要素の面上にある．一方，要素の面を共有する隣接要素間には，数値の C_0 級の連続性が保持されるという特徴がある．よって，等値面を形成するポリゴンの辺が接する隣接要素は，等値面と必ず交差する．自己増殖的な等値面生成手法では，等値面と交差する要素内部にポリゴンを生成した後，ポリゴンの辺が接する隣接要素が FIFO キューに登録される．ただし，一度 FIFO キューに登録された要素はその時点でマークされ，同一等値面の生成のために再び FIFO キューには登録されない．FIFO キューに登録された要素は登録順に抽出され，同様にポリゴンを生成し，その隣接要素を FIFO キューに登録する．この処理はキューが空になるまで反復され，すべての等値面との交差要素が処理される (図 5.7 参照)．

文献 [59] では，この手法は断面生成のみに適用されていて，等値面生成には適用されていない．自己増殖的な等値面生成手法では，等値面と交差する要素を最低 1 個，自己増殖の出発点として指定しなくてはならない．文献 [59] では，この出発点を確実に抽出する手法については触れていない．断面は必ずボリュームの境界面に接するので，ボリュームの境界面上の要素との交差判定によって交差要素を抽出することが出来るが，等値面の場合は，閉曲面や，多数の非連結な部分にわかれた曲面が生成される可能性も高いので，交差要素の抽出は容易ではない．

以下，本論文で用いられている自己増殖的手法のアルゴリズムを示す．

1. 等値面と交差する 1 要素を指定し，FIFO キューに登録し「処理済」マークをつける．
2. 以下の処理を，FIFO キューが空になるまで反復する．
 - (a) FIFO キューから要素を 1 個抽出する．

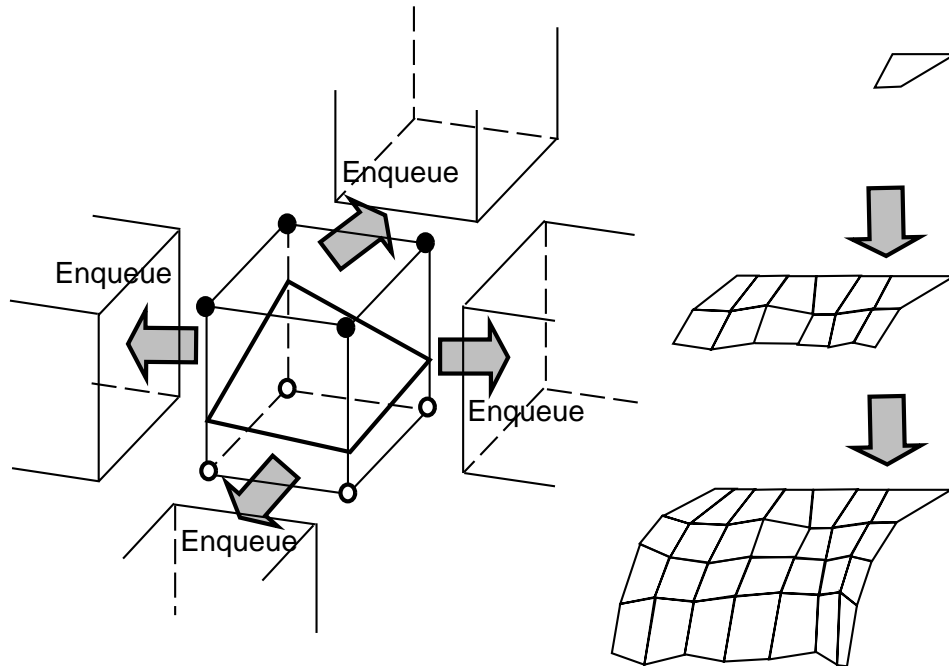


Figure 5.7: 自己増殖的な等値面生成手法

- (b) 等値面と交差する要素辺を検出する．
 - (c) 要素辺上に生成される等値面のポリゴン頂点を，ハッシュテーブルを用いて登録する．すでにハッシュテーブルにポリゴン頂点が登録されている場合には，新しくポリゴン頂点を生成せずに，登録されているポリゴン頂点を用いる．
 - (d) マーチング・キューブ法 [43] を用いて，ポリゴン頂点を結び，ポリゴンを生成する．
 - (e) 当該要素と面を共有する隣接要素のうち，処理済マークがつけていられなくて，かつ等値面と交差する要素があれば，それをすべて FIFO キューに登録し，「処理済」マークをつける．
3. 生成された各々のポリゴン頂点について，式 5.2 を用いて座標値を算出する．
 4. 生成された各々のポリゴン頂点について，法線ベクタを算出する．

5.5 むすび

本章では，3次元数値解析結果のデータ構造であるボリューム・データを定義し，その代表的なグラフィックス表示手法である等値面生成，および等値面生成の高速化手法について，従来の研究成果を概観した．

近年活発に報告されている高速化手法の多くは，等値面と交差しない多くの要素を処理から省略する手法である．しかし，その手法のいずれもが，

- 前処理量が $O(n)$ より大きい．
- 等値面生成処理量が $O(n)$ 程度であるので，大規模なボリュームにおいて処理量が大きくなる．
- 要素が直交座標軸方向に整列していない不規則なボリューム・データへの実装が難しい

のいずれかの問題点を持っており，このすべての問題点を解決した手法はまだ報告されていない．

第 6 章

極点グラフを用いた等値面高速生成手法

6.1 はじめに

第 5 章では、ボリューム・ビジュアライゼーションの代表的な手法である等値面生成の高速化手法について列挙した。表 5.2 からわかるように、従来の高速化手法は、

- 前処理量が $O(n)$ より大きい。
- 等値面生成処理量が $O(n)$ 程度であるので、大規模なボリュームにおいて処理量が大きくなる。
- 要素が直交座標軸方向に整列していない不規則なボリューム・データへの実装が難しい

のいずれかの問題点を残していた。

本論文では、これらの高速化手法の問題点を踏まえて、等値面生成の高速化手法に下記のような要件を設ける。

要件 6.1 等値面生成の高速化手法は、ボリュームの要素数を n とした時に、前処理の処理量が 高々 $O(n)$ 程度で安定していることが望ましい。

要件 6.2 等値面生成の高速化手法は、特に大規模なボリュームにおいて、効率的に等値面を生成できることが望ましい。ボリュームの要素数を n とした時に、等値面生成の処理量が $O(n)$ より小さいことが望ましい。

要件 6.3 等値面生成の高速化手法は、非構造ボリュームに適用できることが望ましい。

本章では、極点グラフというデータ構造を用いて、等値面生成を高速化する手法を提案する [29]。

6.2 節では、極点グラフ法の概要を述べる。極点グラフ法では、まずボリューム・データ中の極大点および極小点を抽出し、それを連結するグラフを生成し、グラフの辺が通過する要素を登録する。同時に、ボリュームの境界面に接する要素を、要素頂点のスカラー値でソートして登録する。登録された要素との交差判定により、任意の等値面との交差要素が最低 1 個抽出され、自己増殖的に等値面を生成することが出来る。この手法における等値面の生成処理量は $O(n)$ 以下である。また、本手法は非構造ボリュームにも適用できる。

6.3 節では，極点グラフ法の各処理部について，詳しく示す．具体的には，極大点および極小点の抽出部，グラフの生成部，ボリュームの境界面に接する要素の登録部，等値面の生成部について示す．

6.4 節では，極点グラフ法の前処理，および等値面生成の処理量について評価する．

6.5 節では，極点グラフ法を実装して処理速度を測定して結果を示し，極点グラフ法の有用性を考察する．

6.2 極点グラフを用いた等値面生成の概要

本節では，自己増殖的な等値面生成手法の適用に必要な，等値面との交差要素の抽出方法について検討する．まず本章では，等値面を以下の2種類に分類する(図 6.1 参照)．

定義 6.1 本論文では，すべての面素の辺が2枚の面素によって共有されている等値面を，閉曲面 (Closed isosurface) と定義する．

定義 6.2 本論文では，1枚の面素のみに接し，ボリュームの境界面に接する面素の辺が存在する等値面を，境界つき曲面 (Isosurface with boundary) と定義する．

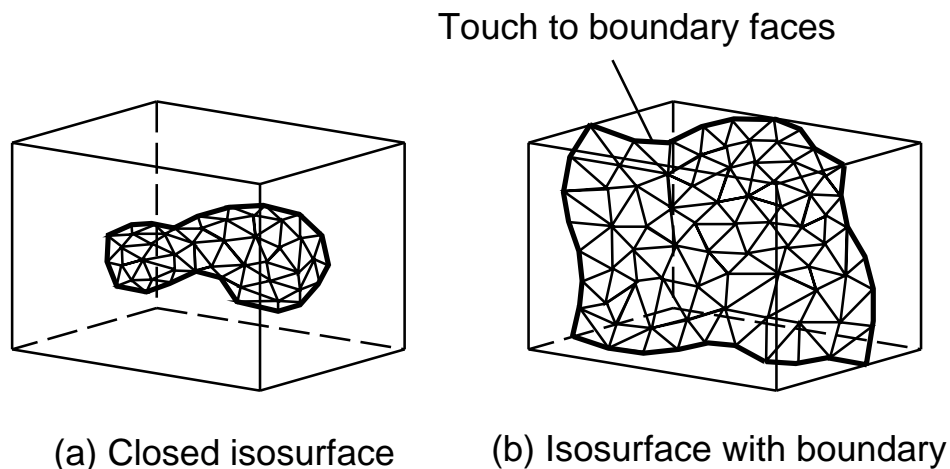


Figure 6.1: 閉曲面および境界つき曲面となる等値面の定義

この定義により，境界つき曲面となる等値面との交差要素は，ボリュームの境界面に接する要素との交差判定により抽出することが出来る．

続いて、閉曲面となる等値面との交差要素の抽出方法を導くために、以下の定理を用いる (定理の証明は本章の付録を参照)。

定理 6.1 等値面が閉曲面を形成し、閉曲面に囲まれる領域、およびボリューム中の閉曲面外部の領域が体積を有するとき、閉曲面内部および外部の領域には、最低 1 個の極大点、または極小点が存在する。ただし、領域内部のスカラー値が一定値である場合を除く。

定理 6.2 ボリューム内の極大点および極小点をあらかじめ抽出し、それらをすべて連結するグラフを生成すると、最低 1 本のグラフの辺は、閉曲面を形成する等値面と交差する

図 6.2 (a)(b) は、定理 6.2 を、2 次元での問題に単純化して図解したものである。図 6.2(a) は、2 次元のスカラー場の等高線、および極大点と極小点を示したものである。太線で示した、閉曲線となる 2 本の等高線の内側および外側には、最低 1 個の極大点または極小点が存在することが示されている。図 6.2(b) は、図 6.2(a) に示される極大点および極小点を連結するグラフを生成した例である。このグラフは、太線で示した 2 本の等高線と、1 個以上の交点を持つことが示されている。

図 6.2(c) は、図 6.2 (b) を 3 次元に拡張したものである。この図においても、極大点および極小点を連結するグラフは、太線で示した 2 曲面を形成する等値面のそれぞれの部分と、1 個以上の交点を持つことが示されている。

本論文では、閉曲面となる等値面との交差要素を抽出するために、上記の定理を利用した極点グラフというデータ構造を用いる。ここで本論文ではまず、ボリューム・データにおける極大点および極小点を、下記の通り定義する。

定義 6.3 本論文では、要素頂点 N のスカラー値が、 N に接するすべての要素 E において、 E がもつすべての要素頂点 N_i のスカラー値よりも大きい時、 N を 極大点 (Local maximum point) と定義する。

定義 6.4 本論文では、要素頂点 N のスカラー値が、 N に接するすべての要素 E において、 E がもつすべての要素頂点 N_i のスカラー値よりも小さい時、 N を 極小点 (Local minimum point) と定義する。

定義 6.5 本論文では、極大点および極小点の総称を、極点 (Extremum point) と定義する。

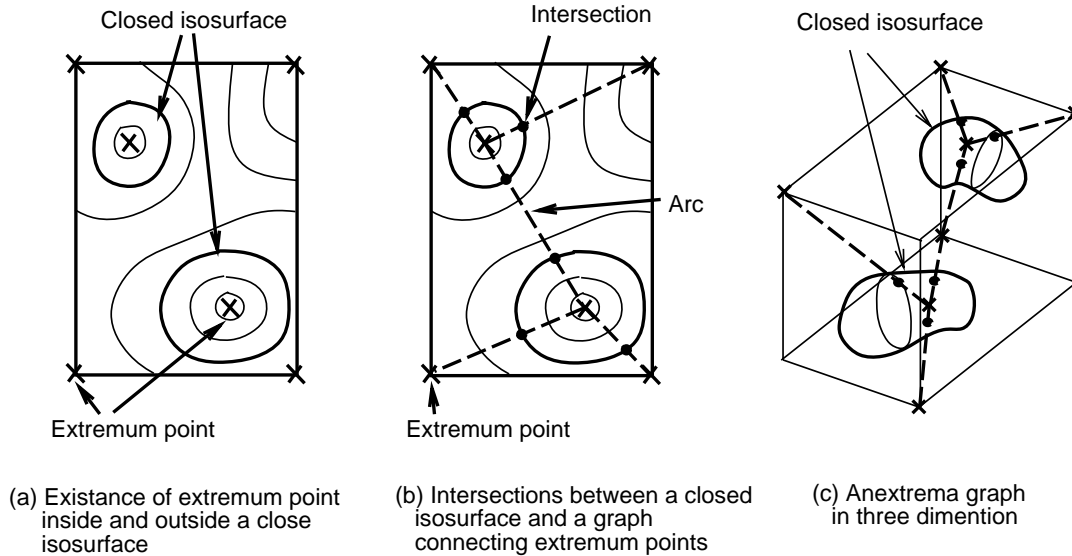


Figure 6.2: 等値面と極点に関する定理

続いて、等値面との交差要素を抽出するために本章で構築するデータ構造を、下記の通り定義する。図 6.3 は、極点グラフおよび境界面要素リストを図解したものである。

定義 6.6 本論文では、ボリューム内の極点をすべて連結するグラフを生成し、グラフの辺が交差する要素を要素リストに保存したデータ構造を、極点グラフ (Extrema graph) と定義する。

定義 6.7 本論文では、ボリュームの境界面に接する要素を要素リストに保存したデータ構造を、境界面要素リスト (Boundary element list) と定義する。

本章では、極点グラフおよび境界面要素リストを生成する処理を「前処理」と位置付ける。前処理では、まずボリューム内の極大点および極小点を抽出する。続いて、抽出された極大点および極小点のうち、距離の近い点を連結する辺を生成し、この処理をすべての頂点が連結されるまで反復する。辺が通過する要素は、それぞれの辺の要素リストに保存される。同時に、ボリュームの境界面に接する要素を抽出し、境界面要素リストに保存する。本論文では、交差要素抽出の高速化のために、境界面要素リストに登録された要素を、要素の頂点がもつスカラ値の最小値、および最大値でソートし、2種類のリストとして保存する。以上の前処理は、1個のボリューム・データに対して1回限り実行される。

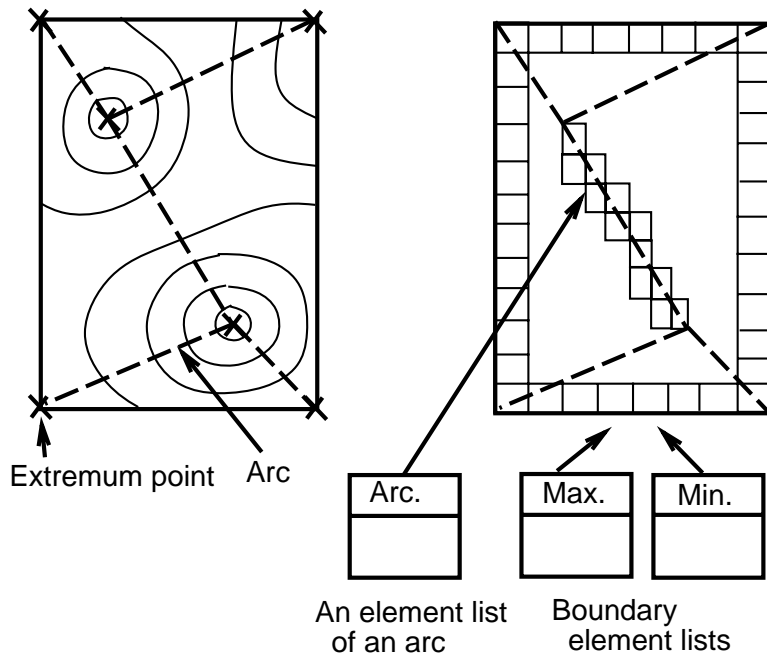


Figure 6.3: 本章で生成される要素リストの構成

本論文では、等値面のスカラ値が指定されたときに、極点グラフおよび境界面リストに登録された要素に対して、等値面との交差判定を実行する。等値面が閉曲面を形成するときには、定理 6.2 より、極点グラフの辺に登録された要素から最低 1 個の交差要素が抽出される。等値面が境界つき曲面を形成するときは、境界面要素リストに登録された要素から、最低 1 個の交差要素が抽出される。抽出された要素を出発点とすることで、等値面は自己増殖的に生成される。本論文では以上の処理を「本処理」と位置付ける。

図 6.4 は、以上の前処理および本処理の概要を、疑似コードで示したものである。

本手法は、要素間の隣接関係が保持されてあるボリュームであれば、構造ボリュームにおいても非構造ボリュームにおいても有用な手法である。本論文では、非構造格子ボリュームを主な対象として以後の議論を進める。

```

void main(){

    /* Begin Pre-process */
    Extract extremum points;
    Generate an extrema graph;
    Generate boundary element lists;
    /* End Pre-process */

    /* Begin Main-process */
    while( 1 ){
        Specify the scalar value  $C$ ;
        Visit elements registered in the extrema graph;
        if( intersected elements are found )
            Propagate an isosurface;
        Visit elements registered in boundary element lists;
        if( intersected elements are found )
            Propagate an isosurface;
        if( no more isosurfaces are required ) break;
    }
    /* End Main-process */

}

```

Figure 6.4: 極点グラフ法を用いた等値面生成手法の概要

6.3 極点グラフを用いた等値面生成手法の各処理

6.3.1 極大点・極小点の抽出

本章で提案する手法に必要な要素頂点は、定義 6.3 および定義 6.4 にしたがって抽出される。

まず、それぞれの要素ごとに、スカラ値が最も小さい要素頂点以外のすべての要素頂点に「not-minimum」というマークをつける。また、スカラ値が最も大きい要素頂点以外のすべての要素頂点に「not-maximum」というマークをつける。要素頂点に一度つけられたマークは、他の要素の処理においても消されることはないものとする。この処理がすべての要素について終わった後に、「not-maximum」のマークがつけられていない要素頂点を抽出することで、極大点を抽出することができる。同様に、「not-minimum」のマーク

がつけられていない要素頂点を抽出することで、極小点を抽出することができる。

このアルゴリズムでは、多数の同値を持つ隣接要素頂点のグループが、すべて極点として抽出される可能性がある。本手法ではこのグループのうち1点だけが抽出されれば十分であり、これらの隣接要素極点がすべて極点として抽出されることは処理量の点で好ましくない。本論文では、それぞれの要素の要素頂点について、スカラ値が最小または最大である要素頂点が複数ある場合には、それらのすべての要素頂点に「plural-minimums」「plural-maximums」という別のマークをつける。この処理がすべての要素について終わった後に、「plural-minimums」または「plural-maximums」のいずれか片方のマークをされた要素頂点が見つかったら、その要素頂点を極点として登録し、「plural-minimums」または「plural-maximums」のマークを消す。さらに、この要素頂点に隣接するひとつの要素を出発点として、隣接要素を順時探索し、この要素頂点と同値を持つ要素頂点の「plural-minimums」または「plural-maximums」のマークを消す。以上の操作により、同値を持つ要素頂点のうち1点のみが、極点として登録される。

図 6.5 は、本手法による極点抽出のアルゴリズムを、擬似コードで示したものである。

6.3.2 極点グラフの生成

本節で示す極点グラフの生成手法では、ポリウム中の2極点を選択し、これを連結する辺を生成する。この処理を、すべての極点が連結されるまで反復することで、極点グラフを生成する。

ここで、等値面生成の処理量や、メモリ使用量の観点から、グラフのコストが小さくなること、つまり極点グラフに登録される要素数が少なくなることが好ましい。そこで本手法では、極点グラフのコストについて、下記の通り仮定する。

仮定 6.1 距離の近い2極点を選択して連結することで、極点グラフのコストが小さくなる可能性が高い。

本手法では上記の仮定に基づいて、距離の近い極点を連結する。まず、任意の極点1個を「出発点 (Start point)」として選択する。続いて、出発点以外の極点を出発点からの距離でソートする。ソートされた極点のうち、最も距離の近い極点を「到達点 (Goal point)」として選択し、2点を連結する辺を生成する。

```

void ExtractExtremumPoints(){
    for( each element ) {
        Mark non-maximum nodes;
        if( plural unmarked nodes )
            Mark plural-maximums nodes;
        Mark non-minimum nodes;
        if( plural unmarked nodes )
            Mark plural-minimums nodes;
    }
    for( each node ) {
        if( Unmarked as non-maximum )
            Register the node;
        if( Unmarked as non-minimum )
            Register the node;
        if( Marked as plural-maximums or plural-minimums ) {
            Register the node;
            Unmark the adjacent plural-maximums or plura-minimums nodes;
        }
    }
}

```

Figure 6.5: 極点の抽出アルゴリズム

本手法では、極点を連結する辺を生成する過程において、それぞれの極点 X_i に、連結性を表す番号 C_i を記録する。まず、辺の生成処理を開始する前に、それぞれの極点の極点番号をつける。2 極点 X_i と X_j が辺で連結された時に、 X_i と X_j につけられた番号 C_i と C_j を比較し、小さい方の番号を両者につける。さらに、大きい方の番号をつけられていた他のすべての極点の番号を、小さい方の番号に変更する。例えば $C_i < C_j$ である場合には、 X_j の番号を C_i に修正すると同時に、他のすべての極点につけられた番号を参照し、番号 C_j がつけられていた極点の番号をすべて C_i に修正する。(図 6.6 参照)。

この処理によって、同一番号をつけられた極点は、1 本または複数本の辺によって、ひと続きに連結されたことが示されている。出発点と到達点の選択の際には、この番号を比較し、番号の異なる極点を選択することで、連結されていない 2 極点を選択することが出来る

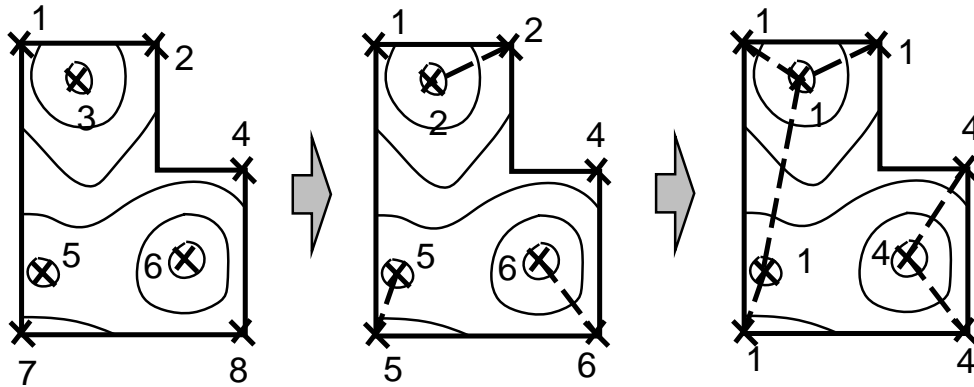


Figure 6.6: 極点の連結性の判定

出発点と到達点が決まったら，この2極点を連結する辺を生成する．ここで本手法では，極点グラフを構成する辺のデータ構造を，下記の通り定義する．

定義 6.8 本章では，極点グラフの辺 A は，両端の極点 $\{X_1, X_2\}$ ， A が通過する要素 $\{E_1, E_2, \dots\}$ ， A が通過するすべての要素がもつ要素頂点のスカラー値の最大値および最小値 $\{S_{max}, S_{min}\}$ より構成されるものとする．

本手法ではまず，出発点から到達点への半直線の方方向ベクタを算出し，半直線を通過する要素を，出発点から隣接順に探索する (図 6.7 参照)．この処理は，到達点を要素頂点にもつ要素に到達するまで反復される．この処理を通して，探索された要素 E_i はすべてグラフの辺 A に登録される．同時に，探索された要素 E_i の要素頂点 N_1, N_2, \dots がもつスカラー値と，グラフの辺 A の S_{max}, S_{min} との大小比較が実行される．

凹部や空洞をもつボリュームでは，半直線が到達点よりも前でボリューム外部に出ることがある．ボリューム外部に線分が出る時には，要素の探索はボリューム境界面に到達し，隣接要素の探索が停止してしまう．このような場合には，出発点からの距離でソートされた極点を参照し，次に距離が近く，かつ異なる番号をつけられている極点を「到達点」に選択して，同様な処理を行なう (図 6.8 参照)．

ボリューム形状が複雑な場合には，どの極点とも線分で連結することができない極点があり得る．このような場合には，極点を直線ではなく折れ線で連結することを考える (図 6.9 参照)．まず，最も距離が近く，かつ異なる番号をつけられている極点を，再び「到達点」として選択する．出発点に隣接する要素 E_0 を出発して， E_0 のもつすべての要素

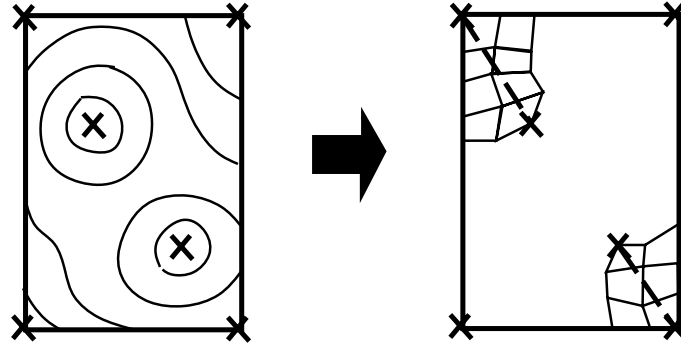


Figure 6.7: 直線によるグラフ辺の生成

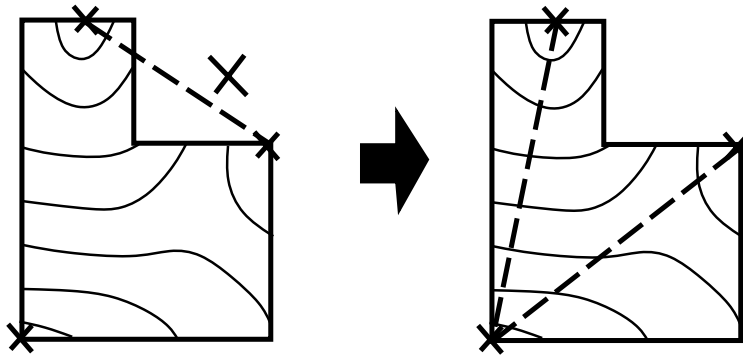


Figure 6.8: 直線によるグラフ辺生成が出来ない場合

頂点 N_1, N_2, \dots について，到達点までの距離 $dist_i$ を算出する．続いて， E_0 の面のうち，境界面でない面について， $dist_i$ の総和 $\sum dist_i$ を求め， $\sum dist_i$ の値が最も小さい面 F_{min} を特定する． F_{min} を共有する隣接要素 E_1 を探索して，同様な処理を反復する．この処理を，到達点を要素頂点にもつ要素が探索されるまで反復する．線分上の探索と同様に，この処理を通して，探索された要素 E_i はすべてグラフの辺 A に登録される．同時に，探索された要素 E_i の要素頂点 N_1, N_2, \dots がもつスカラ値と，グラフの辺 A の S_{max}, S_{min} との大小比較が実行される．この処理では，同一要素が複数回探索される可能性がある．そこで，1度探索された要素にはマークをつけ，マークされていない隣接要素によって共有されている面のみに対して $\sum dist_i$ を算出する．

折れ線による極点連結処理は，必ずしもすべての極点の連結を保証するものではない．

例えば，要素探索の経路が渦巻状になり，マークされた要素に囲まれた要素に到達して探索が行き止まる可能性がある．このような場合には，探索を折り返してマークされていない要素を探することで，別の経路によって連結することができる．本論文では，その処理については議論しないものとする．しかし，本論文の 6.5 節に示す実行例では，すべての例において最初に選択した到達点との連結を実現している．よってこの処理は，非常に高い可能性で 2 極点を連結することがわかる．

折れ線による極点連結処理は，探索されたすべての要素 E_i の要素頂点 N_1, N_2, \dots において，到達点との距離を算出する．この処理には平方根の算出を要するので，処理量が大きくなることが多い．そのためこの処理は，線分による極点連結が不可能な場合にだけ適用することが望ましい．

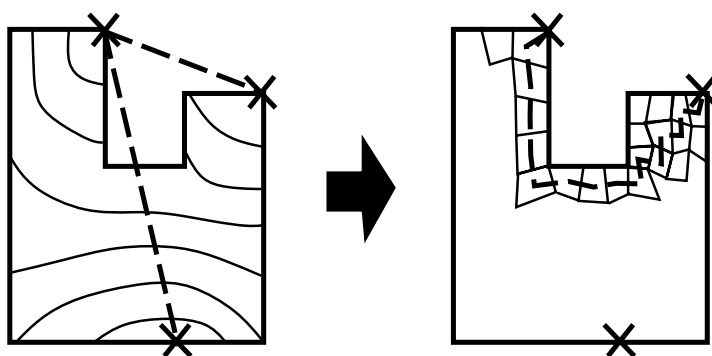


Figure 6.9: 折れ線によるグラフ辺の生成

図 6.10 は，本手法による極点グラフ生成のアルゴリズムを，擬似コードで示したものである．

6.3.3 境界面要素リストの生成

本章では，境界面要素リストを生成するために，境界面に接する要素のデータ構造を，下記の通り定義する．

定義 6.9 本章では，境界面に接する要素 E を，要素頂点 $\{N_1, N_2, \dots\}$ および隣接要素 $\{E_1, E_2, \dots\}$ の他に， E がもつ要素頂点のスカラ値の最大値 S_{min} および S_{max} より構成されるものとする．

```

void GenerateExtremaGraph(){
    for (each extremum point) {
        status=FALSE;
        sort other different-numbered extremum points;
        for( each extremum point in the sorted extremum point list) {
            if( succeed to connect two points by a straight arc ) {
                status=TRUE; break;
            }
        }
        if( status==FALSE ) {
            for( each extremum point in the sorted extremum point list) {
                if( succeed to connect two points by a polygonal arc ) {
                    status=TRUE; break;
                }
            }
        }
    }
}

```

Figure 6.10: 極点グラフの生成アルゴリズム

本手法ではまず、境界面に接する各要素 E に対して、 S_{max} および S_{min} を算出する。続いて、境界面に接する要素を S_{max} でソートしたリスト $Blist_{max}$ を生成する。同時に、境界面に接する要素を S_{min} でソートしたリスト $Blist_{min}$ を生成する。ここで、2面以上の境界面をもつ要素が存在することがあるので、 $Blist_{max}$ 上に同一要素が重複しないように、 $Blist_{min}$ 上においても同様に同一要素が重複しないように注意する。

6.3.4 自己増殖的手法を用いた等値面生成

本論文では、前節までに示された、極点抽出・極点グラフ生成・境界面要素リスト生成の各処理を、「前処理」と位置づける。この前処理は、1個のボリューム・データに対して、1回だけ実行される。本節で説明する「本処理」は、等値面のスカラ値が指定されるたびに、前処理にて生成された極点グラフおよび境界面要素リストが参照されて実行される。

本手法による等値面生成のアルゴリズムを、下記に示す。

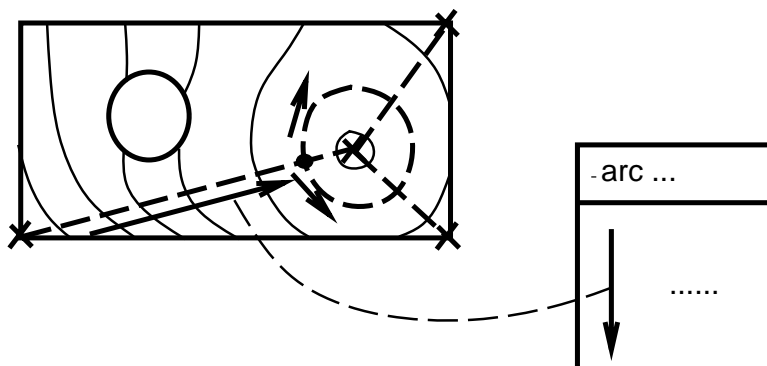
1. すべての要素を探索し，直前の等値面生成処理でいくつかの要素につけられた「処理済」マークを消す．
2. 等値面のスカラ値 C を決定する．
3. 極点グラフの各辺 A について，以下の処理を実行する．
 - (a) A に登録された S_{max} および S_{min} と C を比較する． $S_{max} < C$ または $S_{min} > C$ である場合には，以下の処理を実行しない．
 - (b) A に登録されている要素 E_i のうち「処理済」マークのつけられていない要素に対して，等値面との交差判定を実行し「処理済」マークをつける．
 - (c) 等値面との交差要素が検出されたら，その要素を出発点にして，自己増殖的に等値面を生成する．
4. 境界面要素リストを用いて，以下の処理を実行する．
 - (a) C がボリュームの最大値に近い場合には， $Blist_{max}$ 用いて， $S_{max} < C$ である要素を抽出する．抽出された要素のうち「処理済」マークのつけられていない要素に対して，等値面との交差判定を実行し「処理済」マークをつける．
 - (b) C がボリュームの最小値に近い場合には， $Blist_{min}$ 用いて， $S_{min} > C$ である要素を抽出する．抽出された要素のうち「処理済」マークのつけられていない要素に対して，等値面との交差判定を実行し「処理済」マークをつける．
 - (c) 等値面との交差要素が検出されたら，その要素を出発点にして，自己増殖的に等値面を生成する．

なお，自己増殖的な等値面生成については，本論文では 5.4.2 節に示されたアルゴリズムを用いている．5.4.2 節に示されたアルゴリズム中では，等値面と交差する要素に「処理済」マークがつけられる．

図 6.11 (a) は，閉曲面を形成する等値面の生成例である．極点グラフの辺が通過する要素に対して，等値面との交差判定が実行される．点線で表現されるような，閉曲面となる等値面は，極点グラフの最低 1 本の辺と交差するので，矢印に示すような方向で自己増殖的に生成される．

図 6.11 (b) は，開曲面を形成する等値面の生成例である．点線で表現されるような，境界つき曲面となる等値面は，極点グラフの辺と交点をもたない場合にも，境界面要素リストに登録された最低 1 個の要素と交差するので，矢印に示すような方向で自己増殖的に生成される．

(a) Search for on arcs



(b) Search for on boundary

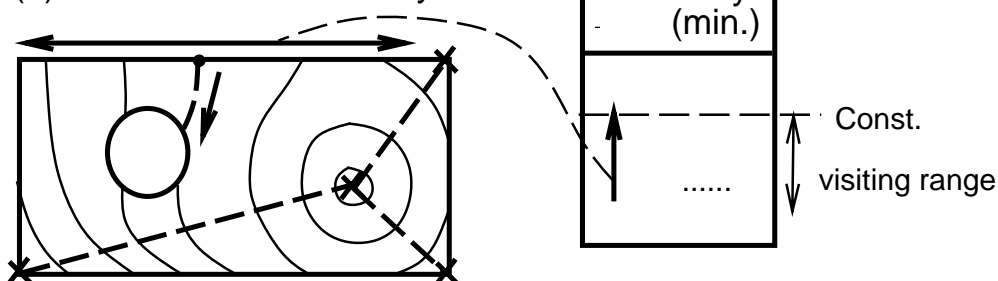


Figure 6.11: 等値面との交差要素の抽出例

6.4 極点グラフを用いた等値面生成手法の処理量

本章で報告する等値面生成手法の処理量を，前処理と本処理にわけて考察する．ここで，要素の総数を n とし，要素頂点の総数は要素の総数に比例するものとする．

前処理は，極点抽出部・極点グラフ生成部・境界面要素リスト生成部の各部分に分けることができる．極点抽出部では，すべての要素を 1 回ずつ処理し，その後すべての要素頂点を 1 回ずつ処理するので，処理量は $O(n)$ である．極点グラフ生成部では，グラフの

辺が直線または直線に近い折れ線である場合には、仮定 5.1 より、 $O(n^{1/3})$ の要素数を探索する。よってこの時、処理量は $O(n^{1/3})$ である。その他に、極点を距離でソートする処理の総処理量は、極点数を m とすると $O(m^2 \log m)$ である。多くのデータにおいて、極点数は要素数よりもはるかに小さいので、この処理量は十分小さいことが多い。しかし極点数が多い場合には、この処理量が加速的に増加する。境界面要素リスト生成部では、境界面が平面または平面に近い面の集合である時には、仮定 5.1 より、境界面要素リストに登録される要素数は $O(n^{2/3})$ である。この時、その処理量は $O(n^{2/3} \log n^{2/3})$ である。

以上の評価により、前処理の処理量の総計は $O(n) + O(n^{1/3}) + O(m^2 \log m) + O(n^{2/3} \log n^{2/3})$ であると評価される。通常は m は n にくらべて無視できるくらい小さいことが多く、その場合には要件 6.1 を満たすことができる。しかし、 m が大きくなるにつれて処理量は加速的に大きくなり、要件 6.1 を満たすといえないほど大きな処理量となることもある。

本処理の処理量は、探索される要素数に比例する。極点グラフに登録された要素数は、グラフの辺が直線または直線に近い折れ線である場合には、仮定 5.1 より、 $O(n^{1/3})$ の要素数を探索する。よってこの時、処理量は $O(n^{1/3})$ である。境界面要素リストに登録された要素数は、境界面が平面または平面に近い面の集合である時には、仮定 5.1 より $O(n^{2/3})$ である。よってこの時、処理量は $O(n^{2/3})$ である。自己増殖的手法によって処理される要素数は、等値面を形成する要素の総数である。等値面が平面またはそれに近い面である時には、仮定 5.1 より、その要素数は $O(n^{2/3})$ である。よってこの時、処理量は $O(n^{2/3})$ である。

以上の評価により、本処理の処理量は、 $O(n)$ より低いと評価されるので、要件 6.2 を満たしたということがいえる。

6.5 実行例

本節では、本手法を実装して実行した結果を示す。なお、計算機は IBM PowerStation RS/6000 (Model 560) を用いた。

図 6.12 および 図 6.13 は、極点グラフを生成した例を示したものである。図 6.14 および 図 6.15 は、各々図 6.12 および 図 6.13 に示される極点グラフを用いて、等値面を生成した例である。図中の紫色の十字が極点を表している。また、グラフの辺の色、および等値面の色は、スカラー値の大小を表している。

ここで、極点グラフ法による等値面生成の処理速度と、全要素に対して交差判定を実行する単純な手法による処理速度、要素のソート [21] および階級化 [17] を用いた手法による処理速度を比較する。本節に示す処理速度測定では、有限要素法の数値解析結果によって得られる、四面体要素で構成される 4 種類の非構造ボリュームを用いて、20 枚の異なるスカラ値をもつ等値面の生成に要する処理量を測定した。

表 6.1 は、処理速度測定に用いた 4 種類のボリューム・データの規模を示したものである。

Table 6.1: 処理速度測定に用いるボリューム・データ

データ	1	2	3	4
N_e	20736	61680	346644	557868
N_b	2048	5432	16908	17876
N_n	4002	11624	62107	97473

ここで、表中のそれぞれの変数は、

N_e : ボリューム・データ中の要素の総数

N_b : ボリュームの境界面に接する要素の総数

N_n : ボリューム・データ中の要素頂点の総数

を示すものである。

表 6.2 は、各々の手法が要する前処理の処理時間を示したものである。

Table 6.2: 前処理の処理時間

データ	1	2	3	4
T_{sort}	1.228	6.259	38.027	58.579
T_{filt}	0.285	0.647	4.084	6.589
T_{egra}	0.684	1.823	8.028	15.229

ここで、表中のそれぞれの変数は、

T_{sort} : 要素をソートする手法で要する処理時間

T_{filt} : 要素を階級化する手法で要する処理時間

T_{egra} : 極点グラフ法で要する処理時間

を示すものである。

表 6.2 は、極点グラフ法における前処理の処理時間は、要素をソートする手法よりも小さく、要素を階級化する手法よりも大きいことが示されている。なお、全要素に対して交差判定を実行する手法では、前処理は不必要である。

表 6.3 は、極点グラフ法における前処理の、各々の処理に要する処理時間を示したものである。

Table 6.3: 極点グラフ法の前処理の処理時間

データ	1	2	3	4
T_{pe}	0.541	1.382	6.290	9.719
T_{pc}	0.062	0.166	0.600	4.151
T_{pb}	0.081	0.275	1.138	1.359
N_e	21	46	135	540
N_g	468	1048	3250	10210
N_{pa}	1	0	16	36

ここで、表中のそれぞれの変数は、

T_{pe} : 極点の抽出に要する処理時間

T_{pc} : 極点グラフの生成に要する処理時間

T_{pb} : 境界面要素リストの生成に要する処理時間

N_e : 極点の総数

N_g : 極点グラフに登録された要素の総数

N_{pa} : 折れ線連結によって生成された極点グラフの辺の総数

を示すものである。

表 6.3 によると、極点グラフ法の前処理では、極点の抽出に大きな処理量を要していることが示されている。極点の抽出処理では、ボリューム・データ中のすべての要素および要素頂点が 1 回ずつ処理されるので、ボリューム・データ中の一部の要素のみが処理される他の処理部と比較して、多くの処理量を要することが多い。

また、極点グラフに登録される要素数 N_g は、境界面要素リストに登録される要素数 N_{bc} よりも大幅に少ないことがわかる。

データ No. 4 における T_{pc} の値は、他のボリューム・データと比較して極端に大きい。この最も大きな理由として、出発点と到達点を選択するために、極点を距離でソートする処理量の増加が考えられる。データ No. 4 の極点数 N_e は、他のボリューム・データよりも大きく、そのために極点を距離でソートする処理量が加速的に増加している。もうひとつの理由として、折れ線による辺の生成の処理量の増加が考えられる。データ No. 4 の折れ線による辺の数 N_{pa} は、他のボリューム・データよりも大きくなっている。このことから、極点グラフ法の前処理の処理時間は、要素数だけでなく、ボリューム・データの極点数や形状などの、より複雑な要因に大きく依存することがわかる。このことは、極点グラフ法の前処理の処理量は、単純には見積もりにくく、処理量が不安定であるかのような印象をうけやすい、という問題点を示している。

続いて、等値面生成処理の処理時間を比較する。表 6.4 は、各手法を用いて 20 枚の等値面を生成した総処理量を示したものである。

Table 6.4: 等値面生成処理の処理量

データ	1	2	3	4
T_{all} (sec.)	5.068	11.416	63.113	103.072
T_{sort} (sec.)	2.696	3.142	12.125	20.717
T_{filt} (sec.)	2.800	3.945	17.801	22.153
T_{egra} (sec.)	2.065	2.561	6.524	11.786

ここで、表中のそれぞれの変数は、

T_{all} : 全要素に対して等値面との交差判定を実行する手法の処理時間

T_{sort} : は、要素をソートする手法の処理時間

T_{filt} : は、要素を階級化する手法の処理時間

T_{egra} : は、極点グラフ法の処理時間

を示すものである。

表 6.4 は、極点グラフ法が 4 手法の中で最も効率的に等値面を生成し、特に要素数の大きい大規模なボリューム・データにおいて他の手法よりも効率的に等値面を生成していることがわかる。

表 6.5 は、極点グラフ法における等値面生成の各処理に要する処理時間を示したものである。

Table 6.5: 極点グラフ法における等値面生成処理の処理時間

データ	1	2	3	4
T_{mg} (sec.)	0.008	0.088	0.248	0.197
T_{mb} (sec.)	0.019	0.032	0.091	0.144
T_{mt} (sec.)	1.471	1.748	4.495	8.331
T_{mv} (sec.)	0.567	0.693	1.690	3.114
N_{cg}	3063	4774	15659	14196
N_{ci}	51357	62377	148836	277457

ここで、表中のそれぞれの変数は、

T_{mg} : 極点グラフに登録された要素との交差判定の処理時間

T_{mb} : 境界面要素リストに登録された要素との交差判定の処理時間

T_{mt} : 自己増殖的な等値面生成手法によって、等値面を構成する三角形面素を生成する処理時間。

T_{mv} : 等値面を構成する三角形面素の頂点の、座標値および法線方向の算出の処理時間

N_{cg} : 等値面との交差要素の抽出のために、等値面との交差判定を実行した要素の総数

N_{ci} : 等値面と交差する要素の総数

を示すものである。

表 6.5 では、極点グラフ法における等値面との交差要素の抽出の処理量 T_{mg} および T_{mb} が、非常に小さいことが示されている。また、抽出のために交差判定を実行した要素数 N_{cg} も、等値面と交差する要素数 N_{ci} と比較して非常に小さいことが示されている。

6.6 むすび

本章では、ボリューム・データ中の極大点および極小点を連結するグラフの辺、およびボリュームの境界面から、等値面と交差する要素を抽出し、自己増殖的に等値面を生成する、高速な等値面生成手法を提案した。

本章で提案した極点グラフ法は、本章の冒頭に示した要件のうち、要件 6.2、要件 6.3 の 2 点を満たす手法であることが示された。一方、要件 6.1 については、極点が極端に多くない一般的なデータでは満たされるが、極点の増加とともに前処理量が加速的に増加するので、必ずしも要件を満たすとは限らない。

付録

(定理 6.1 の証明)

中間値の定理により、有限な領域に分布する連続な関数は、その数値が定数でない限り、最大値および最小値を持つ。よって、閉曲面に囲まれる領域、およびボリューム中の閉曲面外部の領域には、最大値および最小値が存在する。ただし、最大値または最小値の一方が等値面のスカラー値に一致する場合には、その値はボリューム・データ中の極大値や極小値でない場合もある。よって、各領域には、極大点または極小点の両方の存在は保証されないが、少なくとも一方の存在は保証される。

(定理 6.2 の証明)

定理 6.1 により、極大点および極小点は閉曲面となる等値面の内部領域および外部領域の双方に存在する。もしこれらをすべて連結するグラフが閉曲面となる等値面とまったく交差しないとしたら、そのグラフは等値面の内部領域と外部領域で非連結な 2 つのグラフになってしまうので、矛盾が生じる。よって、ボリューム内の極大点および極小点をすべて連結するグラフの最低 1 本の辺は、閉曲面となる等値面と交差する。

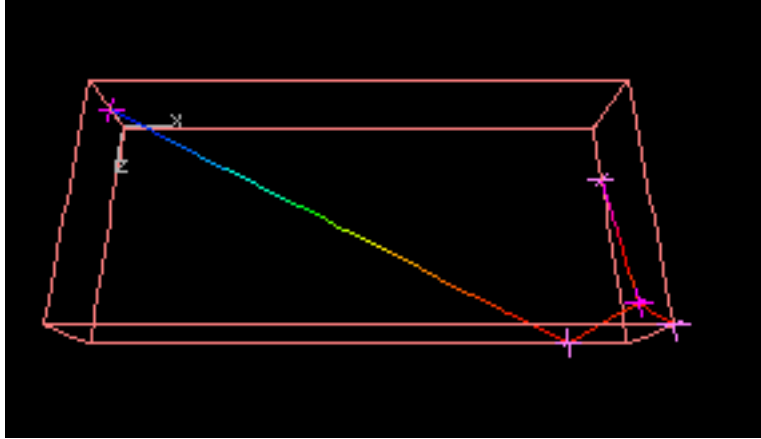


Figure 6.12: 極点グラフの生成例 (1)

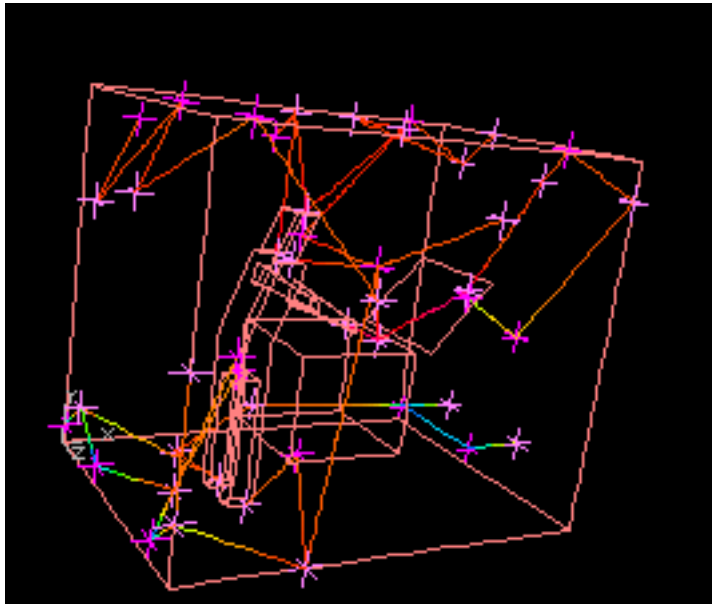


Figure 6.13: 極点グラフの生成例 (2)

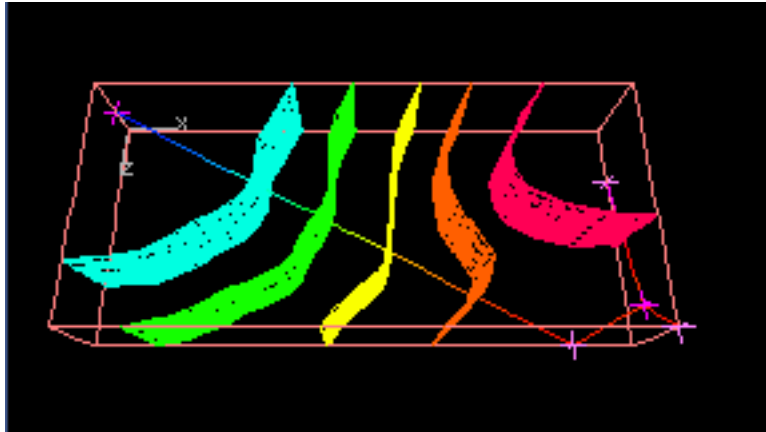


Figure 6.14: 極点グラフを用いた等値面生成例 (1)

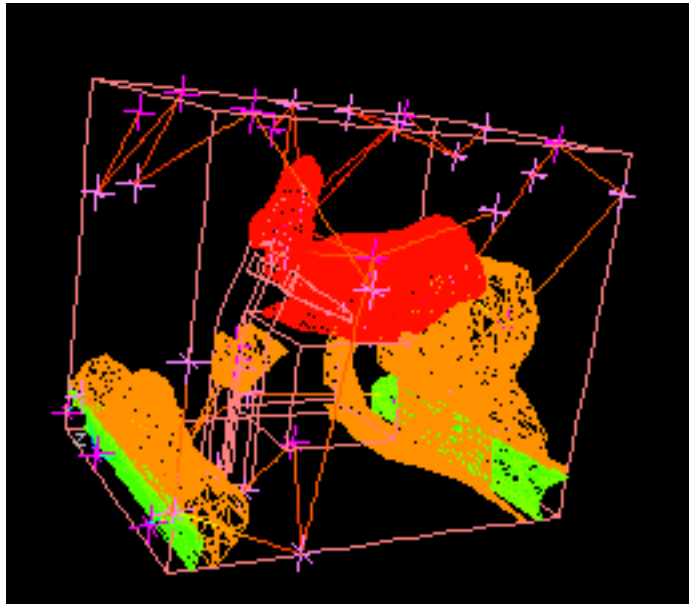


Figure 6.15: 極点グラフを用いた等値面生成例 (2)

第 7 章

細線化手法を用いた等値面高速生成手法

7.1 はじめに

第6章で提案した極点グラフ法は、要件6.2および要件6.3を満たすものの、要件6.1については、

- 極点数が多いボリューム・データ、極点グラフの生成に折れ線連結を多用するボリューム・データでは、前処理量が加速的に増加する。一般にボリューム・データの要素数は利用者にとって既知であるが、極点数は既知ではない。よって利用者にとって、前処理量の見積りが難しいので、前処理量が不安定であるような印象をうけやすい。

という問題点があった。

本章では、極点グラフ法よりもさらに高速に等値面を生成し、かつその前処理量が安定している手法として、画像認識の分野で用いられている細線化手法に類似したボリュームの細線化手法 [30] を提案する。

7.2節では、極点グラフ法における、境界面要素リストに登録された要素との交差判定の必要性について述べる。ボリュームの貫通穴の個数を保持する極点グラフを生成することが出来れば、等値面の非連結なすべての曲面は、極点グラフの少なくとも1辺と交差する。この時、境界面要素リストに登録された要素との交差判定は不要である。

本章では、ボリュームの貫通穴の種数を保持する極点グラフを生成するために、画像認識で用いられている細線化手法に類似した手法をボリューム・データに適用することを考える。7.3節では、画像認識における細線化手法の概要を示す。7.4節では、細線化手法に類似した手法をボリューム・データに適用することで、ボリュームの種数を保持しながら極点を連結するボリュームの骨格が生成できることを示す。7.5節では、ボリュームの細線化手法の各処理について、詳しく示す。

7.6節では、本手法における前処理、および等値面生成の処理量について評価する。

7.7節では、本手法を実装して処理速度を測定して結果を示し、本手法の極点グラフ法に対する有用性を考察する。

7.8節では、本手法を適用したボリューム・レンダリングの高速化手法を示す。まず、ボリューム・レンダリングの従来研究成果を概観する。続いて、視点からの半透明な等距離面を重ね描きするボリューム・レンダリング手法に、本手法を適用した高速化手法を示す。

7.2 ボリュームデータと等値面の位相的性質

前章で示した通り，極点グラフ法では，等値面との交差要素の抽出のために，極点グラフに登録された要素と，境界面要素リストに登録された要素に対して，等値面との交差判定を実行している．本節では，境界面要素リストに登録された要素の交差判定の必要性について考える．

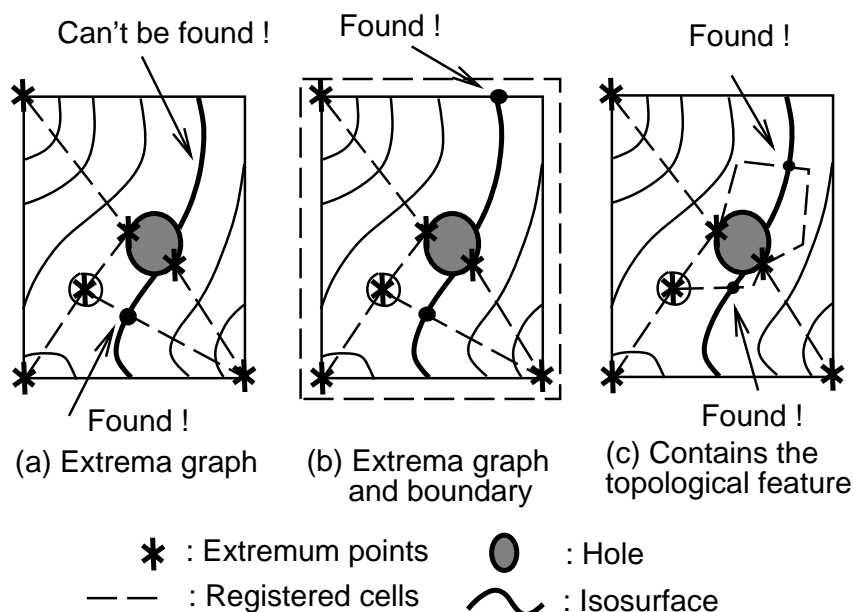


Figure 7.1: 貫通穴をもつボリュームにおける極点グラフと等値面の位相的性質

図 7.1 (a) は，貫通穴をもつボリュームの断面と，その断面上の等値面を表したものである．この例では，確かに極点グラフはすべての極点を連結しているが，貫通穴によって分断された等値面のすべての部分とは交差していない．分断された等値面のすべての部分に対して交差要素を抽出するためには，図 7.1 (b) に示す通り，境界面に接する要素との交差判定が必要である．逆にいえば，図 7.1 (c) に示すような，貫通穴の周囲を一周するように極点を連結する極点グラフがあれば，貫通穴によって分断された等値面のすべての部分と交差する辺が存在する．このような場合には，境界面要素リストに登録された要素との交差判定が不要になるので，さらに高速に等値面を生成することが出来る．

ここで，第 6 章で示した定理 6.1 を境界つき曲面にも拡張した定理を示す (本章で示

す定理の証明は、すべて本章の最後の付録に示すものとする)。

定理 7.1 境界つき等値面によってボリュームが複数の領域に分断され、その各領域が体積を有するとき、各領域には最低 1 個の極大点、または極小点が存在する。ただし、各領域におけるスカラ値が同一値である場合を除く。

また、境界つき等値面とボリュームのトポロジーについて、下記の定理を示す。

定理 7.2 複数の領域 C_1, C_2, \dots に分断された境界つき曲面である等値面によって、ボリュームが複数の領域に分断されるとする。ボリュームが貫通穴を持たない時に、分断された領域のうちの 2 領域 V_a および V_b が、等値面の非連結な複数の領域 C_1 および C_2 の両方を共有して接することはない。

定理 7.1 および定理 7.2 を用いることで、第 6 章で示した定理 6.2 を、下記のように拡張することが出来る。

定理 7.3 貫通穴を持たないボリュームに対して極点グラフを生成すると、等値面の非連結なすべての曲面は、最低 1 本のグラフの辺と交差する。貫通穴を持つボリュームにおいては、すべての曲面と極点グラフの辺との交差を保証することが出来ない。

定理 7.3 より、ボリュームが貫通穴を持たない場合には、極点グラフが等値面の非連結なすべての曲面との交差が保証されるが、ボリュームが貫通穴を持たない場合には保証されないことが示された。ここで、境界つき曲面となる等値面によって、貫通穴を持つボリュームが複数の領域に分断され、そのうちの 2 領域 V_a および V_b が等値面の非連結な複数の境界つき曲面を介して接するとする。この時、ボリュームの種数を保持しながら生成されたグラフは、2 領域 V_a および V_b に接する等値面のすべての非連結な境界つき曲面と交差する。よって、極点グラフがボリュームの種数を保持すれば、貫通穴を持つボリュームにおいても極点グラフと等値面の交差が保証される。この時に、自己増殖的な等値面生成に必要な交差要素の抽出に、境界面要素リストを用いる必要がなくなるので、等値面生成をさらに高速化できると予想される。

本論文では、ボリュームの種数を保持した極点グラフを生成するために、画像認識の分野で用いられている細線化手法に類似した手法を用いることを考える。

7.3 細線化手法

細線化手法 [49] は、白黒画像のような、2 値をもつ離散点の集合で表現される 2 次元空間において、ある同一値をもつ離散点の集合で与えられる領域内部に、その領域の骨格に相当する幅 1 の離散点の領域を生成する手法である。本来は、画像認識などの分野における、形状理解の目的で利用されることが多い。

画像認識における細線化手法では、同一値をもつ領域の境界に接する画素から順に処理を進め、骨格形成に不要と判定された画素を順に除去する。この処理は、すべての画素が 1 回ずつ処理され、残存するすべての画素がその境界に接するまで反復される。画素の必要性の判定条件には、さまざまな条件が提案されているが、代表的な判定条件の一例を下記に示す。

- 当該画素と辺を共有するすべての残存隣接画素 (最大 4 個) が、当該画素と辺または頂点を共有する残存隣接画素を通過してひと続きに探索できる場合に、当該画素を除去することが出来る。さもなければ、当該画素の除去によって残存隣接画素の連結性を失う可能性があるため、当該画素は除去されない。
- 当該画素が辺を共有する残存隣接画素を 1 個のみ持ち、かつ当該画素が頂点を共有する残存隣接画素を持たない場合には、当該画素は骨格の端点にあり、当該画素の除去によって骨格が縮小されるため、当該画素は除去されない。

上記の条件について、図 7.2 を例にして説明する。画素 p は、その残存必要性を判定される当該画素である。また、 p と辺または頂点を隣接する残存隣接画素には、便宜上 1 ~ 8 の番号がつけられているとする。

図 7.2(b)-(e) において、網づけされた画素は、除去されていない画素である。ここで、図 7.2(b)(c) では、 p を通過せずに、 p と辺を共有するすべての残存隣接画素を、番号順に通過して探索することが出来る。例えば、図 7.2(b) では、 p と辺を共有する残存隣接画素 6, 8 を、6, 7, 8 の順に探索できる。図 7.2(c) では、残存隣接画素 4, 6, 8 を、4, 5, 6, 7, 8 の順に探索できる。このような場合には、当該画素 p は除去される。一方、図 7.2(d)(e) では、辺を共有するすべての残存隣接画素を番号順にひと続きに探索することが出来ない。このような場合には、当該画素 p は除去されない。

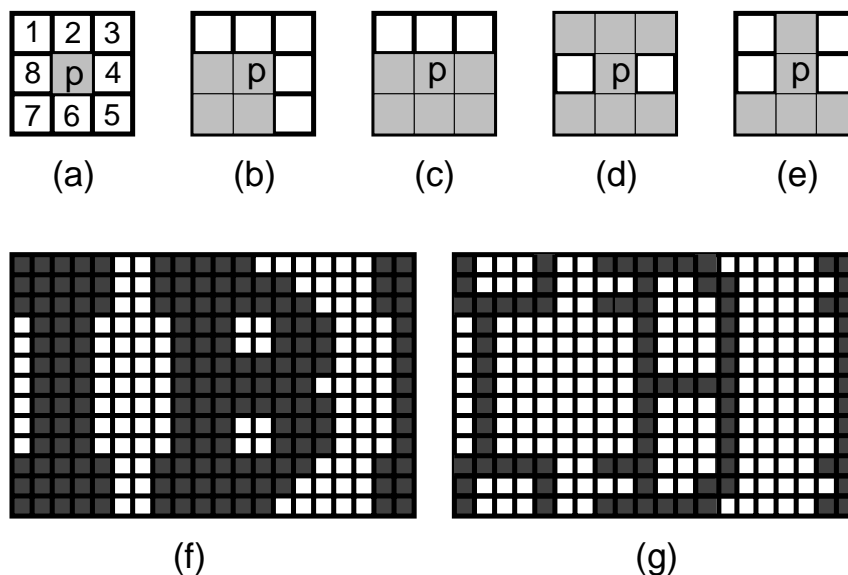


Figure 7.2: 画像の細線化処理

図 7.2 (g) は，図 7.2 (f) に示される画像から，細線化処理によって生成された骨格の例である．この骨格は，黒く塗られた領域の突起や穴などの形状的特徴を保持している．そのため，この骨格は画像の形状理解などの目的に使用することが出来る．

7.4 細線化手法を用いた等値面生成手法の概要

本節では，前節で示した細線化手法と類似したアルゴリズムで，ボリュームの種数を保存しながら極点を連結する骨格を生成する手法を報告する．本章では，この手法をボリュームの細線化手法 (Volume thinning)，または単に細線化手法 (Thinning) と呼ぶ．画像認識における細線化手法において，領域の境界に隣接する画素が順次処理されるのと同様に，本論文では非除去要素群の境界に隣接する要素を順次処理し，不必要な要素を除外してボリュームの骨格を生成する．

なお，本章で提案する細線化手法は，原理的には構造ボリュームと非構造ボリュームの両方に適用が可能である．本論文では，四面体要素または六面体要素で構成される非構造ボリュームに限定してアルゴリズムを示す．

画像認識における細線化手法では，明示的に保存したい画素を指定しない．しかし，

本章ではポリュームの極点を連結することを目的としているので，極点に接する要素が保存されなくてはならない．そこで，極点に接する要素を保存するために，下記の定義を設ける．

定義 7.1 本章で提案する細線化手法では，各々の極点に対し，極点に接する要素を1個ずつ選択し，明示的にその要素を保存する．本章では，この保存要素を指定することを，要素をマークすると呼ぶ．

ポリュームの細線化手法では，定義 7.1 を利用して，下記の定義に基づいて要素を1個ずつ除外して，骨格を生成する．

定義 7.2 本章で提案する細線化手法では，非除去要素群の境界面に接し，かつマークされていない要素を，マークされた要素の連結性と，ポリュームの種数を保持できる限り，除去するものとする．ここで，非除去要素群の境界面とは，非除去要素のもつ面のうち，他の非除去要素に接していない面を意味する．ここで，マークされた要素の連結性の保持とは，マークされた1要素を出発点にして隣接要素を探索することによって，他のマークされた要素に到達可能な状態を保持することを意味する．

図 7.3 は，本章で提案するポリュームの細線化手法を図解したものである．本手法では，まずポリューム中の極点を抽出し，極点に接する要素をマークする．続いて要素を1個ずつ除去し，要素の線列からなるポリュームの骨格を生成する．

ここで，要素の必要性を判定するために，ポリュームの細線化手法では，定義 7.5 に示される判定条件を用いるものとする．

定義 7.3 本章では，要素 E と， E の2個の隣接非除去要素 E_a および E_b の3要素に共有される辺を，三共有辺と呼ぶ．

定義 7.4 本章では，要素 E と， E の3個の隣接非除去要素 E_a ， E_b ，および E_c の4要素に共有される要素頂点を，四共有頂点と呼ぶ．

定義 7.5 本章で提案する細線化手法では，非除去要素群の境界面に接し，かつマークされていない要素 E について， E がもつすべての三共有辺および四共有頂点が非除去要素群の境界面上にない時， E を骨格から除去する．

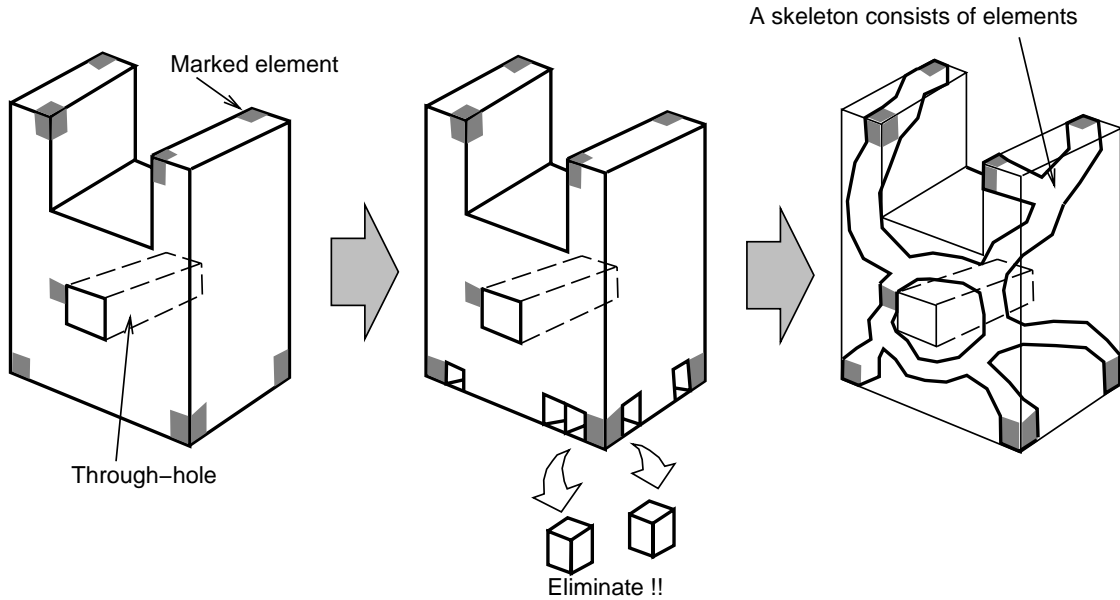


Figure 7.3: ボリユームの細線化手法

定義 7.5 に示される判定条件は、細線化手法によって生成される骨格の要素数を減らすことが考慮されている。図 7.4 は、骨格中に偶発的に生成されるループの例である。当該要素 E は、非除外要素群の境界面上に四共有頂点 N を持つ。もし E が骨格から除外されると、 N の周囲に偶発的に貫通穴が生じてしまい、ボリユームの境界面の種数を保持できない。しかも貫通穴の生成によって、骨格に残存する要素数が不必要に増加する。よって本章で提案するボリユームの細線化手法では、 E のような要素を除去しない。

続いて、定義 7.5 を用いた細線化手法が、マークされた要素の連結性、ボリユームの種数、および空洞を囲む境界面と他の境界面との非連結性を保持することを、それぞれ定理 7.4、定理 7.5、定理 7.6 に示す。

定理 7.4 非除去要素群の境界面に接し、かつマークされていない要素 E について、すべての三共有辺および四共有頂点が非除去要素群の境界面上にない時、 E を除去してもマークされた要素の連結性は保持される。

定理 7.5 非除去要素群の境界面に接し、かつマークされていない要素 E について、すべての三共有辺および四共有要素が非除去要素群の境界面上にない時、 E を除去してもボリユームの境界面の種数は保持される。

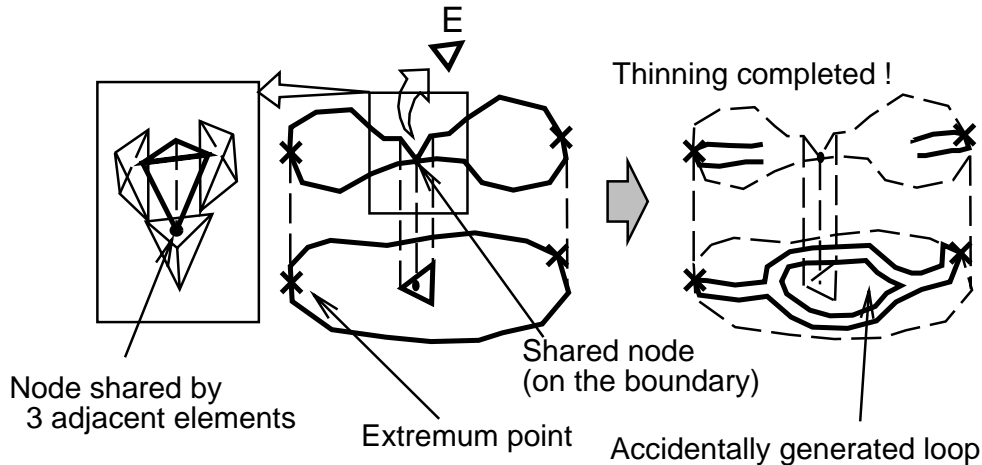


Figure 7.4: 骨格中の偶発的なループの生成

定理 7.6 非除去要素群の境界面に接し、かつマークされていない要素 E について、すべての三共有辺および四共有要素が非除去要素群の境界面上にない時、 E を除去しても空洞を囲む境界面と他の境界面との非連結性は保持される。

これらの定理より、定義 7.5 に示された要素の除去条件は、定義 7.2 に示された細線化手法の定義にしたがった除去条件であることが示された。

7.5 細線化手法を用いた等値面生成手法の各処理

本節では、ボリュームの細線化手法、および細線化手法によって生成された骨格を用いた等値面生成処理に関する、具体的な実装方法について述べる。四面体要素で構成される非構造ボリュームに適用された、ボリュームの細線化手法の疑似コードを、図 7.5 に示す。このアルゴリズムは、

1. データ構造の整備のための前処理、および極点に接する要素のマーク
2. 細線化手法による要素の除去
3. 空洞周辺に残存する層状の要素の除去 (詳細は後述)
4. 残存要素 (=骨格を構成する要素) のスカラ値による階級化

の各処理に大別される .

```
void VolumeThinning(){
  Classify and enqueue elements;
  Number boundary faces and nodes;
  while (1){
    if(  $C_1$  FIFO is not empty ) Process a  $C_1$  element;
    else if(  $C_2$  FIFO is not empty ) Process a  $C_2$  element;
    else if(  $C_3$  FIFO is not empty ) Process a  $C_3$  element;
    else if( There are layers around voids )
      Prick one of the layer;
    else break;
  }
  Extract non- $C_0$  elements and marked elements;
  Classify the extracted elements by scalar values of their nodes;
}
```

Figure 7.5: 四面体要素で構成されるポリュームの細線化手法の擬似コード

7.5.1 細線化手法のためのデータ構造の整備

本手法では, 5.2 に示されたポリューム・データのデータ構造に, 下記の内容を追加して細線化手法を実行する .

定義 7.6 本章では, 要素 E は, 要素頂点 $\{N_1, N_2, \dots\}$, および要素の面を共有する隣接要素 $\{E_1, E_2, \dots\}$ の他に, 後述する要素の階級 $\{C\}$, およびマークされているか否かを示す符号 $\{M\}$ を保持するものとする . また, E に隣接する非除去要素数を, 要素の階級と定義する . 本章では, 要素の階級を C_n (四面体要素において $n = -1, 0, 1, 2, 3, 4$, 六面体要素において $n = -1, 0, 1, 2, 3, 4, 5, 6$ である) と表示する . また本手法では, 骨格から除去された要素の階級を, 便宜上 C_0 と表示する . さらに, 極点に接してマークされた要素の階級を, 便宜上 C_{-1} と表示する .

定義 7.7 本章では, 境界面 B は, その面をもつ要素 $\{E\}$ と, 要素中の面の番号 $\{i\}$ の他に, 連結な境界面群の番号 $\{n_B\}$ を保持するものとする . 本章では, ポリュームの外部

にある連結な境界面群に1番，ポリウームの空洞に接する連結な境界面群に，それぞれ2番以降の番号をつける．

定義 7.8 本章では，要素頂点 N は，3次元座標値 $\{x, y, z\}$ と，その位置における数値解析結果のスカラー値 $\{S\}$ の他に， N が接する連結な境界面群の番号 $\{n_B\}$ ($n_B = 1, 2, 3, \dots$) を保持するものとする．ただし， N が境界面に接していない場合には， $n_B = 0$ とする．

定義 7.9 本章では，極点 X は，極点に相当する要素頂点 $\{N\}$ ，および X に接する一要素 $\{E\}$ を保持するものとする．

本手法ではまず，6.3.1 節に示された手法で，極点を抽出し，各々の極点に接する要素 E をひとつずつ選択し，その要素の階級を C_{-1} とする．続いて， C_{-1} でない要素に階級をつける．さらに，境界面に接し，かつ隣接要素をもつ要素 C_n (四面体要素において $n = 1, 2, 3$ ，六面体要素において $n = 1, 2, 3, 4, 5$) に対する FIFO キューを確保し，対応する要素を FIFO キューに登録する．

細線化手法では，FIFO に登録された，非除去要素群の境界面に接する要素を抽出し，その必要性を判定する．もし当該要素が不必要であると判定されたら，その階級を C_0 に変更する．同時に， C_0 とされた当該要素に隣接する要素のうち，その階級が C_0 でも C_{-1} でもないものに対して，その階級を C_n から C_{n-1} に修正する．階級を修正された隣接要素は， C_n の FIFO から消去され，かわりに C_{n-1} の FIFO に登録される．

本手法では同時に，連結な境界面群を構成する境界面に，境界面群ごとに固有な番号をつける．続いて，境界面がもつ要素頂点に対し，その境界面の番号をつける．細線化手法によって要素 E が階級を C_0 に修正された時に，境界面群の番号 n_B が 0 である要素頂点には， E がもつ他の要素頂点につけられた境界面群の番号 n_B を代入する．

7.5.2 細線化手法による要素の除去

前節で示したデータ構造の整備の後に，非除去要素群の境界面に接する要素が FIFO から抽出され，細線化手法が開始される．

本手法では，まずすべての C_1 要素が抽出される． C_1 要素の FIFO からすべて要素が抽出されたときに，はじめて C_2 要素が抽出される．もし C_2 要素の処理によって，その隣接要素の階級が C_1 に修正されたときには，他の C_2 要素よりも先に C_1 要素を抽出す

る．同様にして， C_n 要素までの要素がすべて抽出されたときに，はじめて C_{n+1} 要素が抽出される．もし C_{n+1} 要素の処理によって，その隣接要素の階級が C_n 以下に修正されたときには，他の C_{n+1} 要素よりも先に C_n 以下の要素を抽出する．

本手法で抽出された要素のうち， C_1 要素は三共有辺や四共有頂点を持たないので，すべて除去される． C_2 以上の階級にある要素は，下記に示す判定条件で不必要と判定された要素だけが C_0 に階級を修正され，骨格から除去される．

四面体要素の判定条件

四面体要素で構成されるボリュームでは，FIFO から抽出された要素は下記の判定条件にしたがって処理される．

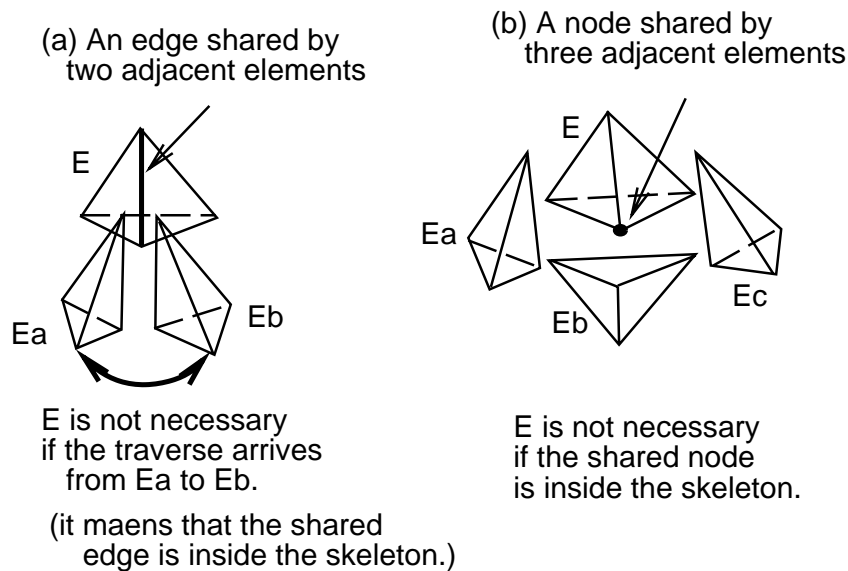


Figure 7.6: 四面体要素の判定条件

C_1 要素: FIFO から抽出されたすべての C_1 である要素 E は，階級を C_0 に修正される．

C_2 要素: C_2 である要素 E の非除外隣接要素を E_a および E_b と表す． E_a を出発点として，三共有辺に接する E_a の隣接非除去要素を順次探索し，探索が E_b に到達す

るとき，三共有辺は非除去要素要素群の境界面上にない．この時， E の階級は C_0 に修正される (図 7.6 (a) 参照) ．

C_3 要素: C_3 である要素 E の非除外隣接要素を E_a , E_b , および E_c と表す． E の四共有頂点の n_B の値が 0 でない場合には，四共有頂点は非除去要素群の境界面上にあるので， E は除去されない． n_B の値が 0 である場合には， E の階級は C_0 に修正される (図 7.6 (b) 参照) ．

六面体要素の判定条件

六面体要素で構成されるボリュームでは，FIFO から抽出された要素は下記の判定条件にしたがって処理される．

C_1 要素: FIFO から抽出されたすべての C_1 である要素 E は，階級を C_0 に修正される．

C_2 要素: C_2 である要素 E の非除外隣接要素を E_a および E_b と表す．この時， E には三共有辺が存在する場合と存在しない場合がある．三共有辺が存在する場合には，四面体要素の判定と同様に， E_a を出発点として，三共有辺に接する隣接非除去要素を順次探索する．探索が E_b に到達するとき，三共有辺は非除去要素要素群の境界面上にない．この時， E の階級は C_0 に修正される．一方，三共有辺が存在しない場合には， E の除去によって， E_a と E_b 要素の連結性は保持されなくなる．よってこの場合， E は除外されない (図 7.13 (a) 参照) ．

C_3 要素: C_3 である要素 E の非除外隣接要素を E_a , E_b , および E_c と表す．この時， E には四共有頂点が 1 個存在するか，三共有辺が 2 個存在するかのいずれかである．前者の場合には，四共有頂点の n_B の値が 0 である場合には，四共有頂点は非除外要素群の境界面上にないので， E の階級が C_0 に修正される．後者の場合に，三共有辺の 1 本が E と E_a と E_b に共有され，もう 1 本が E と E_b と E_c に共有されているとする．この時， C_2 である要素の判定と同様に， E_a から E_b にむかって隣接非除去要素を探索し，さらに E_b から E_c にむかって同様に探索する．探索が両方とも成功する場合には，三共有辺は両方とも非除外要素群の境界面上にないので， E の階級は C_0 に修正される．

C_4 要素: C_4 である要素 E には, E には四共有頂点が2個存在するか, 三共有辺が4個存在するかのいずれかである. 前者の場合には, 2個の四共有頂点の n_B の値が両方とも0である場合には, 四共有頂点は2個とも非除外要素群の境界面上にないので, E の階級は C_0 に修正される. 後者の場合には, E の除外によって, 骨格に不必要なループが生じる可能性があるので, E は除去されない.

C_5 要素: C_5 である要素 E には, E には四共有頂点が4個存在する. 4個の四共有頂点の n_B の値がすべて0である場合には, 四共有頂点は4個とも非除外要素群の境界面上にないので, E の階級は C_0 に修正される.

7.5.3 空洞周辺の非除去要素層の除去

本手法では, FIFO から非除去要素が抽出されなくなった時点で, 階級が C_0 でない要素, およびマークされた要素を, 骨格を構成する要素とみなす. 非構造ボリュームでは, 各々の貫通穴の周辺には, 骨格のループが生成されている. そのため, 骨格の種数は貫通穴の個数に等しくなる. また, 空洞の周辺には, 骨格中に要素の層が生成され, 空洞の境界面の非連結性が保持されている.

ここで, 定理 7.3 に示されるように, 貫通穴を持たないボリュームでは, 極点グラフと等値面との交差が保証されている. つまり, 空洞周辺の要素の層は, 等値面との交差要素の抽出には必ずしも必要ではない.

そこで本手法では, FIFO からすべての非除去要素が抽出された後に, 空洞周辺の要素の層を骨格から除去する. 本手法では, 下記のようなアルゴリズムによって, 要素の層を除去する (図 7.7 参照).

1. 層を構成する要素 (ただしマークされていない要素に限る) をいくつか除去し, 骨格の外側の連結な境界面群と, 層の内側にある連結な境界面群を, 連結させる.
2. 除去された要素の隣接非除去要素について, その階級を C_n から C_{n-1} に修正し, FIFO に登録する.
3. FIFO に登録された要素を抽出して細線化手法を再開し, 層を構成する要素の多くが除去される.

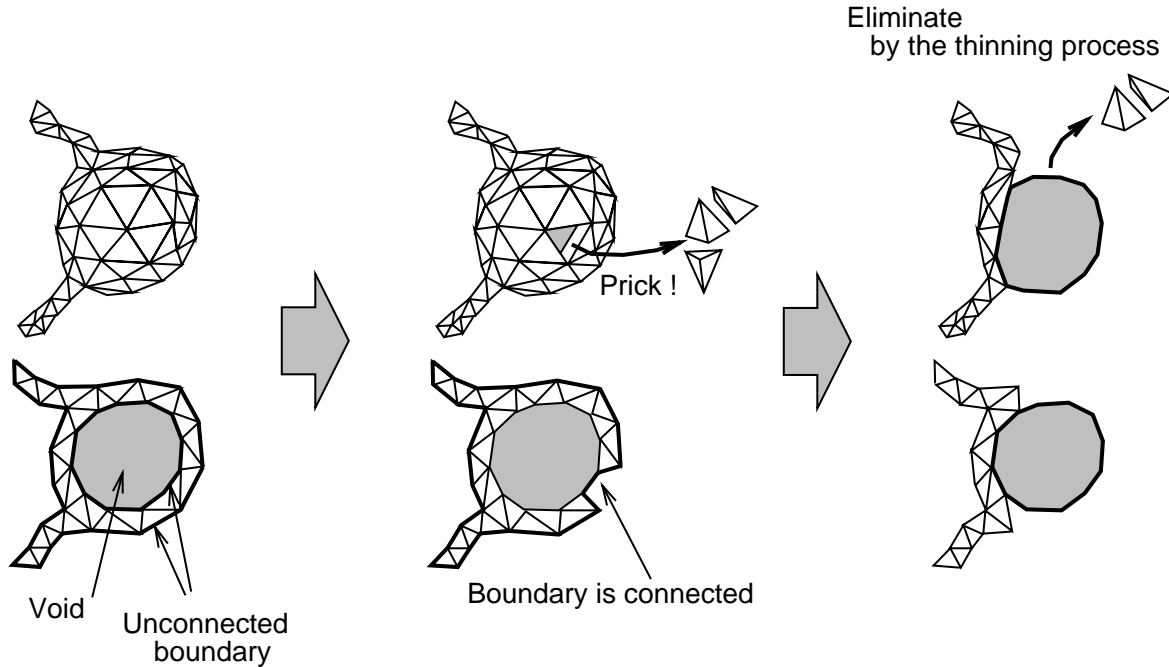


Figure 7.7: 空洞周囲の要素の層の除去

以下，上記のアルゴリズムを，四面体要素で構成されるボリュームについて考察する．本手法ではまず，非除去要素 E の4つの要素頂点 N_1, N_2, N_3, N_4 の境界面群の番号 n_B を参照し，4要素頂点のうち2頂点 N_1, N_2 において $n_B = 1$ であり，残りの2頂点 N_3, N_4 において $n_B > 1$ である要素を抽出する．この要素 E は，辺 N_1N_2 が骨格の外側の境界面上にあり，辺 N_3N_4 が層の内側の境界面上にある．続いて， E の隣接非除去要素のうち，3要素頂点において $n_B = 1$ である要素 E_1 と，3要素頂点において $n_B > 1$ である要素 E_2 を選択する．この時， E_1 は骨格の外側の境界面に接し， E_2 は層の内側の境界面に接している．よって， E, E_1, E_2 の3要素を除去することで，それまで非連結だった骨格の外側の境界面と層の内側の境界面は連結される．続いて， E, E_1, E_2 の隣接非除去要素を FIFO に登録することで，細線化を再開することができる．

上記の処理は，すべての FIFO から要素の抽出が終わったときに実行され，それぞれの空洞ごとに1回ずつ実行される．また，六面体要素で構成されるボリュームについても，同様な手法によって要素の層を除去することが出来る．

7.5.4 非除去要素の登録

本手法では、前節までに示した処理を終了した後に、階級が C_0 でない要素、およびマークされた要素を抽出してリストに登録する。等値面はリストに登録された最低1個の要素と交差するので、このリストに登録された要素との交差判定を実行し、交差要素を出発点として自己増殖的に等値面を生成することが出来る。

本論文では、交差要素の抽出の高速化のために、文献 [17] に示される手法と同様な手法を用いて、非除去要素を要素頂点のスカラ値の大小で階級化し、階級ごとにわけてリストに登録する。等値面のスカラ値が指定されると、そのスカラ値に相当する階級に登録された要素のみに対して、等値面との交差判定が実行される。

7.5.5 自己増殖的手法を用いた等値面生成

本論文では、前節までに示された、極点抽出・細線化処理の各処理を、「前処理」と位置づける。この前処理は、1個のボリュームデータに対して1回だけ実行される。本節で説明する「本処理」は、等値面のスカラ値が指定されるたびに、前処理にて生成されたボリュームの骨格を用いて実行される。

本手法による等値面生成のアルゴリズムを、下記に示す。

1. いくつかの要素につけられている「処理済」マークを消す。
2. 等値面のスカラ値を決定する。
3. 等値面のスカラ値に該当する、骨格中の要素を保持している階級を指定する。
4. 指定された階級に登録されている要素について、以下の処理を実行する。
 - (a) 登録されている要素のうち、「処理済」マークのつけられていない要素に対して、等値面との交差判定を実行し、「処理済」マークをつける。
 - (b) 等値面との交差要素が検出されたら、その要素を出発点にして、自己増殖的に等値面を生成する。

なお、自己増殖的な等値面生成については、本論文では 5.4.2 節に示されたアルゴリズムを用いている。

7.6 細線化手法を用いた等値面生成手法の処理量

細線化手法を用いた等値面生成手法の処理量を，前処理と本処理にわけて考察する．ここで，要素の総数を n とし，要素頂点の総数は要素の総数に比例するものとする．

前処理は，極点抽出部・細線化処理部の各部分に分けることが出来る．極点抽出部では，すべての要素を 1 回ずつ処理し，その後すべての要素頂点を 1 回ずつ処理するので，処理量は $O(n)$ である．細線化処理部では，データ構造整備のためにすべての要素と要素頂点を 1 回ずつ処理し，マークされた要素などを除くすべての要素に対して 1 回判定処理を行なうので，処理量は $O(n)$ である．以上の評価により，前処理の処理量の総計は $O(n)$ であると評価される．

本処理の処理量は，探索される要素数に比例する．もし骨格が直線に近い要素の線列の集合であれば，仮定 5.1 より，細線化処理によって生成された骨格中の要素数は， $O(n^{1/3})$ である．また，仮定 5.1 より，自己増殖的手法によって処理される要素数は， $O(n^{2/3})$ である．

7.7 実行例

本節では，本手法を実装して実行した結果を示す．なお，計算機は IBM PowerStation RS/6000 (Model 560) を用いた．

図 7.14 および 図 7.15 は，細線化手法によって生成されたポリュームの骨格の例を示したものである．図 7.16 および 図 7.17 は，図 7.14 および 図 7.15 に示される骨格を用いて，等値面を生成した例である．図 7.18，図 7.19，図 7.20，および 図 7.21 は，図 7.15 に示される骨格を生成する細線化処理の過程を示したものである．ここで，図 7.14 に示す骨格の色は，隣接非除去要素数を示している．また，図 7.15 に示す骨格の色，図 7.16 および 図 7.17 に示す等値面の色は，それぞれスカラ値の大小を表している．

ここで，細線化手法によって生成された骨格を用いた等値面生成の処理速度と，要素のソート [21] および階級化 [17] を用いた手法，および極点グラフ法による処理速度を比較する．本章に示す処理速度測定では，有限要素法の数値解析結果によって得られる，四面体要素で構成される 5 種類の非構造ポリュームを用いて，20 枚の異なるスカラ値をもつ等値面の生成に要する処理量を測定した．なお，本節では，6.5 節における処理速度測定で用いた非構造ポリュームと同一ポリュームを用いているが，OS やコンパイラ等の相

違により，その処理速度測定結果は必ずしも一致していない．

表 7.1 は，極点グラフ法および細線化手法によって要素リストに登録される要素数，および各手法の前処理に要する処理時間をまとめたものである．

Table 7.1: 極点グラフ法および細線化手法の登録要素数・および各手法の前処理の処理時間

データ	1	2	3	4	5
N_e	20736	61680	346644	557868	458664
N_n	4002	11624	62107	97473	80468
N_{ep}	21	46	135	540	5986
N_{e1}	2516	6480	20158	28086	121492
N_{e2}	436	1365	3757	10967	55536
T_{sort} (sec.)	1.15	4.12	47.95	52.29	63.60
T_{filt} (sec.)	0.19	0.52	2.75	4.76	3.66
T_{egra} (sec.)	0.58	1.53	7.46	14.51	76.78
T_{thin} (sec.)	1.13	3.22	16.93	26.86	23.22

ここで，表中のそれぞれの変数は，

N_e : 要素の総数

N_n : 要素頂点の総数

N_{ep} : 極点の総数

N_{e1} : 極点グラフ法で，極点グラフおよび境界面要素リストに登録された要素の総数

N_{e2} : 細線化手法で，ボリュームの骨格として登録された要素の総数

T_{sort} : 要素をソートする手法の処理時間

T_{filt} : 要素を階級化する手法の処理時間

T_{egra} : 極点グラフ法の処理時間

T_{thin} : 細線化手法の処理時間

を示すものである。

表 7.1 は、極点グラフ法、および細線化手法における前処理の処理時間は、要素をソートする手法よりも小さく、要素を階級化する手法よりも大きいことが示されている。

N_{e1} と N_{e2} を比較してわかる通り、細線化手法によって登録される要素数は、極点グラフ法によって登録される要素数よりも大幅に少ない。このことから、細線化手法では極点グラフ法と比較して、等値面との交差要素の抽出に要する処理量を低減することができると思われる。

また、極点グラフ法における前処理の処理時間 T_{egra} は、要素数に比例していないが、細線化手法における前処理の処理時間 T_{thin} は要素数にほぼ比例している。特に、表 7.1 のデータ no.5 は、データ no.4 よりも要素数が少ないにもかかわらず、前処理に要する処理時間が大きくなっている。この結果は、第 6 章で示した通り、極点数の増加にともなうところが大きい。この結果から、細線化手法は、極点グラフ法における前処理量の不安定さという問題点を解決する手法であることが示されている。

続いて、細線化手法によって生成されたボリュームの骨格の形状について考察する。表 7.2 は、細線化手法によって登録された要素の階級をまとめたものである。

Table 7.2: 細線化手法によって登録された要素の階級別要素数

データ	1	2	3	4	5
C_1	21	45	124	491	3208
C_2	397	1259	3453	9552	20574
C_3	17	61	178	875	15312
C_4	1	0	2	49	16436

表 7.1 と表 7.2 と比較してわかる通り、階級が C_1 である要素数は、極点数にほぼ等しい。このことから、ボリュームの骨格を構成する要素の線列は、そのほとんどが極点に接する要素を端点としていることが推測される。また、データ no.5 では、極点数より圧倒的に多い C_3 および C_4 の階級の要素が残存している。この要素数を低減することができるかどうかを見きわめることは、本研究の今後の課題につながるものと思われる。

続いて、等値面生成処理の処理時間を比較する。表 7.3 は、各手法を用いて 20 枚の等値面を生成した総処理量を示したものである。

ここで、表中のそれぞれの変数は、

Table 7.3: 等値面生成の処理量

データ	1	2	3	4	5
N_t	67875	80995	135358	1164616	494480
N_v	34921	43158	71358	588796	251506
T_{sort} (sec.)	5.69	6.85	15.24	100.92	53.48
T_{filt} (sec.)	5.78	8.36	25.51	107.89	59.31
T_{egra} (sec.)	3.31	4.18	7.56	57.72	26.49
T_{thin} (sec.)	3.27	3.96	7.22	57.12	25.21

N_t : 等値面の三角形面素の総数

N_v : 等値面の三角形面素の頂点の総数

T_{sort} : 要素をソートする手法の処理時間

T_{filt} : 要素を階級化する手法の処理時間

T_{egra} : 極点グラフ法の処理時間

T_{thin} : 細線化手法の処理時間

を示すものである。

表 7.3 からわかる通り，細線化手法における等値面生成に要する処理時間は，他の手法と比べて小さい．その最も大きな理由は，細線化手法における，等値面との交差要素の抽出に要する処理量が，他の手法と比べて非常に小さいことである．表 7.4 は，各手法において，等値面との交差要素の抽出に要する処理時間をまとめたものである．

Table 7.4: 等値面との交差要素の抽出の処理時間

データ	1	2	3	4	5
T_{sort} (sec.)	2.24	2.86	8.65	42.65	26.86
T_{filt} (sec.)	3.13	5.95	22.59	65.08	40.20
T_{egra} (sec.)	0.71	0.98	2.07	12.14	6.35
T_{thin} (sec.)	0.65	0.74	1.76	11.58	5.27

ここで、表中のそれぞれの変数は、

T_{sort} : 要素をソートする手法において、等値面との交差要素の抽出に要する処理時間

T_{filt} : 要素を階級化する手法において、等値面との交差要素の抽出に要する処理時間

T_{egra} : 極点グラフ法において、等値面との交差要素の抽出、および隣接交差要素の FIFO への登録に要する処理時間

T_{thin} : 極点グラフ法において、等値面との交差要素の抽出、および隣接交差要素の FIFO への登録に要する処理時間

を示すものである。

表 7.5 は、各手法において、等値面との交差判定を実行した要素数をまとめたものである。この表から、細線化手法では、等値面との交差判定を実行する要素数が他の手法よりも低減されており、その結果等値面との交差要素の抽出に要する処理量が低減されている。

Table 7.5: 等値面との交差判定を実行した要素数

データ	1	2	3	4	5
N_{sort}	51348	61766	101951	879735	378712
N_{filt}	89548	179088	786220	1927026	1278742
N_{egra}	54420	66618	115639	907319	550605
N_{thin}	51762	63089	105762	890134	436658

ここで、表中のそれぞれの変数は、

N_{sort} : 要素をソートする手法において、等値面との交差判定を実行した要素数

N_{filt} : 要素を階級化する手法において、等値面との交差判定を実行した要素数

N_{egra} : 極点グラフ法において、等値面との交差判定を実行した要素数

N_{thin} : 細線化手法において、等値面との交差判定を実行した要素数

を示すものである。

7.8 ボリューム・レンダリングへの応用

本節では，細線化手法を用いた等値面生成手法を，ダイレクト・アプローチの代表的な手法であるボリューム・レンダリングの高速化に応用する手法を提案する．本節ではまず，ボリューム・レンダリングに関する，従来の研究成果を概観する．続いて，視点からの等距離面を重ね描きする高速なボリューム・レンダリング手法について示し，等距離面の生成に細線化手法を適用した高速化手法を示す．

7.8.1 レイ・キャスティングを用いたボリューム・レンダリング

レイ・キャスティング法 [18] [38] は，レイ・トレーシング法に類似した手法でボリューム・データを表現する手法で，最も代表的なボリューム・レンダリング手法である．レイ・キャスティング法では，図 7.8(a) に示す通り，従来のレイ・トレーシング法における物体表面との交点計算をしないかわりに，ボリューム・データ中でレイをまっすぐ通過させ，その過程に設定した複数のサンプリング点における輝度計算結果を重ね描きする．また，レイ・キャスティング法では，従来のレイ・トレーシング法における鏡面反射方向や透過屈折方向へのレイの発生は実行しない．

レイ・キャスティング法では，それぞれの画素に対してレイを発生し，まずボリューム・データの境界面との交点計算を実行する．続いて，レイと交点をもつ境界面に接する要素を出発して，レイを通過する隣接要素を順に探索し，その過程で等間隔でレイ上にサンプリング点を設定する．要素の探索が終了したら， m 個のサンプリング点に対し，以下の式を用いて，視点から遠いサンプリング点から順に輝度値を加算する．

$$C_t = C_{t-1}(1 - a(t)) + c(t)a(t) \quad (7.1)$$

ただし，

C_t : 輝度値の t 番目のサンプリング点までの累算値 ($C_0 = 0.0$)

$c(t)$: t 番目のサンプリング点における輝度値 (RGB 独立)

$a(t)$: t 番目のサンプリング点における不透明率

である．

5.2.3 節で述べた通り，ボリューム・データのダイレクト・アプローチでは，従来のレンダリング技術に用いられる物理モデルと異なる，独自の物理モデルを構築する必要がある．ボリューム・レンダリングでは，不均一に分布する無限小の微粒子から構成される半透明のゲルを，ボリューム・データ中の物理モデルとして仮想する．このモデルにおいて，輝度値 $c(t)$ は， t 番目のサンプリング点 P_t において微粒子が視線方向に発する光エネルギー量に相当する．また，不透明率 $a(t)$ は， P_t における微粒子の密度に相当する．このモデルは，本来は大気中に浮遊する不定形状の写実的な表現，具体的には雲や炎などのグラフィックス表示手法に用いられたものである [5] [33] ．

通常のレンダリングでは，物体表面における質感を表現するためには，輝度値の算出式に法線方向を用いることが不可欠である．ボリューム・レンダリングにおいても，画像の奥行き感を表現するためには，通常のレンダリングの輝度値の算出式を導入することが不可欠である．そこで，ボリューム・レンダリングでは，微粒子の集合に法線方向を仮想する．法線方向は，ボリューム・データ中のスカラ値の勾配方向が用いられる．

ボリューム・レンダリングを科学技術計算結果のグラフィックス表示に用いる際には，不透明率 $a(t)$ を P_t におけるスカラ値 $S(P_t)$ の関数で表す．また，輝度値 $c(t)$ を， P_t における光線の反射率 $r(t)$ およびスカラ値の勾配方向 $G(t)$ の関数で表し，光線の反射率 $r(t)$ を P_t におけるスカラ値 $S(P_t)$ の関数で表す．

$$c(t) = E + (G(t) \cdot L)lr(t) \quad (7.2)$$

ただし，

L: 光線の方向ベクタ

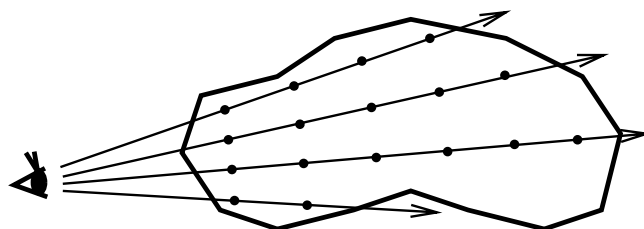
l: 光線のエネルギー

を示すものである．

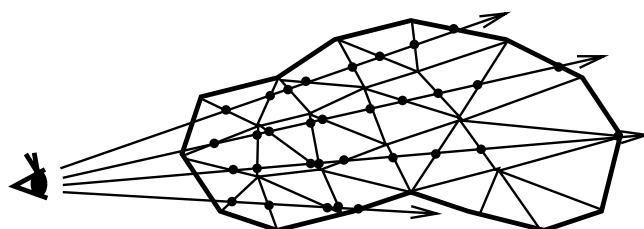
式 7.2 を用いて輝度計算を実行するためには， P_t におけるスカラ値 $S(P_t)$ ，およびその勾配方向 $G(P_t)$ が算出される必要がある．定義 5.5 に示す通り，スカラ値 $S(P_t)$ は P_t を内部に含む要素 E の要素頂点におけるスカラ値を線形補間して算出する．ボリューム・レンダリングでは，この定義をスカラ値の勾配方向の算出にも拡張し， E の要素頂点における勾配方向を線形補間して算出することが多い．そのため，ボリューム・レンダリングを実行する準備処理では，要素頂点にはスカラ値を与えるだけでなく，スカラ値の勾配方向ベクトルを算出する必要がある．

7.8.2 ポリゴンの半透明表示手法を用いた高速ボリューム・レンダリング

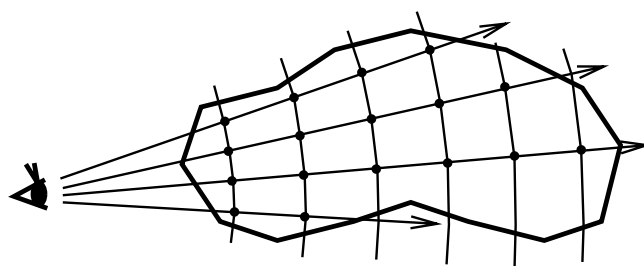
ボリューム・レンダリングの高速化手法には，レイ・キャスト法を高速化する手法と，ポリゴン表示を用いた高速化手法がある．前者には，周波数空間への変換によって処理量を $O(n^{2/3} \log n)$ に低減する手法 [61]，要素を分類してサンプリング点を低減する手法 [26] [39] などがある．本節では，汎用的なグラフィックスハードウェアやソフトウェアとの併用により高速化の効果が非常に高くなる，ポリゴン表示手法について概観する．



(a) Ray Casting



(b) Splatting



(c) Equi-distant surface rendering

Figure 7.8: ボリューム・レンダリング

ポリゴン表示を用いた最も典型的な高速ボリューム・レンダリング手法は，体積を仮想した要素頂点や，要素の面を重ね描きする手法 [65] [69] である．これらの手法は，要素

内部の任意の点におけるスカラー値や勾配方向の算出が不要であるので，非常に高速に画像を生成することが出来る．単純にデータ順に要素頂点や要素面を重ね描きしたのでは，描画順と視点からの距離順が無関係になるので，要素頂点や要素面を順列する手法 [68] と併用することが望ましい．また，これらの手法は，同一画素におけるサンプリング点の間隔が不均一 (図 7.8(b) 参照) であるので，画像の奥行き感を確実に期待することは出来ない．

レイ・キャスティングとほぼ同等な画像のボリューム・レンダリングをポリゴン表示で実現する手法として，視点からの等距離面をポリゴン近似して視点から遠い順に重ね描きする手法 [35] が報告されている．この手法では，図 7.8(c) に示す通り，サンプリング点がレイ・キャスティング法と同等に設定され，しかもレイ・キャスティング法よりも大幅に高速であることが示されている．一方，文献 [35] に指摘されている通り，この手法では処理時間の多くを等距離面生成に費やされており，その高速化が課題であるとされている．

7.8.3 ボリューム・レンダリングへの応用

前節で述べた通り，半透明な視点からの等距離面を重ね描きするボリューム・レンダリング [35] 手法は，等距離面の生成の処理量が大きいことが問題点であった．ここで，視点からの等距離面は，視点を (V_x, V_y, V_z) とすると

$$(x - V_x)^2 + (y - V_y)^2 + (z - V_z)^2 - C = 0 \quad (7.3)$$

を満たす点 (x, y, z) の集合であるので，等値面の一種であると考えることが出来る．つまり，等距離面の重ね描きによるボリューム・レンダリング手法は，等値面生成の高速化手法を適用することで，さらに高速化を実現することができると考えられる．このボリューム・レンダリング手法を高速化するために，ボリュームの細線化手法を用いた等値面生成手法を，視点からの等距離面の生成に適用する．

図 7.9 (a)(b) は，視点がボリューム・データの内部，および外部にある場合の，視点からの距離値の極点，およびそれを連結する骨格を示したものである．図 7.9 (a) では，ボリューム内部にある視点自身が距離 0 の極小点となる以外は，極点はすべてボリュームの境界面上にある．図 7.9 (b) では，極点はすべてボリュームの境界面上にある．細線化処理によってこの極点の骨格を生成し，骨格を形成する要素と等距離面との交差判定を実行

することで，等距離面との交差要素が抽出され，等距離面が効率的に生成される．本手法

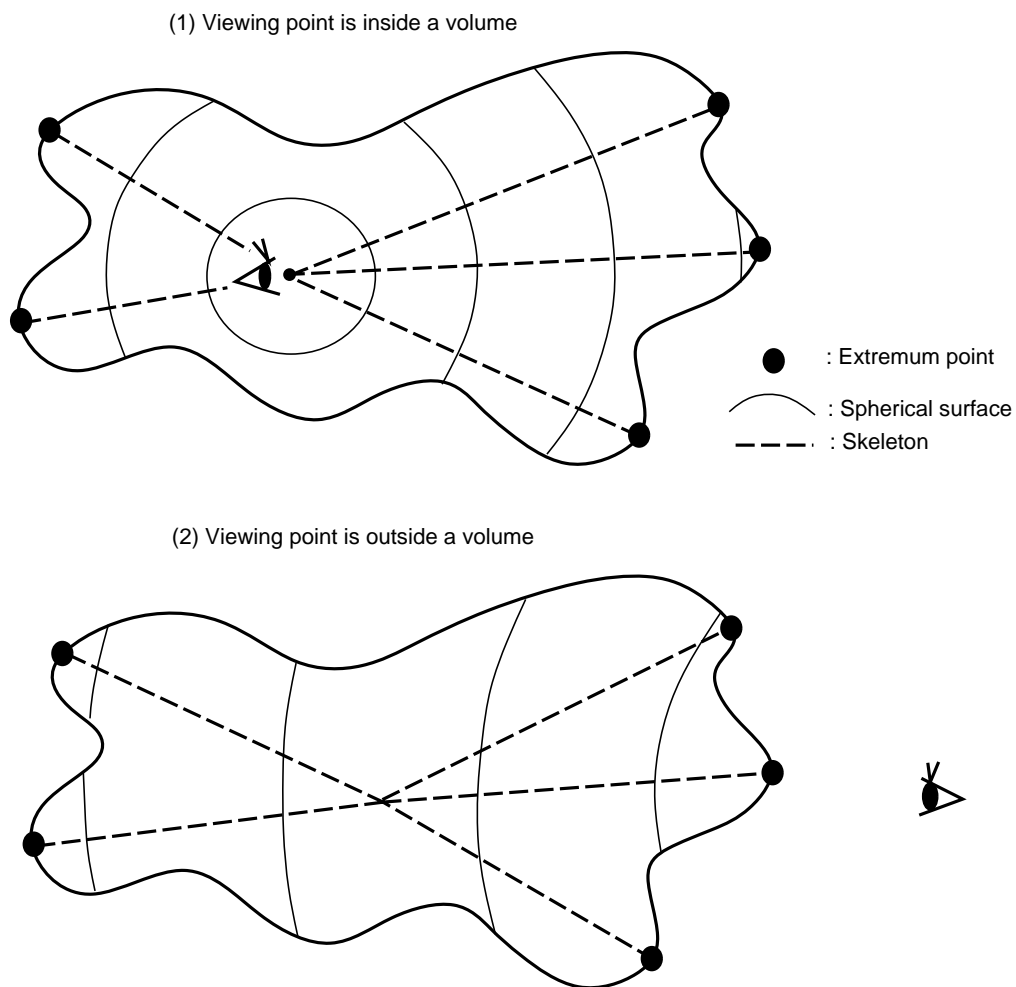


Figure 7.9: 等距離面を用いたポリウム・レンダリングの高速化

を適用したポリウム・レンダリングのアルゴリズムを，下記に示す．

1. 各要素頂点ごとに，視点からの距離を算出する．
2. 6.3.1 節で示した手法を用いて，視点からの距離の極大点および極小点を抽出する．
3. 細線化手法を用いて，極大点および極小点を連結する骨格を生成する．
4. 以下の処理を等距離面の枚数だけ反復する．

- (a) 等距離面の視点からの距離を決定する．
- (b) 骨格を形成する要素に対して，等距離面との交差判定を行ない，交差要素を抽出する．
- (c) 抽出された交差要素を出発点にして，自己増殖的に等値面を生成する．
- (d) 等値面の三角形面素を半透明表示する．

本手法は，特に大規模なボリューム，および等距離面の生成枚数が多い時に，特に効率的に等距離面を生成する手法と考えられる．

ところで，ボリューム・レンダリングによって利用者が望む画像を得るためには，ある程度の試行錯誤を要することが多い．具体的には，

- 視点の移動
- スカラ値から色や半透明度を算出する関数の修正
- 等距離面の生成枚数の変更

などを操作しながら，反復的に画像を生成することが多い．本手法では，視点を変更するたびに，各要素頂点における視点からの距離の算出，極点の抽出，骨格の生成などの処理が必要なので，視点を頻繁に変更するような場合には必ずしも効率的ではない．一方，色や半透明度の算出，等距離面の枚数などは，骨格の形状に影響をおよぼさない．よって，これらの操作が反復されるような場合には，骨格の再生成を必要としないので，本手法は効率的にボリューム・レンダリングを実行できる．

7.9 むすび

本章では，細線化処理を用いてボリュームの骨格を生成し，その骨格を生成する要素の中から等値面と交差する要素を抽出し，自己増殖的に等値面を生成する，高速な等値面生成手法を提案した．

本手法は，極点グラフ法と比較して，以下のような特徴がある．

- ボリュームの骨格を構成する要素数は，極点グラフ法で登録される要素数よりも大幅に小さい．よって本手法で等値面との交差判定を実行される要素数は，極点グラ

フ法より少ないので、極点グラフ法よりも等値面生成の処理時間を短縮することが出来る。

- 細線化処理では、大半の要素を1回だけ処理するので、その処理量は $O(n)$ である。よって本手法における前処理量は、極点グラフ法よりも安定している。

これらの特徴より、本手法は第6章の冒頭に示した、要件6.2、要件6.1、要件6.3のすべての条件を満たす手法であることが示された。

また、本章では、本手法が視点からの等距離面を重ね描きするボリューム・レンダリング[35]の高速化にも用いることが出来ることを示した。本手法を適用することで、視点変更が頻繁である場合を除き、非常に効率的にボリューム・レンダリングを連続実行することが出来る。

付録

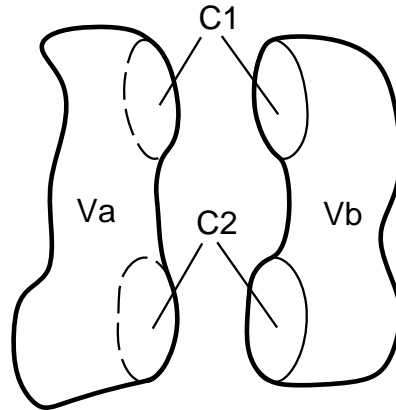
(定理7.1の証明)

中間値の定理により、有限な領域に分布する連続な関数は、その数値が定数でない限り、最大値および最小値を持つ。よって、各領域には、最大値および最小値が存在する。ただし、最大値または最小値の一方が等値面のスカラ値に一致する場合には、その値はボリューム・データ中の極大値や極小値でない場合もある。よって、各領域には、極大点または極小点の両方の存在は保証されないが、少なくとも一方の存在は保証される。

(定理7.2の証明)

貫通穴を持たないボリュームが、非連結な2つの境界つき曲面 C_1 および C_2 を形成する等値面によって、2つの領域 V_a および V_b に分断されるとする(図7.10参照)。この時、 V_a および V_b の境界面のオイラー数 E_a および E_b は、いずれも0となるので、ボリュームの境界面のオイラー数 E_v は、 $E_v = E_a + E_b = 0$ である。この時の種数 $G = 2 - E_v/2 = 1$ であるので、貫通穴を持つことになり、矛盾が生じる。よって、貫通穴を持たないボリュームは、非連結な2つの境界つき曲面を形成する等値面によって、2つの領域に分断されることはない。

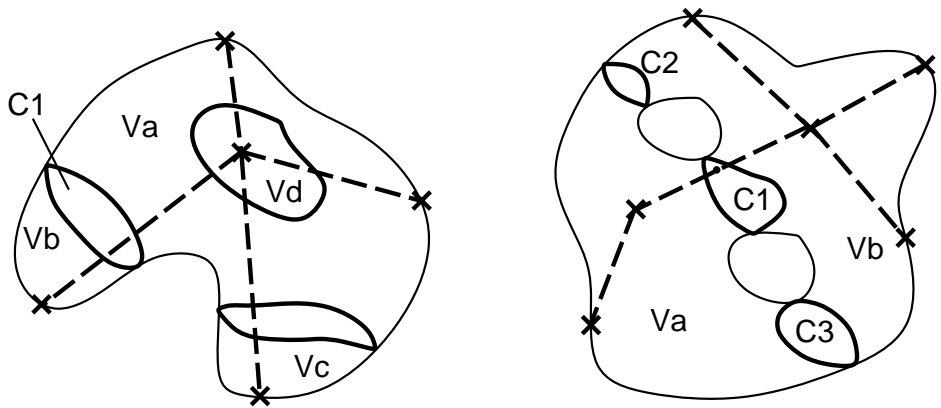
空洞をもつボリュームや、3つ以上の境界つき曲面を形成する等値面についても、同様に証明できる。よって、ボリュームが貫通穴を持たない時に、分断された領域のうちの2



(a) Two regions of a volume divided by an isosurface

Figure 7.10: ボリュームと等値面のトポロジーに関する定理 (1)

領域 V_a および V_b が、等値面の非連結な複数の境界つき曲面を介して接することはない。



(a) No through-hole:
All parts of an isosurface
intersect the extrema graph.

(b) Two through-holes:
All parts of isosurface
do not always intersect.

Figure 7.11: ボリュームと等値面のトポロジーに関する定理 (2)

(定理 7.3 の証明)

定理 7.1 により、極点は等値面によって分断されるボリュームの各領域に存在する。ま

た，定理 7.2 により，ボリュームが貫通穴を持たない時，等値面によって分断される 2 領域 V_a および V_b は，等値面の 1 曲面 C_1 のみを共有する．もし極点グラフが V_a および V_b に共有される等値面とまったく交差しなしたたら，グラフは V_a の内部と V_b の内部で非連結な 2 つのグラフとなってしまうので，矛盾が生じる (図 7.11 (a) 参照)．よって，貫通穴を持たないボリュームに対して極点グラフを生成すると，等値面の非連結なすべての曲面は，最低 1 本のグラフの辺と交差する．

一方，定理 7.2 により，ボリュームが貫通穴を持つ時，2 領域 V_a および V_b は等値面の複数の曲面 C_1, C_2, \dots を共有することがある．この場合には，極点グラフが V_a および V_b に共有される等値面のうち 1 曲面 C_1 のみと交差するとしても，そのグラフは V_a の内部と V_b の内部で非連結にはならないので，矛盾は生じない (図 7.11 (b) 参照)．

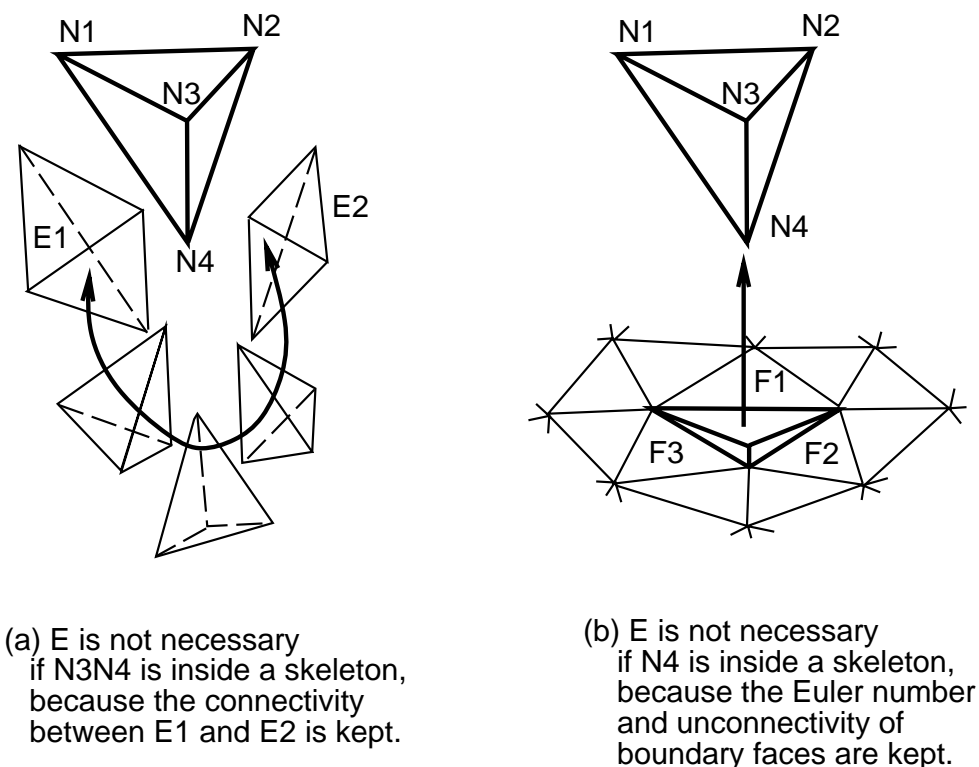


Figure 7.12: ボリュームと等値面のトポロジーに関する定理 (3)

(定理 7.4 の証明)

E は四面体要素であるとし，その要素頂点を $\{N_1, N_2, N_3, N_4\}$ と表す．また， E の 2 面

$\{N_1, N_2, N_3\}$ および $\{N_1, N_2, N_4\}$ が非除去要素群の境界面に接し, 辺 $\{N_3, N_4\}$ は 2 個の非除去要素 E_1, E_2 に共有される三共有頂点であり, $\{N_3, N_4\}$ は非除去要素群の境界面上にないとする. この時, E と E_1 に共有される面, E と E_2 に共有される面に限らず, 辺 $\{N_3, N_4\}$ をもつ n_F 枚の要素面はすべて, 非除去要素群の境界面上にない. よって, E, E_1, E_2 を含む, 辺 $\{N_3, N_4\}$ をもつ n_F 個の要素は, n_F 枚の面を通してループ状に連結している. この時, E_1 と E_2 の連結性は, E を除去しても保持される (図 7.12(a) 参照).

非除去要素群の境界面上にない四共有頂点 N_4 をもつ四面体要素の場合には, 3 辺 $\{N_1, N_4\}, \{N_2, N_4\}, \{N_3, N_4\}$ がともに非除去要素群の境界面上にない三共有辺となる. よって, 同様に E を除去しても隣接非除去要素の連結性を保持することが出来る.

六面体要素の場合にも, 同様にして E を除去した時の隣接非除去要素の連結性を保持することが出来る.

ボリウム・データ中のすべての非除去要素において, 隣接要素どうしの連結性を保持しながら要素を除去するので, マークされた非除去要素の連結性は保持される.

(定理 7.5 の証明)

E は四面体要素であるとし, その要素頂点を $\{N_1, N_2, N_3, N_4\}$ と表す. また, E の 1 面 $\{N_1, N_2, N_3\}$ が非除去要素群の境界面に接し, 要素頂点 N_4 は 3 個の非除去要素に共有される四共有頂点であり, N_4 は非除去要素群の境界面上にないとする. この時, 要素 E を除去することにより, 要素頂点 N_4 が非除去要素群の境界面上に現れる. 面 $\{N_1, N_2, N_3\}$ が非除去要素群から除去されるかわりに, 面 $\{N_2, N_3, N_4\}$, 面 $\{N_3, N_4, N_1\}$, 面 $\{N_4, N_1, N_2\}$ の 3 面が, 非除去要素群の境界面上に現れる. また, 辺 $\{N_1, N_4\}$, 辺 $\{N_2, N_4\}$, 辺 $\{N_3, N_4\}$ の 3 辺が, 非除去要素群の境界面上に現れる. この時, E の除去前の, 非除去要素群の境界面上の要素頂点数, 境界面数, 辺数をそれぞれ n_N, n_F, n_E とすると, E の除去後のオイラー数は,

(オイラー数) $= (n_N + 1) + (n_F - 1 + 3) - (n_E + 3) = n_N + n_F + n_E$ であるので, E の除去前から変化しない (図 7.12(b) 参照). よってこの場合, E を除去してもボリウムの種数は保持される.

四面体要素 E が四共有頂点でなく三共有辺をもつ場合, また E が六面体要素である

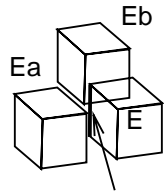
場合にも，同様にして種数が変化しないことが証明される．

(定理 7.6 の証明)

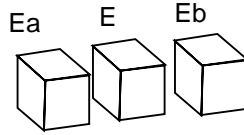
E は四面体要素であるとし，その要素頂点を $\{N_1, N_2, N_3, N_4\}$ と表す．また， E の 1 面 $\{N_1, N_2, N_3\}$ が非除去要素群の境界面に接し，要素頂点 N_4 は 3 個の非除去要素に共有される四共有頂点であり， N_4 は非除去要素群の境界面上にないとする．また，辺 $\{N_1, N_2\}$ ，辺 $\{N_2, N_3\}$ ，辺 $\{N_3, N_1\}$ を面 $\{N_1, N_2, N_3\}$ と共有する非除去要素群の境界面を，それぞれ F_1, F_2, F_3 とする．この時 E の除去により，面 $\{N_1, N_2, N_4\}$ ，面 $\{N_2, N_3, N_4\}$ ，面 $\{N_3, N_1, N_4\}$ の 3 面が，非除去要素群の境界面上に現れ，それぞれ F_1, F_2, F_3 に連結する．よって， E の除去により非除去要素群の境界面上に現れる 3 面はすべて，除去される面 $\{N_1, N_2, N_3\}$ に連結していた面に連結し，他の非連結な面と連結しない (図 7.12(b) 参照)．よって， E を除去しても空洞を囲む境界面と他の境界面との非連結性は保持される．

四面体要素 E が四共有頂点でなく三共有辺をもつ場合，また E が六面体要素である場合にも，同様にして空洞を囲む境界面と他の境界面との非連結性が保持されることが証明される．

(a) C2 element

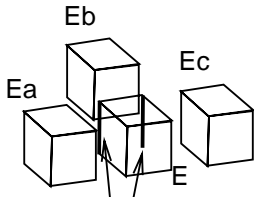


E is not necessary
if the shared edge is
inside the skeleton.

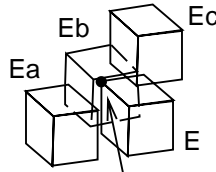


If Ea and Eb do not
have shared edges,
E2 may not be eliminated.

(b) C3 element

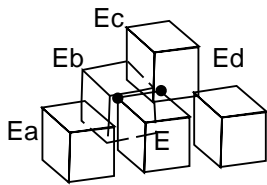


E is not necessary
if both the shared edges
are inside the skeleton.

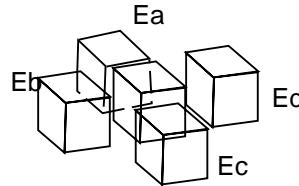


E is not necessary
if the shared node
is inside the skeleton.

(c) C4 element

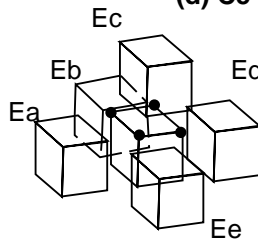


E is not necessary
if both the shared node
are inside the skeleton.



If there are not shared nodes,
E may not be eliminated.

(d) C5 element



E is not necessary
if all four shared nodes
are inside the skeleton.

Figure 7.13: 六面体要素の判定条件

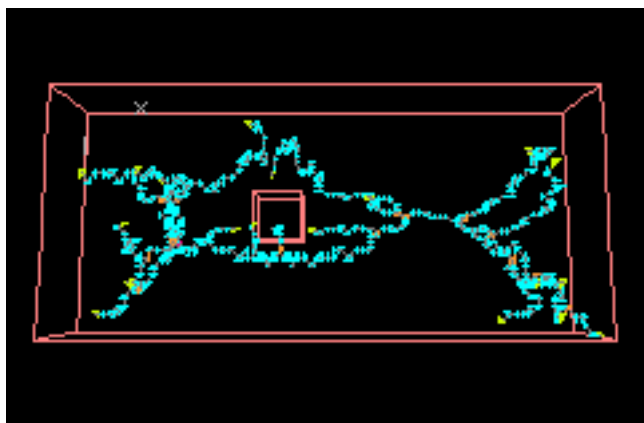


Figure 7.14: 細線化手法による骨格の生成例 (1)

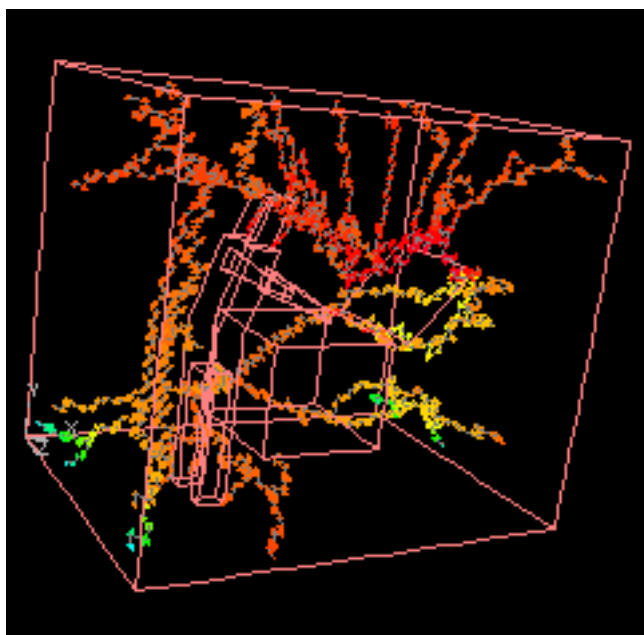


Figure 7.15: 細線化手法による骨格の生成例 (2)

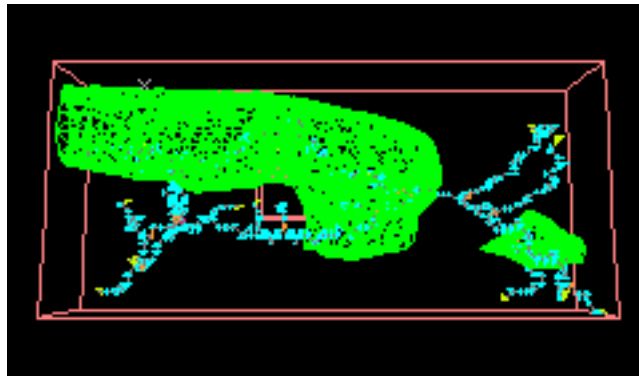


Figure 7.16: 細線化手法を用いた等値面生成例 (1)

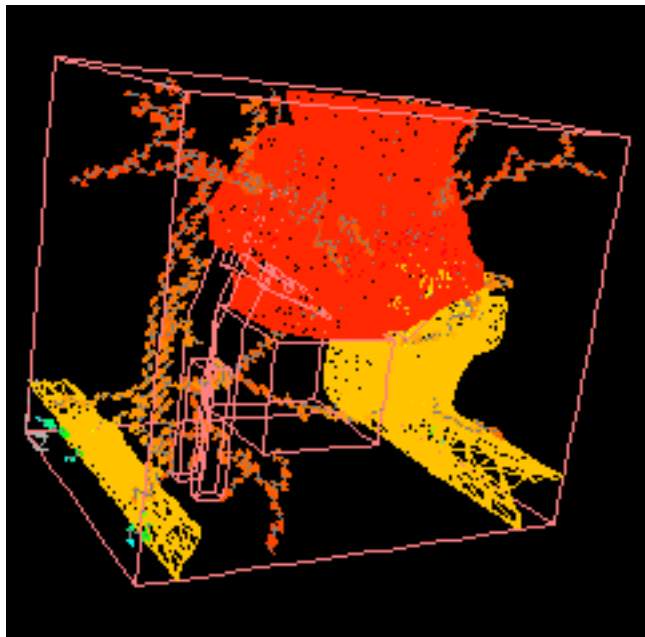


Figure 7.17: 細線化手法を用いた等値面生成例 (2)

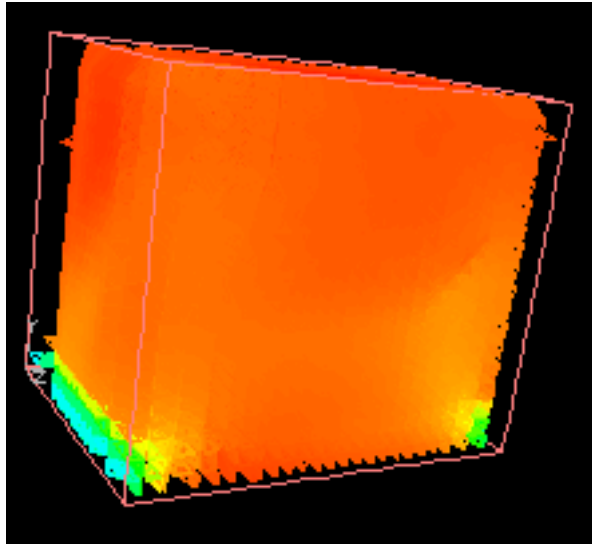


Figure 7.18: 細線化処理の過程 (1)

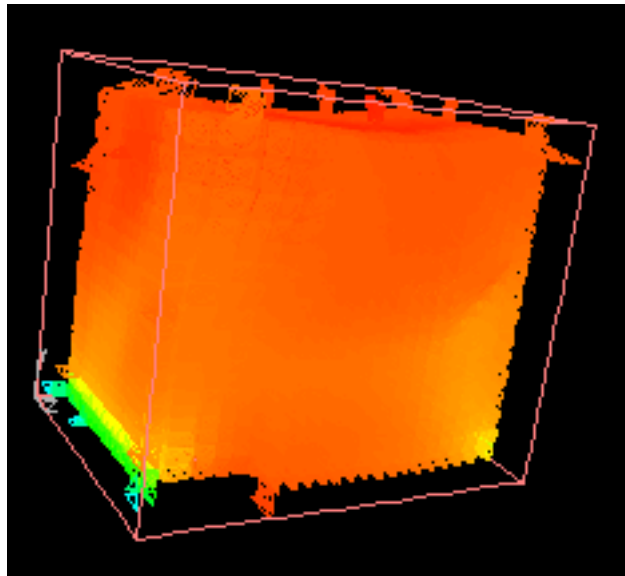


Figure 7.19: 細線化処理の過程 (2)

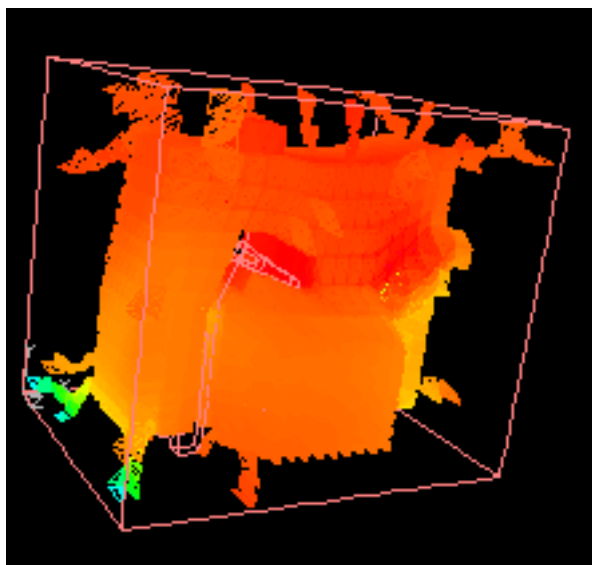


Figure 7.20: 細線化処理の過程 (3)

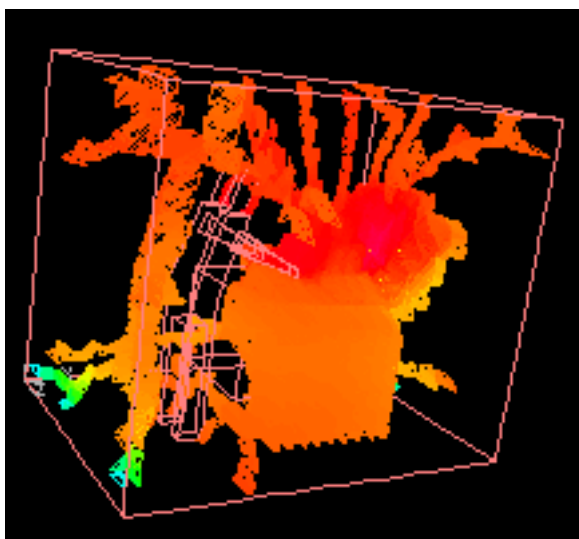


Figure 7.21: 細線化処理の過程 (4)

第 8 章

結論

本章では、本論文の各章で報告した内容を列挙し、その内容を総括する。

第1章では、コンピュータ・グラフィックスの研究開発の現状、ラジオシティ法および等値面生成に関する従来の研究の概観した。続いて、本論文の第3章、4章、6章、7章で提案する内容について、その概要を述べた。さらに、本論文の位置付けと構成を示した。

第2章では、まずラジオシティ法の概要を紹介した。続いて、ラジオシティ法の入力形状の分割技術に適用されている2次元メッシュ生成技術、輝度分布を考慮した適応的メッシュ生成手法について概観した。特に、適応的メッシュ生成手法の代表的な手法である、適応的再分割法と輝度不連続線法について詳しく紹介し、その問題点について示した。

第3章では、簡略な輝度不連続線法を提案した。本手法では、影多面体と受光面との交線を求めることで、照射領域、半影領域、本影領域の境界線のみを輝度不連続線として生成する。本手法は、従来の輝度不連続線と比較して、輝度不連続線の生成数を減少し、またBSP法を用いないので、輝度不連続線の交点計算の計算量を減少すると同時に、極端に小さい要素や細長い要素の生成を低減する。また、輝度変化の大きい半影領域のみにおいて細かく要素を生成し、他の領域において半影領域よりも大きく要素を生成することができるので、要素数の増加を抑えながら写実的な画像を生成することが容易になる。

第4章では、輝度分布参照型バブル・メッシュ法を提案した。本手法では、一樣な要素サイズで生成されたメッシュを用いて1回目のラジオシティ法を実行し、その輝度変化から要素サイズ値を算出し、要素サイズ値を参照しながらメッシュを再生成する。本手法は、1回目のラジオシティ法でメッシュを細かく生成し、輝度変化の小さい部位の要素を大きく再生成することができる。よって、従来の適応的再分割法と比較して、要素数の増加を抑えながら、小さな陰影を欠落することなく表現することが容易になる。また本手法は、なめらかに数値が変化する要素サイズのサンプリング値を用いているので、歪みの小さいメッシュを生成することができる。また、利用者が与えるパラメータは要素サイズ値であり、これは輝度差のしきい値を与える従来の手法のパラメータよりも操作性が高いと思われる。

第5章では、まずボリューム・ビジュアライゼーションとボリューム・データ概念について述べた。続いて、ボリューム・データの代表的なグラフィックス表示手法である等値面生成手法について概観した。さらに、等値面と交差しない3次元要素との処理を省略することによる、等値面生成の高速化手法について報告し、その問題点を列挙した。

第 6 章では、極点グラフ法を提案した。本手法では、ボリューム・データ中の極大点および極小点を連結する極点グラフに登録された 3 次元要素、およびボリュームの境界面に接する 3 次元要素から、等値面と交差する 3 次元要素を抽出し、自己増殖的に等値面を生成する。本手法の等値面生成処理量は $O(n)$ 以下であるので、特に大規模なボリューム・データにおいて、効率的に等値面を生成することが出来る。また、本手法は非構造ボリュームにも適用することができる。

第 7 章では、ボリュームの細線化手法を提案した。本手法では、ボリュームのトポロジを保持しながら、ボリューム中の極点を連結する 3 次元要素の線列を生成する。その線列を生成する 3 次元要素の中から等値面と交差する要素が抽出されるので、抽出された交差要素を出発点として自己増殖的に等値面を生成することができる。本手法では、ボリュームの境界面に接する 3 次元要素との交差判定を必要としないので、極点グラフ法よりもさらに高速に等値面を生成することが出来る。また、本手法における細線化処理では、大半の要素を 1 回ずつ処理するので、その処理量は $O(n)$ で安定している。また、本手法は非構造ボリュームにも適用ができる。

第 3 章、および第 4 章で提案された手法は、いずれもラジオシティ法の算出結果を対話的に連続表示するために、要素数の増加を抑えながら、写實的に陰影等の輝度変化を表現する手法であると位置づけられる。よって、本手法を用いることで、ラジオシティ法を用いて輝度を算出した写實的な幾何データを高速にグラフィックス表示することが容易になる。この研究成果は、ラジオシティ法の主な利用目的である景観シミュレーションにおいて、対話性と写実性の両者をともに向上した点において、大きな意義があるものと思われる。

第 6 章、および第 7 章で提案された手法は、大規模なボリュームにおいても高速に等値面を生成する手法であると位置づけられる。よって、本手法を用いることで、大規模な数値解析結果においても、数値を変化しながら対話的に等値面を連続生成表示することが容易になる。この研究成果は、等値面生成手法の主な利用目的である数値解析結果のグラフィックス表示において、大規模な数値解析結果においても対話的なグラフィックス表示を容易にし、数値解析結果の理解を容易にした点において、大きな意義があるものと思われる。

なお、本論文の成果には直接関連しないが、ラジオシティ法の算出結果、および等値面生成結果の対話的な連続表示をさらに容易にする研究として、下記のようなアプローチ

も考えられる。

第 3 章で提案した簡略な輝度不連続線法は、輝度不連続線の生成数、受光面の分割数を大幅に減少する有用な手法である。しかし、非常に多くの物体が光源の直射光を遮蔽する情景においては、輝度不連続線が大量に発生する可能性がないとはいえない。大量の輝度不連続線の生成を回避する手段として、輝度不連続線が集中する部位において、一度生成された輝度不連続線を消去するような手法を確立することが考えられる。具体的には、本影領域どうしを分断する輝度不連続線、要素サイズよりも十分距離の小さい位置に存在する輝度不連続線の消去が考えられる。

第 3 章で提案した簡略な輝度不連続線法、および第 4 章で提案した輝度参照型バブル・メッシュ法は、原理的には併用が可能な手法である。よって、この 2 手法を併用することで、さらに効果的にメッシュを生成することができるものと思われる。

第 6 章、および第 7 章で提案した等値面生成の高速化手法は、等値面との交差要素の抽出に要する処理量を非常に小さくした有用な手法である。この手法を発展して、さらに高速に等値面を生成するためには、3 次元要素と等値面の交差判定以外の処理を高速化する必要がある。具体的には、面素頂点の検索の高速化や、勾配の少ない領域における面素数の削減などが考えられる。

謝辞

本論文の執筆にあたり，終始懇切なる御指導，御鞭撻を賜わり，常に暖かい御激励を頂きました，早稲田大学理工学部情報学科大石進一教授に，深く感謝の意を表します．

また，本論文の執筆にあたり，多くの懇切な御指導を賜りました，早稲田大学理工学部電子通信学科堀内和夫教授，電子通信学科安田靖彦教授，応用物理学科大頭仁教授，情報学科村岡洋一教授に，深く感謝致します．

また，本研究に関して終始御協力を賜わり，また多くの御指導を賜りました，日本アイ・ビー・エム(株)東京基礎研究所 岩野和生所長，杉本和敏氏，清水和哉氏に，深く感謝致します．

また，本研究の共同研究者として，終始御指導，御協力を頂きました，カーネギー・メロン大学嶋田憲司助教授，東京大学山口泰助教授，岩手大学土井章男氏，日本アイ・ビー・エム(株)東京基礎研究所小山田耕二氏，青野雅樹氏，大淵竜太郎氏，山田敦氏に，深く感謝致します．

また，本論文の執筆に関して終始御指導，御協力を頂きました，中央大学牧野光則助教授，早稲田大学遠藤靖典氏に，深く感謝致します．

また，本論文の執筆に関して終始御協力を頂きました，早稲田大学大石研究室，早稲田大学堀内研究室，中央大学牧野研究室の関係諸氏に，深く感謝致します．

また，本研究に関して終始御協力を頂きました，日本アイ・ビー・エム(株)東京基礎研究所の関係諸氏に，深く感謝致します．

最後に私事ながら，筆者の研究生活を支えて下さった，妻の伊藤みき，および家族に感謝致します．

参考文献

- [1] Aho A. V., Hopcroft J. E., and Ullman J. D., *Data Structure and Algorithms*, Addison Wesley Publishing Company, 1983.
- [2] Arvo J., *The Irradiance Jacobian for Partially Occluded Polyhedral Sources*, Computer Graphics, Vol. 28, No. 3, pp. 343-350, 1994.
- [3] Baum D. R., *Improving Radiosity Solutions through the Use of Analytically Determined Form-factors*, Computer Graphics, Vol. 23, No. 3, pp. 325-334, 1989.
- [4] Baum D. R., Mann S., Kevin P. S., and Winget J. M., *Making Radiosity Usable: Automatic Preprocessing and Meshing Techniques for the Generation of Accurate Radiosity Solutions*, Computer Graphics, Vol. 25, No. 4, pp. 51-60, 1991.
- [5] Blinn J. F., *Light Reflection Functions for Simulation for Clouds and Dusty Surfaces*, Computer Graphics, Vol. 16, No. 3, pp. 21-29, 1982.
- [6] Bloomenthal J., *Polygonization of Implicit Surfaces*, Computer Aided Geometric Design, Vol. 5, No. 4, pp. 341-355, 1988.
- [7] Boissonnat J. D., *Shape Reconstruction from Planar Cross Sections*, Computer Vision, Graphics, and Image Processing, Vol. 41, No. 1, pp. 1-29, 1988.
- [8] Campbell A. T., and Fussell D., *Adaptive Mesh Generation for Global Diffuse Illumination*, Computer Graphics, Vol. 24, No. 4, pp. 155-164, 1990.
- [9] Chin N., and Feiner S., *Near Real-Time Shadow Generation Using BSP Trees*, Computer Graphics, Vol. 23, No. 3, pp. 99-106, 1989.

- [10] Christiansen H. N., and Sederberg T. W., *Convension of Complex Contour Line Definitions into Polygonal Element Meshes*, Computer Graphics, Vol. 12, No. 3, pp. 187-192, 1978.
- [11] Cohen M. F., and Greenberg D. P., *The Hemi-cube: A Radiosity Solution for Complex Environments*, Computer Graphics, Vol. 19, No. 3, pp. 31-41, 1985.
- [12] Cohen M. F., Greenberg D. P., Immel D. S., and Brock P. J., *An Efficient Radiosity Approach for Realistic Image Synthesis*, IEEE Computer Graphics and Applications, Vol. 6, No. 3, pp. 26-35, 1986.
- [13] Cohen M. F., Chen S. E., Wallace J. R., and Greenberg D. P., *A Progressive Refinement Approach for Realistic Image Synthesis*, Computer Graphics, Vol. 22, No. 4, pp. 75-84, 1988.
- [14] Doi A., and Koide A., *An Efficient Method of Triangulating Equi-valued Surfaces by Using Tetrahedral Cells*, IEICE Transactions, Vol. E74, No. 1, pp. 214-224, 1991.
- [15] Doi A., *Bidirectional Radiosity Approach for Complex Environments*, IEEE Tokyo Section, Denshi Tokyo No. 32, pp. 93-99, 1993.
- [16] Gallagher R. S., and Nagtegaal J. C., *An Efficient 3-D Visualizatin Technique for Finite Element Models and Other Coarse Volumes*, Computer Graphics, Vol. 23, No. 3, pp. 185-194, 1989.
- [17] Gallagher R. S., *Span Filtering: An Optimization Scheme for Volume Visualization of Large Finite Element Models*, Proceedings of IEEE Visualization '91, pp. 68-74, 1991.
- [18] Garrity M. P., *Raytracing Irregular Volume Data*, Computer Graphics, Vol. 24, No. 5, pp. 35-40, 1990.
- [19] Gelder A. V., and Wilhelms J., *Topological Considerations in Isosurface Generation*, ACM Transactions on Graphics, Vol. 13, No. 4, pp. 337-375, 1994.

- [20] George P., *Automatic Mesh Generation*, Wiley, 1991.
- [21] Giles M., and Haimes R., *Advanced Interactive Visualization for CFD*, Computer Systems in Engineering, Vol. 1, No. 1, pp. 51-62, 1990.
- [22] Goral C. M., Torrance K. E., Greenberg D. P., and Battaile B., *Modeling the Interaction of Light Between Diffuse Surfaces*, Computer Graphics, Vol. 18, No. 3, pp. 213-222, 1984.
- [23] Gouraud H., *Continuous Shading of Curved Surfaces*, IEEE Transactions of Computers, Vol. C-20, No. 6, pp. 623-629, 1971.
- [24] Heighway E. A., *A Mesh Generator for Automatically Subdividing Irregular Polygons into Quadrilaterals*, IEEE Transactions on Magnetics, Vol. MAG-19, pp. 2535-2538, 1983.
- [25] Howie C. T., and Blake E. H., *The Mesh Propagation Algorithm for Isosurface Construction*, Computer Graphics Forum(Eurographics), Vol. 13, No. 3, pp. C-65-74, 1994.
- [26] 石崎, 牧野, 格子密度による可変歩幅ボリューム・レイ・キャストイング, 情報処理学会グラフィクスとCAD 研究報告, Vol. 94-CG-71, pp. 13-18, 1994.
- [27] 伊藤, 嶋田, 土井, ラジオシティ法のための輝度不連続線およびメッシュの生成に関する効率的な一手法, 情報処理学会論文誌, Vol. 36, No. 11, pp. 2632-2641, 1995.
- [28] 伊藤, 山田, 嶋田, 青野, 大淵, 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, 情報処理学会グラフィクスとCAD 研究報告, 96-CG-81, pp. 73-78, 1996.
- [29] Itoh T., and Koyamada K., *Automatic Isosurface Propagation by Using an Extrema Graph and Sorted Boundary Cell Lists*, IEEE Transactions on Visualization and Computer Graphics, Vol. 1, No. 4, pp. 319-327, 1995. Itoh T., Yamaguchi Y., and Koyamada K.,

- [30] Itoh T., Yamaguchi Y., and Koyamada K., *Volume Thinning for Automatic Isosurface Propagation*, Proceedings of IEEE Visualization '96, pp. 303-310, 1996.
- [31] Joe B., and Simpson R. B., *Triangular Meshes for Regions of Complicated Shape*, International Journal for Numerical Methods in Engineering, Vol. 23, pp. 751-778, 1986.
- [32] Johnston B. P., Sullivan J. M. Jr., and Kwasnik A., *Automatic Conversion of Triangular Finite Element Meshes to Quadrilateral Elements*, International Journal for Numerical Methods in Engineering, Vol. 31, pp. 67-84, 1991.
- [33] Kajiya J. T., and Herzen B. P., *Ray Tracing Volume Densities*, Computer Graphics, Vol. 18, No. 3, pp. 165-174, 1984.
- [34] Koyamada K., *Visualization of Simulated Airflow in a Clean Room*, Proceedings of IEEE Visualization '92, pp. 156-163.
- [35] Koyamada K., Uno S., Doi A., and Miyazawa T., *Fast Volume Rendering by Polygonal Approximation*, Journal of Information Processing, Vol. 15, pp. 535-544, 1992.
- [36] 小山田, 伊藤, 3次元数値シミュレーション結果のための「計測」システム, 情報処理学会ハイパフォーマンスコンピューティング研究報告, 93-HPC-48, pp.121-128, 1993.
- [37] Koyamada K., and Itoh T., *Fast Generation of Spherical Slicing Surfaces for Irregular Volume Rendering*, The Visual Computer, Vol. 11, No. 3, pp. 166-175, 1995.
- [38] Levoy M., *Display of Surfaces from Volume Data*, IEEE Computer Graphics and Applications, Vol. 8, No. 5, pp. 29-37, 1988.
- [39] Levoy M., *Efficient Ray Tracing of Volume Data*, ACM Transactions on Graphics, Vol. 9, No. 3, pp. 245-261, 1990.
- [40] Lischinski D., Tampieri F., and Greenberg D. P., *Combining Hierarchical Radiosity and Discontinuity Meshing*, Computer Graphics, Vol. 27, No. 3, pp. 199-208, 1993.

- [41] Livnat Y., Shen H., and Johnson C. R., *A Near Optimal Isosurface Extraction Algorithm Using the Span Space*, IEEE Transactions on Visualization and Computer Graphics, Vol. 2, No. 1, pp. 73-84, 1996.
- [42] Lo S. H., *Delaunay Triangulation of Non-Convex Planar Domains*, International Journal for Numerical Methods in Engineering, Vol. 28, pp. 2695-2707, 1989.
- [43] Lorensen, W. E., and Cline, H. E., *Marching Cubes: A High Resolution 3D Surface Construction Algorithm*, Computer Graphics, Vol. 21, No. 4, pp. 163-169, 1987.
- [44] Melosh R. J., and Marcal P. V., *An Energy Basis for Mesh Refinement of Structural Continua*, International Journal of Numerical Engineering, Vol. 11, No. 7, pp. 1083-1092, 1977.
- [45] Miller K., *Recent Results on Finite Element Methods with Moving Nodes, Accuracy Estimates and Adaptive Refinements in Finite Element Computations*, Chichester, pp. 225-338, 1986.
- [46] Nishita T., and Nakamae E., *Continuous Tone Representation of Three-dimensional Objects Taking Account of Shadows and Interreflection*, Computer Graphics, Vol. 19, No. 3, pp. 23-30, 1985.
- [47] 西田, 中前, 曲面光源に照射されたパラメトリック曲面の相互反射, 情報処理学会グラフィクスとCAD 研究報告, 90-CG-45-4, 1990.
- [48] 野口, トポロジーって何だろう, ダイヤモンド社, 1986.
- [49] Pavlidis T., *Algorithms for Graphics and Image Processing*, Computer Science Press, 1982.
- [50] Preparata F. P., and Shamos M. I., *Computational Geometry*, Springer-Verlag, 1985.
- [51] Shen H., Hansen C. D., Livnat Y., and Johnson C. R., *Isosurfacing in Span Space with Utmost Efficiency*, Proceedings of IEEE Visualization '96, pp. 287-294, 1996.

- [52] Shephard M. S., *Approaches to the Automatic Generation and Control of Finite Element Meshes*, Applied Mechanical Reviews, Vol. 41, No. 4, pp. 169-185, 1988.
- [53] 嶋田, 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. 11-20, 1993.
- [54] 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.
- [55] Shinagawa Y., and Kunii T. L., *The Homotopy Model: A Generalized Model for Smooth Surface Creation from Cross Sectional Data*, The Visual Computer, Vol. 7, No. 3, pp. 72-86, 1986.
- [56] Sillion F., and Puech C., *A General Two-Pass Method Integrating Specular and Diffuse Reflection*, Computer Graphics, Vol. 23, No.3, pp. 335-344, 1989.
- [57] Silver D., and Zabusky N. J., *Quantifying Visualization for Reduced Modeling in Nonlinear Science: Extracting Structures from Data Sets*, Journal of Visual Communication and Image Representation, Vol. 4, No. 1, pp. 46-61, 1993.
- [58] Sloan S. W., *A Fast Algorithm for Constructing Delaunay Triangulations in the Plane*, Advanced Engineering Software, Vol. 9, No. 1, pp. 34-55, 1987.
- [59] Speray D., and Kennon S., *Volume Probe: Interactive Data Exploration on Arbitrary Grids*, Computer Graphics, Vol. 24, No. 5, pp. 5-12, 1990.
- [60] Thibault W. C., *Set Operations on Polyhedra Using Binary Space Partitioning Trees*, Computer Graphics, Vol. 21, No. 4, pp. 153-162, 1987.
- [61] Totsuka T., and Levoy M., *Frequency Domain Volume Rendering*, Computer Graphics, Vol. 27. No. 3, pp. 271-278, 1993.
- [62] Troutman R., and Max N. L., *Radiosity Algorithms Using Higher Order Finite Element Methods*, Computer Graphics, Vol. 27, No. 3, pp. 209-212, 1993.

- [63] Wallin A., *Constructing Isosurfaces from CT Data*, IEEE Computer Graphics and Applications, Vol. 11, No. 6, pp. 28-33, 1991.
- [64] Weiler K., *The Radial Edge Structure: A Topological Representation for Non-Manifold Geometric Boundary Modeling*, Geometric Modeling for CAD Applications, pp. 3-26, 1986.
- [65] Westover L., *Footprint Evaluation for Volume Rendering*, Computer Graphics, Vol. 24, No. 4, pp. 64-75, 1990.
- [66] Whitted T., *An Improved Illumination Model for Shaded Display*, ACM Communication, Vol. 23, No. 6, pp. 343-349, 1980.
- [67] Welhelms J., and Gelder A. Van, *Octrees for Fast Isosurface Generation*, ACM Transactions on Graphics, Vol. 11, No. 3, pp. 201-227, 1992.
- [68] Williams P. L., *Visibility Ordering Meshed Polyhedra*, ACM Transactions on Graphics, Vol. 11, No. 2, pp. 103-126, 1992.
- [69] Williams P. L., *Interactive Splatting of Nonrectilinear Volumes*, Proceedings of IEEE Visualization '92, pp. 37-44, 1992.
- [70] Wordenweber B., *Finite Element Mesh Generation*, Computer Aided Design, Vol. 16, No. 5, pp. 285-291, 1984.
- [71] Wyvill G., McPheeters C., and Wyvill B., *Data Structure for Soft Objects*, The Visual Computer, Vol. 2, No. 4, pp. 227-234, 1986.
- [72] Zatz H. R., *Galerkin Radiosity: A Higher Order Solution Method for Global Illumination*, Computer Graphics, Vol. 27, No. 3, pp. 213-220, 1993.
- [73] Zienkiewicz O. C., *The Finite Element Method Third Edition*, McGraw-Hill, United Kingdom, 1977.
- [74] Zienkiewicz O. C., and Craig A., *Adaptive Refinement, Error Estimates, Multigrid Solution, and Hierarchic Finite Element Method Concepts*, Accuracy Estimates and Adaptive Refinements in Finite Element Computations, Chichester, pp. 25-59, 1986.

業績

国際論文

1. Itoh T., and Koyamada K., *Automatic Isosurface Propagation by Using an Extrema Graph and Sorted Boundary Cell Lists*, IEEE Transactions on Visualization and Computer Graphics, Vol. 1, No. 4, pp. 319-327, 1995.
2. Koyamada K., and Itoh T., *Fast Generation of Spherical Slicing Surfaces for Irregular Volume Rendering*, The Visual Computer, Vol. 11, No. 3, pp. 166-175, 1995.
3. Koyamada K. and Itoh T., *Seed Specification for Displaying a streamline in an Irregular Volume*, Engineering with Computer, accepted.
4. Doi A. and Itoh T., *A Parallel Radiosity Approach for Complex Environments, Using Hemisphere-Base Formfactor Calculation*, Journal of Computer Animation and Visualization, in process.

国内論文

1. 伊藤, 牧野, 大石, ラジオシティ法を用いた透明体の陰影・集光の表現, 電子情報通信学会論文誌, Vol. J75-D-II, No. 11, pp.1981-1985, 1992.
2. 伊藤, 牧野, 斎藤, 大石, コンピュータグラフィックスにおける分光モデルに関する検討, 電子情報通信学会論文誌, Vol. J75-D-II, No. 11, pp.1986-1989, 1992.
3. 淡路, 細谷, 伊藤, 牧野, 大石, 指の先端位置情報による手の形状モデリング, 電子情報通信学会論文誌, Vol. J75-D-II, No. 11, pp.1978-1980, 1992.

4. 伊藤, 嶋田, 土井, ラジオシティ法のための輝度不連続線およびメッシュの生成に関する効率的な一手法, 情報処理学会論文誌, Vol. 36, No. 11, pp. 2632-2641, 1995.

国際会議

1. Itoh T., and Koyamada K., *Isosurface Generation by Using Extrema Graphs*, Proceedings of IEEE Visualization '94, pp.77-83, 1994.
2. Koyamada K., and Itoh T., *Finding of Critical Points by Using Tetrahedral Cells*, Proceedings of WCCM '94, pp.535-544, 1994.
3. Shimada K., and Itoh T., *Automated Conversion of 2D Triangular Mesh into Quadrilateral Mesh*, Proceedings of the International Conference on Computational Engineering Science '95, pp. 350-355, 1995.
4. Itoh T., Yamaguchi Y., and Koyamada K., *Volume Thinning for Automatic Isosurface Propagation*, Proceedings of IEEE Visualization '96, pp. 303-310, 1996.
5. Shimada K., Mori N., Kondo T., Itoh T., Kase K., and Makinouchi A., *Automated Mesh Generation for Sheet Metal Forming Simulation*, Proceedings of Numisheet '96, pp. 300-307, 1996.
6. Yamada A., Shimada K., and Itoh T., *Energy-minimizing Approach to Meshing Curved Wire-frame Models*, 5th International Meshing Roundtable, pp. 179-191, 1996.

国内会議

1. 伊藤, 牧野, 斎藤, 大石, レイ・トレーシング法における分光モデルに関する検討, 第6回 NICOGRAPH 論文コンテスト論文集, pp.45-51, 1990.
2. 細谷, 伊藤, 牧野, 大石, 光覚の変化を考慮した回転物体の表現法, 第7回 NICOGRAPH 論文コンテスト論文集, pp.77-84, 1991.

3. 小山田, 伊藤, 3次元数値シミュレーション結果のための「計測」システム, 情報処理学会ハイパフォーマンスコンピューティング研究報告, 93-HPC-48, pp.121-128, 1993.
4. 伊藤, 小山田, 極点グラフを用いた等値面生成, 第7回数値流体力学シンポジウム, pp.689-692, 1993.
5. 山田, 波多野, 小山田, 伊藤, 内燃機関用エアクリーナ内部の気流解析結果の可視化, 第7回数値流体力学シンポジウム, pp.705-708, 1993.
6. 伊藤, 嶋田, 土井, ラジオシティ法における輝度不連続境界の算出およびメッシュ生成に関する一手法, 情報処理学会グラフィクスとCADシンポジウム, pp.61-68, 1994.
7. 小山田, 伊藤, CAEポストプロセッサIVSおよび適用事例, 情報処理学会グラフィクスとCAD研究報告, 94-CG-71, pp.1-6, 1994.
8. 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NICOGGRAPH論文コンテスト論文集, pp.44-55, 1994.
9. 伊藤, 山口, 小山田, 細線化処理を用いた高速等値面生成手法, 情報処理学会グラフィクスとCADシンポジウム, pp. 41-49, 1995.
10. 伊藤, 嶋田, 青野, 室橋, 自動四角メッシュ生成手法を用いたラジオシティ法, 第11回NICOGGRAPH論文コンテスト論文集, pp. 105-115, 1995.
11. 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適応的制御方法, 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.
12. 伊藤, 山田, 嶋田, 青野, 大淵, 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, 情報処理学会グラフィクスとCAD研究報告, 96-CG-81, pp. 73-78, 1996.

全国大会

1. 伊藤, 牧野, 大石, レイ・トレーシングにおける虹の表現, 1990年電子情報通信学会春季全国大会, D-607, 1990.

2. 細谷, 伊藤, 牧野, 大石, 光覚の変化を考慮した回転物体の色彩表現法, 1991 年電子情報通信学会秋季全国大会, D-275, 1991.
3. 伊藤, 牧野, 大石, ラジオシティ法による透明凸多面体の陰影・集光の表現, 1991 年電子情報通信学会秋季全国大会, D-277, 1991.
4. 横山, 伊藤, 牧野, 大石, 透明球による間接光を考慮した陰影の表現, 1991 年電子情報通信学会秋季全国大会, D-278, 1991.
5. 伊藤, 牧野, 大石, ラジオシティ法による透明凸多面体の陰影・集光の表現 (2), 1992 年電子情報通信学会春季全国大会, 1992.
6. 淡路, 細谷, 伊藤, 牧野, 大石, 指の先端位置情報による手の形状モデリングに関する研究, 1992 年電子情報通信学会春季全国大会, 1992.
7. 山主, 伊藤, 牧野, 大石, ラジオシティ法による鏡面反射の表現に関する研究, 1992 年電子情報通信学会春季全国大会, 1992.
8. 伊藤, 小山田, 位置をパラメータとした等値面生成ツール, 情報処理学会第 47 回全国大会, 4R-8, 1993.
9. 土井, 伊藤, プログレッシブ・リファインメント・ラジオシティ法における高速化手法, 情報処理学会第 47 回全国大会, 2V-6, 1993.
10. 伊藤, 土井, 嶋田, ラジオシティ法における輝度不連続線処理の低減のための一手法, 情報処理学会第 49 回全国大会, 1R-9, 1994.
11. 土井, 伊藤, RenderMan Interface を用いた高画質レンダリング環境の構築, 情報処理学会第 50 回全国大会, 2C-8, 1995.
12. 土井, 嶋田, 伊藤, ラジオシティ法における Delaunay3 角形を用いた適応的再メッシュ分割法, 平成 7 年度電気関係学会東北支部連合大会, 1995.
13. 伊藤, 嶋田, 四角形メッシュの適応的再分割に関する一手法とラジオシティ法への応用, 情報処理学会第 51 回全国大会, 4S-3, 1995.

14. 嶋田, 伊藤, 三角形メッシュから四角形メッシュへの自動変換方法, 情報処理学会第 51 回全国大会, 7Q-1, 1995.
15. 伊藤, 小山田, 林, 室橋, インターネットを用いた高速等値面表示システム, 情報処理学会第 53 回全国大会, 5Q-4, 1996.

記事

1. 伊藤, 牧野, コンピュータ・グラフィックスの基本的テクノロジー, 情報処理, Vol. 38, No. 3, 1997.
2. 伊藤, 小山田, ボリューム・データのグラフィックス表示, 情報処理, Vol. 38, No. 3, 1997.
3. 青野, 大淵, 伊藤, 写實的レンダリングアルゴリズム, 情報処理, Vol. 38, No. 3, 1997.