

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第3928016号

(P3928016)

(45) 発行日 平成19年6月13日(2007.6.13)

(24) 登録日 平成19年3月16日(2007.3.16)

(51) Int. Cl.

F I

G06F 17/50 (2006.01)

G06F 17/50 612J

G06T 17/20 (2006.01)

G06F 17/50 622C

G06T 17/20

請求項の数 4 (全 32 頁)

(21) 出願番号	特願2004-104732 (P2004-104732)	(73) 特許権者	503360115
(22) 出願日	平成16年3月31日(2004.3.31)		独立行政法人科学技術振興機構
(65) 公開番号	特開2005-293021 (P2005-293021A)		埼玉県川口市本町4丁目1番8号
(43) 公開日	平成17年10月20日(2005.10.20)	(73) 特許権者	501036982
審査請求日	平成16年3月31日(2004.3.31)		萩原 一郎
			神奈川県鎌倉市今泉台5-10-8
		(73) 特許権者	504129146
			篠田 淳一
			千葉県船橋市本町3-12-18
		(74) 代理人	100078776
			弁理士 安形 雄三
		(74) 代理人	100114269
			弁理士 五十嵐 貞喜
		(74) 代理人	100093090
			弁理士 北野 進

最終頁に続く

(54) 【発明の名称】 最大角度法を用いた三角形メッシュ生成方法及びプログラム

(57) 【特許請求の範囲】

【請求項1】

コンピュータの演算処理によって、三次元散乱点群データから初期三角形メッシュを自動生成する方法であって、

c 1 . 前記コンピュータに入力された前記三次元散乱点群データは、八分木アルゴリズムによって木構造が生成され、ポイントリストによって整列されるステップと、

c 2 . 前記木構造を用いて、各点から最短距離にある点を結び、これを長さの短い順に並べ、第1の線分リストを生成し、空の第2の線分リスト及び空の要素リストを用意するステップと、

c 3 . 前記第1の線分リストの先頭にある線分を選択し、選択した線分の最大角度点を求め、選択した線分及びその両端点と前記最大角度点を結んでできた2線分から構成される3線分のうち、前記第2の線分リストに無いものがあれば、これを前記第2の線分リストへ格納し、また、生成した三角形が前記要素リストになければ、これを前記要素リストへ格納するステップと、

c 4 . 生成した三角形を基準面として、その周囲の散乱点の分布状況をトポロジー判断関数によって交差の生成を判断し、交差生成の可能性が大と判断された場合に、生成した三角形を前記要素リストから削除すると共に、選択した線分を前記第1の線分リストから削除し、ステップc 3に戻り、また、交差生成の可能性がないと判断された場合に、ステップc 5に進むようにするステップと、

c 5 . 生成した三角形と共有部分をもつ三角形を求め、求めた三角形と生成した三角形と

10

20

の共有関係によって、交差の生成を判断し、交差生成の可能性が大と判断された場合に、生成した三角形を前記要素リストから削除すると共に、選択した線分を前記第1の線分リストから削除し、ステップc3に戻り、また、交差生成の可能性がないと判断された場合に、既にステップc3で前記要素リストに格納されている生成した三角形は確定し、ステップc6に進むようにするステップと、

c6．選択した線分からのトポロジー判断を利用することによって分断面を生成し、確定した三角形と逆の方向にある散乱点群に対して、ステップc3からステップc5までの処理を実行し、もう一つの三角形を生成するステップと、

c7．選択した線分を前記第1の線分リストから削除し、前記第1の線分リストが空であれば、前記初期三角形メッシュが生成され、処理が終了し、また、前記第1の線分リストが空でなければ、ステップc3に戻るようにするステップと、

10

を有することを特徴とする最大角度法を用いた三角形メッシュ生成方法。

#### 【請求項2】

コンピュータの演算処理によって、請求項1の方法によって生成された初期三角形メッシュに存在する折線ループ内に三角形メッシュを自動生成する方法であって、

d1．前記コンピュータに入力された初期三角形メッシュにおいて、未処理の折線ループを選択するステップと、

d2．選択された折線ループ内に三角形メッシュが存在するかどうかを判断し、選択された折線ループ内に三角形メッシュが存在した場合に、内側の折線ループの最短線分を選択すると共に、外側の折線ループの節点から最大角度点を選択し、ステップd3に進み、一方、選択された折線ループ内に三角形メッシュが存在しないと判断された場合に、折線ループ内の最短線分を選択し、折線ループの節点から最大角度点を選択し、ステップd3に進むようにするステップと、

20

d3．三角形を生成し、三角形の生成によって新しい折線ループが生成されたかどうかを判断し、新しい折線ループが生成された場合に、ステップd2に戻り、一方、新しい折線ループが生成されない場合に、折線ループの処理が終了したかどうかを判断し、折線ループの処理が終了していない場合に、ステップd2に戻るようにするステップと、

d4．ステップd3で折線ループの処理が終了したと判断された場合に、前記初期三角形メッシュに未処理の折線ループが存在するかどうかを判断し、前記初期三角形メッシュに未処理の折線ループが存在した場合に、ステップd1に戻り、一方、前記初期三角形メッシュに未処理の折線ループが存在しない場合に、前記初期三角形メッシュ内の全ての折線ループに三角形メッシュが生成され、三角形メッシュが完成するステップと、

30

を有することを特徴とする最大角度法を用いた三角形メッシュ生成方法。

#### 【請求項3】

三次元散乱点群データから初期三角形メッシュを自動生成するためのコンピュータプログラムであって、

c1．前記三次元散乱点群データを取得し、取得された前記三次元散乱点群データは、八分木アルゴリズムによって木構造が生成され、ポイントリストによって整列される手順と、

c2．前記木構造を用いて、各点から最短距離にある点を結び、これを長さの短い順に並び、第1の線分リストを生成し、空の第2の線分リスト及び空の要素リストを用意する手順と、

40

c3．前記第1の線分リストの先頭にある線分を選択し、選択した線分の最大角度点を求め、選択した線分及びその両端点と前記最大角度点を結んでできた2線分から構成される3線分のうち、前記第2の線分リストに無いものがあれば、これを前記第2の線分リストへ格納し、また、生成した三角形が前記要素リストになれば、これを前記要素リストへ格納する手順と、

c4．生成した三角形を基準面として、その周囲の散乱点の分布状況をトポロジー判断関数によって交差の生成を判断し、交差生成の可能性が大と判断された場合に、生成した三角形を前記要素リストから削除すると共に、選択した線分を前記第1の線分リストから削

50

除し、手順 c 3 に戻り、また、交差生成の可能性がないと判断された場合に、手順 c 5 に進むようにする手順と、

c 5 . 生成した三角形と共有部分をもつ三角形を求め、求めた三角形と生成した三角形との共有関係によって、交差の生成を判断し、交差生成の可能性が大と判断された場合に、生成した三角形を前記要素リストから削除すると共に、選択した線分を前記第 1 の線分リストから削除し、手順 c 3 に戻り、また、交差生成の可能性がないと判断された場合に、既に手順 c 3 で前記要素リストに格納されている生成した三角形は確定し、手順 c 6 に進むようにする手順と、

c 6 . 選択した線分からのトポロジー判断を利用することによって分断面を生成し、確定した三角形と逆の方向にある散乱点群に対して、手順 c 3 から手順 c 5 までの処理を実行し、もう一つの三角形を生成する手順と、

c 7 . 選択した線分を前記第 1 の線分リストから削除し、前記第 1 の線分リストが空であれば、前記初期三角形メッシュが生成され、処理が終了し、また、前記第 1 の線分リストが空でなければ、手順 c 3 に戻るようにする手順と、

をコンピュータに実行させるためのプログラム。

#### 【請求項 4】

請求項 3 のプログラムによって生成された初期三角形メッシュに存在する折線ループ内に三角形メッシュを自動生成するためのコンピュータプログラムであって、

d 1 . 前記初期三角形メッシュを取得し、取得された前記初期三角形メッシュにおいて、未処理の折線ループを選択する手順と、

d 2 . 選択された折線ループ内に三角形メッシュが存在するかどうかを判断し、選択された折線ループ内に三角形メッシュが存在した場合に、内側の折線ループの最短線分を選択すると共に、外側の折線ループの節点から最大角度点を選択し、手順 d 3 に進み、一方、選択された折線ループ内に三角形メッシュが存在しないと判断された場合に、折線ループ内の最短線分を選択し、折線ループの節点から最大角度点を選択し、手順 d 3 に進むようにする手順と、

d 3 . 三角形を生成し、三角形の生成によって新しい折線ループが生成されたかどうかを判断し、新しい折線ループが生成された場合に、手順 d 2 に戻り、一方、新しい折線ループが生成されない場合に、折線ループの処理が終了したかどうかを判断し、折線ループの処理が終了していない場合に、手順 d 2 に戻るようにする手順と、

d 4 . 手順 d 3 で折線ループの処理が終了したと判断された場合に、前記初期三角形メッシュに未処理の折線ループが存在するかどうかを判断し、前記初期三角形メッシュに未処理の折線ループが存在した場合に、手順 d 1 に戻り、一方、前記初期三角形メッシュに未処理の折線ループが存在しない場合に、前記初期三角形メッシュ内の全ての折線ループに三角形メッシュが生成され、三角形メッシュが完成する手順と、

をコンピュータに実行させるためのプログラム。

#### 【発明の詳細な説明】

#### 【技術分野】

#### 【0001】

本発明は、工学、医療、地理学などの多くの分野で広く使用される三角形メッシュを生成する三角形メッシュ生成技術に関し、特に、コンピュータシステムを利用し、二次元空間内若しくは三次元空間内に与えられた散乱点群から、高品質な三角形メッシュを簡単且つ高速に生成できるようにした、三角形メッシュ生成方法及びプログラムに関するものである。

#### 【背景技術】

#### 【0002】

近年、工学、医療、地理学などの多くの分野において、三次元スキャナー等の計測機器で得られた計測データ（つまり、散乱点群）から CAD モデルまたはメッシュモデルを構成し、それに解析に利用するといったことがよく行われている。その際、よく用いられるのが三角メッシュである。そこでは形状の正確な再構成が要求され、また精度のよい解析

10

20

30

40

50

を行うためには高品質なメッシュ、即ち、正三角形に近くなるようなアスペクト比を有する三角形メッシュが必要とされる。

#### 【0003】

散乱点群からの三角形メッシュ生成に関しては、過去に多くの研究がなされてきた。二次元の場合には、Voronoi多角形（以下、ボロノイ多角形と称する）を用いたDelaunayの三角形分割法（以下、ドロネー三角形分割法と称する）が良く知られている。また、ローソンは“四角形を作った場合にその内部に点を持たない任意の四点に対して、この四角形の分割が、分割によってできる二つの三角形において、最小の角が大きくなる”という基準を提案した（非特許文献1を参照）。そして後に、“任意の三点で生成した円の中に、他の点がない場合、この三点によって三角形が構成する”という円を用いた基準も提案した（非特許文献2を参照）。ドロネー三角形分割法がローソンらの手法と同値になることは、シブソンによって証明された（非特許文献3を参照）。

10

#### 【0004】

さらに、これらの高速化アルゴリズムとしては、現在、次の二つの手法がその効率の良さから幅広く使われている。1つ目として、クラインとレンカは、散乱点群を整列化し、そこにローソンによって提出された手法（非特許文献1を参照）を適用する手法を提案した（非特許文献4を参照）。2つ目として、リー、シャハターとブラウンは、平面を三角形に分割して、各三角形内の点群に対してローソンによって提出された手法（非特許文献2を参照）を利用する手法を提案した（非特許文献5と非特許文献6を参照）。

20

#### 【0005】

一方、三次元空間内の散乱点群から曲面三角形メッシュを生成する手法としては、チョウイらがある（非特許文献7を参照）。ここでは分割した各部を射影して二次元化し、そこにドロネー三角形分割法を適用している。この意味でチョウイらの手法は、ドロネー三角形分割法を一般化した方法である。また、ホープは散乱点群を分割し、各部分に付随する小面分とその法線ベクトルを決定し、それらを調整してメッシュを生成する手法を提案した（非特許文献8を参照）。

【非特許文献1】シー・エル・ローソン（C.L.Lawson）著、「ジェネレーション オブ ア トライアングュラー グリッド ウィズ アプリケーション トゥー コンタープロットング, ジェット プロパルション ラボラトリー インターナル テクニカル メモランダム (Generation of a triangular grid with application to contour plotting, Jet Propulsion Laboratory Internal Technical Memorandum)」、1972年,第299号

30

【非特許文献2】シー・エル・ローソン（C.L.Lawson）著、「ソフトウェア フォア シーワン サーフィス インターポーレーション, イン マセマティカル ソフトウェア スリー, イー・デー・ジェイ・アール, ライス, アカデミック プレス (Software for C1 surface interpolation, in Mathematical Software III, ed.J.R.Rice, Academic Press)」、米国,1977年,p.161

【非特許文献3】アール・シブソン（R.Sibson）著、「ローカリー イークウィアングュラー トライアングュレーション, コンピューター ジャーナル (Locally equiangular triangulations, The Computer Journal)」、1978年,第121巻,第3号,p.243

40

【非特許文献4】エー・ケー・クライン（A.K.Cline）、アール・エール・レンカ（R.L.Reynolds）共著、「ストーリージ エフィシヤント メソッド フォア コンストラクション オブ サイエssen トライアングュレーション, ロッキー マウンテン ジャーナル オブ マセマティックス (A storage-efficient method for construction of a thiesen triangulation, Rocky Mountain Journal of Mathematics)」、1984年,第14巻,第1号,p.119

【非特許文献5】デー・デー・リー（D.T.Lee）、ビー・ジェイ・シャハター（B.J.Schachter）共著、「ツー アルゴリズム フォア コンストラクティング ドロネー トライアングュレーション, インターナショナル ジャーナル オブ コンピュータ アンド

50

インフォメーションサイエンス (Two algorithms for constructing a Delaunay Triangulation, International Journal of Computer and Information Sciences)」、1980年、第9巻、第3号、p1691

【非特許文献6】ジェイ・エル・ブラウン (J.L.Brown) 著、「ヴェクテックス ベースド データ ディペンデント トライアングレーション、コンピュータ エイデッド ジオメトリック デザイン (Vertex based data dependent triangulation, Computer Aided Geometric Design)」、1991年、第8巻、第3号、p.239

【非特許文献7】ビー・ケー・チョウイ (B.K.Choi)、エイチ・ワイ・シン (H.Y.Shin)、ワイ・アイ・ヨン (Y.I.Yoon)、ジェイ・ダブリュー・リー (J.W.Lee) 共著、「トライアングレーション オブ スキャタード データ イン スリーデー スペース、コンピュータ エイデッド デザイン (Triangulation of scattered data in 3D space, Computer-Aided Design)」、1988年、第20巻、第5号、p.239

【非特許文献8】エイチ・ホープ (H.Hoppe) 著、「サーフィス リコンストラクション フロム アンオーガナイズド ポイント、ドクトラル シーシス、ユニバーシティ オブ ワシントン (Surface reconstruction from unorganized points, Doctoral thesis, University of Washington)」、1994年

【発明の開示】

【発明が解決しようとする課題】

【0006】

ところが、チョウイらの手法については、対象物の複雑度が増せばその分対象領域を細かく分割しなければならないといった問題があり、また、領域の分割の仕方も問題になってくる。そして、ホープの手法については、メッシュ生成のためには補助的に莫大な数の点を必要とし、計算時間の面からすると非常にコストがかかってしまうという問題がある。また、ホープの手法については微細な構造の欠損といった問題も報告されている。

【0007】

本発明は、上述のような事情よりなされたものであり、本発明の目的は、コンピュータシステムを利用し、二次元空間内若しくは三次元空間内に与えられた散乱点群から、高品質な三角形メッシュを簡単且つ高速に生成できるようにした三角形メッシュ生成方法及びプログラムを提供することにある。

【課題を解決するための手段】

【0008】

本発明は、三角形メッシュ生成方法に関し、本発明の上記目的は、コンピュータの演算処理によって、三次元散乱点群データから初期三角形メッシュを自動生成する方法であって、c1. 前記コンピュータに入力された前記三次元散乱点群データは、八分木アルゴリズムによって木構造が生成され、ポイントリストによって整列されるステップと、c2. 前記木構造を用いて、各点から最短距離にある点を結び、これを長さの短い順に並べ、第1の線分リストを生成し、空の第2の線分リスト及び空の要素リストを用意するステップと、c3. 前記第1の線分リストの先頭にある線分を選択し、選択した線分の最大角度点を求め、選択した線分及びその両端点と前記最大角度点を結んでできた2線分から構成される3線分のうち、前記第2の線分リストに無いものがあれば、これを前記第2の線分リストへ格納し、また、生成した三角形が前記要素リストになれば、これを前記要素リストへ格納するステップと、c4. 生成した三角形を基準面として、その周囲の散乱点の分布状況をトポロジー判断関数によって交差の生成を判断し、交差生成の可能性が大と判断された場合に、生成した三角形を前記要素リストから削除すると共に、選択した線分を前記第1の線分リストから削除し、ステップc3に戻り、また、交差生成の可能性がないと判断された場合に、ステップc5に進むようにするステップと、c5. 生成した三角形と共有部分をもつ三角形を求め、求めた三角形と生成した三角形との共有関係によって、交差の生成を判断し、交差生成の可能性が大と判断された場合に、生成した三角形を前記要素リストから削除すると共に、選択した線分を前記第1の線分リストから削除し、ステップc3に戻り、また、交差生成の可能性がないと判断された場合に、既にステップc3で

前記要素リストに格納されている生成した三角形は確定し、ステップc6に進むようにするステップと、c6. 選択した線分からのトポロジ判断を利用することによって分断面を生成し、確定した三角形と逆の方向にある散乱点群に対して、ステップc3からステップc5までの処理を実行し、もう一つの三角形を生成するステップと、c7. 選択した線分を前記第1の線分リストから削除し、前記第1の線分リストが空であれば、前記初期三角形メッシュが生成され、処理が終了し、また、前記第1の線分リストが空でなければ、ステップc3に戻るようにするステップとを有することにより、或いは、コンピュータの演算処理によって、初期三角形メッシュに存在する折線ループ内に三角形メッシュを自動生成する方法であって、d1. 前記コンピュータに入力された初期三角形メッシュにおいて、未処理の折線ループを選択するステップと、d2. 選択された折線ループ内に三角形メッシュが存在するかどうかを判断し、選択された折線ループ内に三角形メッシュが存在した場合に、内側の折線ループの最短線分を選択すると共に、外側の折線ループの節点から最大角度点を選択し、ステップd3に進み、一方、選択された折線ループ内に三角形メッシュが存在しないと判断された場合に、折線ループ内の最短線分を選択し、折線ループの節点から最大角度点を選択し、ステップd3に進むようにするステップと、d3. 三角形を生成し、三角形の生成によって新しい折線ループが生成されたかどうかを判断し、新しい折線ループが生成された場合に、ステップd2に戻り、一方、新しい折線ループが生成されない場合に、折線ループの処理が終了したかどうかを判断し、折線ループの処理が終了していない場合に、ステップd2に戻るようにするステップと、d4. ステップd3で折線ループの処理が終了したと判断された場合に、前記初期三角形メッシュに未処理の折線ループが存在するかどうかを判断し、前記初期三角形メッシュに未処理の折線ループが存在した場合に、ステップd1に戻り、一方、前記初期三角形メッシュに未処理の折線ループが存在しない場合に、前記初期三角形メッシュ内の全ての折線ループに三角形メッシュが生成され、三角形メッシュが完成するステップとを有することによって効果的に達成される。

【0009】

また、本発明は、三角形メッシュ生成プログラムに関し、本発明の上記目的は、三次元散乱点群データから初期三角形メッシュを自動生成するためのコンピュータプログラムであって、c1. 前記三次元散乱点群データを取得し、取得された前記三次元散乱点群データは、八分木アルゴリズムによって木構造が生成され、ポイントリストによって整列される手順と、c2. 前記木構造を用いて、各点から最短距離にある点を結び、これを長さの短い順に並べ、第1の線分リストを生成し、空の第2の線分リスト及び空の要素リストを用意する手順と、c3. 前記第1の線分リストの先頭にある線分を選択し、選択した線分の最大角度点を求め、選択した線分及びその両端点と前記最大角度点を結んでできた2線分から構成される3線分のうち、前記第2の線分リストに無いものがあれば、これを前記第2の線分リストへ格納し、また、生成した三角形が前記要素リストになければ、これを前記要素リストへ格納する手順と、c4. 生成した三角形を基準面として、その周囲の散乱点の分布状況をトポロジ判断関数によって交差の生成を判断し、交差生成の可能性が大と判断された場合に、生成した三角形を前記要素リストから削除すると共に、選択した線分を前記第1の線分リストから削除し、手順c3に戻り、また、交差生成の可能性がないと判断された場合に、手順c5に進むようにする手順と、c5. 生成した三角形と共有部分をもつ三角形を求め、求めた三角形と生成した三角形との共有関係によって、交差の生成を判断し、交差生成の可能性が大と判断された場合に、生成した三角形を前記要素リストから削除すると共に、選択した線分を前記第1の線分リストから削除し、手順c3に戻り、また、交差生成の可能性がないと判断された場合に、既に手順c3で前記要素リストに格納されている生成した三角形は確定し、手順c6に進むようにする手順と、c6. 選択した線分からのトポロジ判断を利用することによって分断面を生成し、確定した三角形と逆の方向にある散乱点群に対して、手順c3から手順c5までの処理を実行し、もう一つの三角形を生成する手順と、c7. 選択した線分を前記第1の線分リストから削除し、前記第1の線分リストが空であれば、前記初期三角形メッシュが生成され、処理が終

10

20

30

40

50

了し、また、前記第1の線分リストが空でなければ、手順c3に戻るようにする手順とを有することにより、或いは、初期三角形メッシュに存在する折線ループ内に三角形メッシュを自動生成するためのコンピュータプログラムであって、d1.前記初期三角形メッシュを取得し、取得された前記初期三角形メッシュにおいて、未処理の折線ループを選択する手順と、d2.選択された折線ループ内に三角形メッシュが存在するかどうかを判断し、選択された折線ループ内に三角形メッシュが存在した場合に、内側の折線ループの最短線分を選択すると共に、外側の折線ループの節点から最大角度点を選択し、手順d3に進み、一方、選択された折線ループ内に三角形メッシュが存在しないと判断された場合に、折線ループ内の最短線分を選択し、折線ループの節点から最大角度点を選択し、手順d3に進むようにする手順と、d3.三角形を生成し、三角形の生成によって新しい折線ループが生成されたかどうかを判断し、新しい折線ループが生成された場合に、手順d2に戻り、一方、新しい折線ループが生成されない場合に、折線ループの処理が終了したかどうかを判断し、折線ループの処理が終了していない場合に、手順d2に戻るようにする手順と、d4.手順d3で折線ループの処理が終了したと判断された場合に、前記初期三角形メッシュに未処理の折線ループが存在するかどうかを判断し、前記初期三角形メッシュに未処理の折線ループが存在した場合に、手順d1に戻り、一方、前記初期三角形メッシュに未処理の折線ループが存在しない場合に、前記初期三角形メッシュ内の全ての折線ループに三角形メッシュが生成され、三角形メッシュが完成する手順とを有することによって効果的に達成される。

10

#### 【発明の効果】

20

#### 【0010】

本発明に係る二次元空間における最大角度法を用いた三角形メッシュ生成方法によれば、平面上の散乱点群に対しては、ドロネー三角形分割法と同等の機能を有し、かつ計算コストにおいても直接角度の計算をするのではなく、内積の大小によって角度の大小を判断でき、それ以外の条件判断を必要としないので、従来他の手法に比べて高速な処理が可能になるという優れた効果を奏する。

#### 【0011】

また、本発明に係る三次元空間における最大角度法を用いた三角形メッシュ生成方法によれば、非特許文献7に開示されたチョウイらの手法に用いられる射影平面も使用せず、三次元空間内に与えられた散乱点群に属する全ての点を効率よく利用することが可能であり、複雑なモデルの形状の再構成も、トポロジー判断関数の導入により満足のいく再構成の結果が得られるといった優れた効果を奏する。

30

#### 【発明を実施するための最良の形態】

#### 【0012】

平面上の散乱点群から三角形メッシュを生成する最も一般的な方法としてのドロネー三角形分割法においては、必ず各点から最短距離にある点を接続した線分が選択され、また各線分に対する対応角は最大となっている。

#### 【0013】

本発明では、ドロネー三角形分割法のこの特徴を考慮に入れて、二次元空間内の散乱点群にだけでなく、三次元空間内の散乱点群にも拡張し、初期最短線分群を構成した後に、初期最短線分群に対応して最大角度をとる点から三角形要素(以下、三角形とも略する)を生成することによって、高品質な三角形メッシュを簡単に生成するようにしている。これは本発明の第一の着眼点である。

40

#### 【0014】

本発明のもう一つの着眼点とは、三次元空間内の散乱点群から三角形メッシュを生成する際に、発生する三角形要素が交差してしまう現象をいかに克服するかということである。これに対しては、本発明では、後述する<2-1>『三次元空間におけるメッシュの交差についての検討』及び<2-2>『三次元空間における交差の発見と回避』に詳細に説明されたように、まず、三次元空間における交差現象の原因を究明し、そして、トポロジー判断関数により交差が起きそうな場所を特定し、そのような場所(そこには穴が発生す

50

る)については一旦保留し、後にその部分が実際の穴かどうかを判断し、そうでなければ内部に三角形メッシュを生成するという方法を用いて、三次元空間におけるメッシュの交差といった厄介な問題を簡単に解決した。

【0015】

以下、本発明の実施の形態を図面を参照して説明する。

【0016】

本発明は、コンピュータシステムを利用し、二次元空間内(平面内)若しくは三次元空間内に与えられた散乱点群から、最大角度法を用いて、高品質な三角形メッシュを高速に生成する三角形メッシュ生成方法及びプログラムに関する。以下では、本発明の三角形メッシュ生成のアルゴリズムを二次元空間と三次元空間の場合に分けて説明する。

10

【0017】

なお、以下に説明する本発明に係る三角形メッシュ生成方法及び三角形メッシュ生成アルゴリズムに使用される散乱点群として、例えば、実際に三次元スキャナー等の計測機器で実物の表面形状を計測して得られた点群データ(座標値)を用いることが可能である。また、医療の現場において使用されるCTやMRIといった医療機器から得られた病巣部に関する点群データを用いることも可能である。さらに、コンピュータグラフィック(CG)によって生成した点群データを使用することも勿論可能である。座標値で表す、二次元或いは三次元の点群データであれば、本発明を適用し、高品質な三角形メッシュを簡単に生成することができる。

【0018】

20

なお、本発明に係る三角形メッシュ生成アルゴリズムは、ソフトウェアにまたはハードウェアによって実装されることが可能である。勿論、本発明に係る三角形メッシュ生成アルゴリズムは、ソフトウェアとハードウェアの適宜な組み合わせによって実装されることも可能である。

< 1 > 二次元空間(平面)における最大角度法を用いた三角形メッシュ生成方法

まず、本発明で用いられる技術用語の説明をする。ABCにおいて、ACBを線分ABに対する対応角と呼び、点Cの候補が複数存在する場合に、対応角が最大となる点を最大角度点と呼ぶことにする。また、初期散乱点群の中にある点を一つ固定し、その点とその点から最短距離にある点を結んでできる線分をその点に対する最短線分と呼ぶことにする。そして、最短線分を基準に最大角度点を探索し、三角形メッシュを生成する方法を最大角度法と呼ぶ。

30

【0019】

本発明では、この最大角度法を用いて三角形メッシュを生成するので、つまり、線分とその対応角による組を基に三角形メッシュを生成しているので、新しくできた三角形要素が今までに生成された三角形要素と交差を起こす可能性について、検討の必要性がある。以下では、まずこのような交差現象について論じることにする。

< 1 - 1 > 二次元空間における三角形要素交差についての検討

まず、図1に示されるように、二つの三角形が線分ABに対して、最大角度点Cが選択され、ABCが形成されている。即ち、 $a_1 > b_1$ となっている。このとき、図1において、Cと接続する線分CDに対して、最大角度点の選択はAとなり、三角形要素の交差は発生しない。

40

【0020】

また、図2は対象となる線分がADとDCというように共有点を持つ場合を示す。図2の場合も同様にして、線分ADに対して生成されたADBに対して、線分DCに対して選択される最大角度点もBとなり、三角形要素の交差を起こさない。

【0021】

最後に、図3は最初に生成される三角形と次の対象となる線分が共有部分を持たない場合を示す。図3の場合では、それぞれ線分ABに対する最大角度点がE、線分CDに対す

50



る最大角度点がFとなっている。このとき定義からそれぞれの外接円内には他の点を含まないが、交差を起こす場合に必ずE、Fのどちらかが他方の外接円に含まれてしまうので、実際に三角形要素の交差は発生しない。

#### 【0022】

上述した二次元空間（平面）における最大角度法の性質から、生成された三角形の辺を用いて隣接する三角形を逐次生成する方法が考えられるが、以下では三次元空間への拡張も考えて、2通りの最大角度法を用いた三角形メッシュ生成アルゴリズムを示す。

< 1 - 2 > 二次元空間における最大角度法を用いた三角形メッシュ生成アルゴリズム 1（以下、単に片側アルゴリズムとも称する）

10

片側アルゴリズムは以下ようになる。

ステップA1：

二次元空間内に与えられた散乱点群に属するすべての点は、四分木法によって木構造を生成し、ポイントリストによって整理される。

ステップA2：

ポイントリストの先頭の点を取り、関係する木構造点から、その点の最短線分を生成する。

ステップA3：

ステップA2で生成された最短線分に対して、木構造の中の関係する点から最大角度点を探し、新しい三角形と線分を生成する。但し、最大角度点となる点が複数存在する場合（例えば、図4に示すように、線分ABに対する最大角度点として、複数の点、つまり、点C、点D及び点Eが存在する場合）には、最初に選択した点を優先するものとする（図4の場合は、点Cを最大角度点とする）。

20

ステップA4：

ステップA3で生成された線分を線分リストに入れる。

ステップA5：

線分リストの順序に従って、線分毎に、ステップA3と同様にして、新しい三角形と線分を生成し、生成された線分について、ステップA4と同様にして、線分リストに入れる。

。

ステップA6：

二次元空間内に与えられた散乱点群に属する全ての点が、三角形メッシュ構造に含まれるまで、ステップA5を繰り返す。

30

#### 【0023】

上述した片側アルゴリズムをフローチャートに示すと、図5になる。

#### 【0024】

図5に示されるように、片側アルゴリズムに基づいて本発明の三角形メッシュ生成方法では、まず、例えば計測機器で得られた二次元空間内の点群データ（座標値）を入力し（ステップS101）、入力された点群データは、四分木アルゴリズムによって整列される（ステップS102）。整列された点群データから初期点を選択し（ステップS103）、選択された初期点に対する最短線分を生成する（ステップS104）。最短線分に対して半平面内に点が存在するかどうかを判断する（ステップS105）。

40

#### 【0025】

最短線分に対して半平面内に点が存在しない場合に、逆側の半平面から最大角度点を探し（ステップS106）、ステップS108に進む。一方、最短線分に対して半平面内に点が存在した場合に、半平面内から最大角度点を選択し（ステップS107）、三角形要素を生成する（ステップS108）。そして、生成した三角形を要素リストへ格納し（ステップS109）、新しく生成された線分を線分リストに追加する（ステップS110）。次に、未チェックの線分が線分リストにあるかどうかを判断する（ステップS111）。

#### 【0026】

50

未チェックの線分が線分リストにある場合に、線分リストに線分を選択し（ステップ S 1 1 2）、三角形の無い側に点が存在するかどうかを判断する（ステップ S 1 1 3）。三角形の無い側に点が存在した場合に、ステップ S 1 0 7 に戻る。一方、三角形の無い側に点が存在しない場合に、ステップ S 1 1 1 に戻る。また、ステップ S 1 1 1 において、未チェックの線分が線分リストに無いと判断された場合に、二次元空間内の点群データから三角形メッシュが生成され、処理が終了する。

< 1 - 3 > 二次元空間における最大角度法を用いた三角形メッシュ生成アルゴリズム 2（以下、単に両側アルゴリズムとも称する）

次に、三次元空間の場合に拡張するために、各辺の両側で三角形を生成する両側アルゴリズムについて述べる。

10

ステップ B 1 :

二次元空間内に与えられた散乱点群に属するすべての点は、四分木法によって木構造を生成し、ポイントリストによって整理される。

ステップ B 2 :

ステップ B 1 で生成された木構造を用いて、各点から最短距離にある点を結び、これを長さの短い順に並べ、線分リストを作る。

ステップ B 3 :

線分リストの順序に従って、対応する線分の両側において、例えば図 6 に示されるように、木構造の中の関係する点から最大角度点を探し、新しい三角形と線分を生成する。

20

ステップ B 4 :

ステップ B 3 で生成された線分を線分リストに入れる。

ステップ B 5 :

二次元空間内に与えられた散乱点群に属する全ての点が、三角形メッシュ構造に含まれるまで、ステップ B 3 とステップ B 4 を繰り返す。

【 0 0 2 7 】

上述した両側アルゴリズムをフローチャートに示すと、図 7 になる。

【 0 0 2 8 】

図 7 に示されるように、両側アルゴリズムに基づいて本発明の三角形メッシュ生成方法では、まず、例えば計測機器で得られた二次元空間内の点群データ（座標値）を入力し（ステップ S 2 0 1）、入力された点群データは、四分木アルゴリズムによって整列される（ステップ S 2 0 2）。整列された点群データにおいて、各点から最短距離にある点を探索して接続する（ステップ S 2 0 3）。そして、長さの短い順に並べた線分リストを作成する（ステップ S 2 0 4）。

30

【 0 0 2 9 】

次に、線分リストの最初の線分を選択し（ステップ S 2 0 5）、選択された線分の両側でそれぞれ最大角度点を選択し（ステップ S 2 0 6）、三角形要素を生成する（ステップ S 2 0 7）。生成された三角形要素を要素リストへ格納し（ステップ S 2 0 8）、新しく生成された線分を線分リストに追加する（ステップ S 2 0 8）。

【 0 0 3 0 】

40

次に、未チェックの線分が線分リストにあるかどうかを判断する（ステップ S 2 1 0）。未チェックの線分が線分リストにある場合に、ステップ S 2 0 5 に戻る。一方、未チェックの線分が線分リストに無い場合に、二次元空間内の点群データから三角形メッシュが生成され、処理が終了する。

< 1 - 4 > 片側アルゴリズムと両側アルゴリズムとの比較

上述した片側アルゴリズムと両側アルゴリズムとを比較すると、片側アルゴリズムは、初期線分群の生成とその長さによる線分リストの生成（つまり、両側アルゴリズムのステップ B 2）が無い分、両側アルゴリズムより高速であると考えられる。また、結果として得られる三角形メッシュは、両アルゴリズムにおいて一致する。

50

< 1 - 5 > 本発明に係る二次元空間における最大角度法を用いた三角形メッシュ生成アルゴリズムと従来のドロネー三角形分割法との比較

次に、本発明に係る二次元空間における最大角度法を用いた三角形メッシュ生成アルゴリズムと従来のドロネー三角形分割法とを比較してみる。

【 0 0 3 1 】

まず、従来のドロネー三角形分割法の概略を述べる。ドロネー三角形分割法において、まず、二次元空間における有限個の点の集合  $S = \{P_k\}$  をとる。各  $k$  に対して、 $k$  と異なる  $l$  に対して、 $|PP_k| > |PP_l|$  を満たす点  $P$  の集合  $V_k$  をポロノイ多角形または Thiessen 多角形（以下、ティーセン多角形と称する）と呼ぶ。ポロノイ多角形は、その定義から  $P_k$  を含む閉凸多角形となり、その全体は空間全体を埋め尽くす。このポロノイ多角形による分割をディリクレ分割（Dirichlet tessellation）と呼ぶ。このポロノイ多角形を用いると、ドロネー三角形分割法による三角形分割は、次のようにして得られる。

10

【 0 0 3 2 】

即ち、 $S$  の点  $P_k$  と対応するポロノイ多角形  $V_k$  に隣接する各ポロノイ多角形  $V_l$  に対応する点  $P_l$  とを結ぶ。上記の連結法により、 $S$  の点を用いた分割が構成される。このとき、この分割は三角形と凸多角形によって構成されている。 $P_k$  を含む凸多角形の境界線は実は、 $P_k$  とある  $P_l$  を接続した線分の垂直二等分線となっている。凸多角形は、さらに三角形に容易に分割されるので、これにより全て三角形の分割が得られる。これをドロネー三角形分割法と呼ぶ。図 8 においては、細線によってドロネー三角形分割を表し、太線によって囲まれた図形が各ポロノイ多角形になっている。

20

【 0 0 3 3 】

上述したドロネー三角形分割法で得られた三角形メッシュ（以下、ドロネー三角形メッシュとも称する）は、次の性質を有している。まず、ドロネー三角形メッシュにおいて、例えば図 9 に示されるように、各線分に対する対応角は必ずこの方向に関して最大となる。次に、例えば図 10 に示されるように、ドロネー三角形メッシュの任意の節点に対して、その点の最短線分は必ずメッシュの中に存在する。

【 0 0 3 4 】

上述した 2 つの性質は、本発明の両側アルゴリズムの中でも使われている性質であり、実際、本発明の平面における最大角度法を用いた三角形メッシュ生成アルゴリズムで得られる三角形メッシュとドロネー三角形分割法で得られる三角形メッシュは一致する。

30

< 1 - 6 > 二次元空間における最大角度法を用いた三角形メッシュ生成アルゴリズムの適用例

ここで、本発明に係る片側アルゴリズムを実際の平面散乱点群に適用した例を挙げることにする。図 11 の初期散乱点群に対して、図 12 は点 1 から最短距離にある点 9 を結んで最短線分を生成したものである。次に、この最短線分を基にして最大角度点となる点 10 を用いて最初の三角形（これを A とおく）が生成されたのが図 13 である。そして、この三角形を構成する 3 線分、即ち、点 1 と点 10 を結ぶ線分（これを a とおく）、点 9 と点 10 を結ぶ線分（これを b とおく）、点 1 と点 9 を結ぶ線分が、線分リストに格納される。そこで、線分 a, b に対して、三角形 A がある側とは反対の側で三角形をそれぞれ生成したのが図 14 と図 15 である。この処理を繰り返していけば、図 16 を経て図 17 に示すように三角形メッシュが完成する。

40

< 2 > 三次元空間における最大角度法を用いた三角形メッシュ生成方法

最大角度法を平面上の散乱点群に適用する場合、一部の例外を除いて辺の交差は発生しない。しかし、三次元空間内の散乱点群に関しては、状況が平面の場合に比べて格段に複雑になる。また、元々のモデルを散乱点群から正確に再構築できるかどうかという問題は非常に重要であるが、散乱点群のみの情報だけでは、元の曲面の様子、例えばシャープな形状や微小な穴などを完全に予測することは困難である。

50

## 【 0 0 3 5 】

以下では、このような交差現象の排除と形状の正確な再構成という両面からの要請をできるだけ満足するメッシュの構築を実現するためにメッシュ構造の分析を行なう。

## &lt; 2 - 1 &gt; 三次元空間におけるメッシュの交差についての検討

三次元空間内でメッシュの交差として考えられるものには、図 1 8 に示すような三つのタイプがある。但し、見易さのために、前面にあるものを一部濃色を用いて表現している。

## 【 0 0 3 6 】

まず、タイプ A (貫通の場合)とは、図 1 8 ( a ) に示すように、文字通りある三角形が他の三角形内を通過してしまう場合である。次に、タイプ B (面の不整合の場合)とは、図 1 8 ( b ) に示すように、以後の三角形の生成によって、面が形成されなくなる場合である。そして、タイプ C (一辺を三つ以上の三角形が共有する場合)とは、ある線分が 3 つ以上の三角形に共有される場合である。タイプ C の例として、メッシュ内に不要な小さな多面体が生成されるものや、図 1 8 ( c ) の右端のように、曲面に小片が接続したりするものがある。

10

## 【 0 0 3 7 】

ここで、一つの線分 A B とその最大角度点 P に対して、

$$\{P \in R^3 \mid \angle APB = \angle AQB\}$$

20

は一つの閉曲面をなす。この閉曲面によって囲まれた有界な領域を線分 A B から最大角度法によって決まる領域と定義する。このとき、定義からこの領域内には点が存在しない。図 1 9 では、線分 A B と線分 C D それぞれから最大角度法によって決まる領域を表わしている。

## 【 0 0 3 8 】

タイプ A の交差が存在するのは、図 1 9 に示すように、対象となる二つの三角形によって決まるそれぞれの平面の交線 (図 1 9 では、点 E と点 F を通る直線) が、二つの三角形の中を同時に通過する場合である。要素数が多い場合、従来の方法では非常に多くの交差判断の計算が必要となるが、本発明の最大角度法では、平面の場合と同様にして、この種の交差を避けることができる。

30

## 【 0 0 3 9 】

また、本発明の最大角度法は、線分に対して三角形を生成してゆく手法であるので、タイプ B の交差を持つメッシュに対して、繰り返し最大角度法を適用すれば、境界線のみを持つ三角形が、面を持つ三角形データとして認識され、四面体状の構造へと変化していく。これは、即ち、タイプ B の形状がタイプ C の形状へ帰着されることを意味する。従って、交差に関しては、タイプ C の場合についてのみ検討すればよいが、実は、近接する 4 点の範囲では、このような交差は発生しない。しかしながら、メッシュ全体から見ると、このような交差は、確かに発生しうる。これは、既に完成した三角形と周囲の点の分布と線分の選択との矛盾によって起こる。

40

## 【 0 0 4 0 】

交差生成の原因としては、2 つ考えられる。即ち、例えば図 2 0 ( a ) に示されるように、底部の三角形 (つまり、図中の B C D) が生成された後に、他の点からやや距離のある点 A によって生成された三角形 (つまり、図中の A B C、 A C D、 A B D) によって、四面体 (つまり、図中の四面体 A B C D) が生成してしまうもの (つまり、既存メッシュとの矛盾) と、例えば図 2 0 ( b ) に示されるように、襞状の構造が発生してしまうもの (つまり、襞状構造の生成) の 2 種類である。

## &lt; 2 - 2 &gt; 三次元空間における交差の発見と回避

50

本発明では、三次元空間内の散乱点のトポロジー判断、即ち、散乱点群の分布の状況を検査することによって、襞状構造の生成を検出するようにしている。そのために、本発明では、２段階に分けた制御を設けて、襞状構造の生成を検出する。

【 0 0 4 1 】

第 1 段階の制御とは、図 2 1 に示すような個々の三角形と周囲の点との関係による制御であり、第 2 段階の制御とは、図 2 2 及び図 2 3 に示すような線分（つまり、辺）若しくは点を共有する二つの三角形の関係から制御するものである。

【 0 0 4 2 】

まず、第 1 段階の制御では、最大角度法によって生成した三角形を基準面として、周囲の点の分布の特徴を後述する本発明のトポロジー判断関数を利用して判断する。つまり、第 1 段階の制御では、図 2 1 に示されるように、最大角度法によって生成した三角形の重心 D、当該三角形の周囲にある一つの点 P、及び点 P の当該三角形面への正射影 I を用いた関数

【 0 0 4 3 】

【 数 1 】

$$T = f(|DP|, |DI|)$$

によって、当該三角形の周囲の点の分布を判断する。但し、 $|DP|$  は点 D と点 P の距離を表わす。 $|DI|$  は点 D と点 I の距離を表わす。本発明では、上記関数 T をトポロジー判断関数と呼ぶ。

【 0 0 4 4 】

例えば、トポロジー判断関数 T の最も簡単な例は

【 0 0 4 5 】

【 数 2 】

$$T = \frac{|DP|^2}{|DI|^2}$$

である。この例において、 $T < a$ （但し、 $a$  は 1 より大きい定数）を考えると、これは点 P が同じ頂点を持つ二つの円錐に挟まれた領域に存在することを意味する。上記数 2 のトポロジー判断関数 T を用いると、角部とエッジ部に相当する点に対しては、T の値が非常に大きくなる。上記数 2 のトポロジー判断関数 T に基づいて算出された、周囲の点の T の値が大きければ、襞状の構造が生成される可能性が高いと判断して、当該三角形（つまり、最大角度法によって生成した三角形）の生成をやめる。

【 0 0 4 6 】

また、トポロジー判断関数 T の別な例としては

【 0 0 4 7 】

【 数 3 】

$$T = \left(\frac{b}{|PD|}\right)^2 \times \left(\frac{|PI|}{|DI|}\right)^c$$

がある。ここで、 $b$  は当該三角形（つまり、最大角度法によって生成した三角形）に関するパラメータで、 $c$  は正定数である。但し、 $|PD|$  は点 P と点 D の距離を表わし、 $|PI|$  は点 P と点 I の距離を表わし、 $|DI|$  は点 D と点 I の距離を表わす。同様に、上記数 3 のトポロジー判断関数 T に基づいて算出された、周囲の点の T の値が大きければ、襞状の構造が生成される可能性が高いと判断して、当該三角形（つまり、最大角度法によって生成した三角形）の生成をやめる。

10

20

30

40

40

## 【 0 0 4 8 】

次に、第 2 段階の制御では、第 1 段階の制御で得られた三角形（つまり、第 1 段階の制御を行ってから残れた、最大角度法によって生成した三角形）とそれに接続する三角形との関係判断する。同一平面上にない二つの三角形が共通部分を持つとすると、それは、図 2 2 のように 1 辺のみを共有するか、また図 2 3 のように 1 点のみを共有するかのいずれかの場合である。

## 【 0 0 4 9 】

図 2 2 において、角 A は共有辺の中点を始点とする二つの中線のなす角であるのに対して、図 2 3 において、角 A は共有点を始点とする二つの中線のなす角である。また、図 2 2 及び図 2 3 において、角 B は二つの三角形の法線ベクトルのなす角を表わしている。そして、図 2 2 及び図 2 3 において、もし、角 A と角 B が両方ともある値  $b$  より小さい場合に、襷状の構造を生成したと判断し、第 1 段階の制御で得られた三角形の生成をやめる。

10

## 【 0 0 5 0 】

次に、基準となる線分（以下、基準線分とも称する）に対して、確定した三角形とは逆の側に三角形を生成する。具体的には、まず、図 2 4 に示すように、基準線分の中点  $M$  を中心とする一つの球を生成する。この球は、以下の処理に必要な点数を絞り込むために用いられる。次に、この球に含まれる点を基準線分の二つの端点と接続し、三角形を生成する。生成した三角形の平均法線ベクトルを求め、この基準線分を通過する平均平面を生成する。そして、基準線分の中点  $M$  を通過し、且つ平均平面と垂直になる平面（以下、垂直平面とも称する）を生成する。この垂直平面によって、球に含まれる全ての点は二つ異なるグループに分けられる。つまり、球は二つの領域に分けられる。分けられた領域のうち、確定した三角形の無い側で最大角度点を探し、最大角度法によって三角形を生成し、生成された三角形について、上述した 2 段階の制御を行って交差の判断を行なう。

20

< 2 - 3 > 三次元空間における最大角度法を用いた三角形メッシュ生成アルゴリズム 1（以下、単に三次元空間における初期メッシュ生成アルゴリズム、或いは、初期メッシュ生成アルゴリズムと称する）

三次元空間内に与えられた散乱点群から初期三角形メッシュを生成する、初期メッシュ生成アルゴリズムは、以下のようなになる。

ステップ C 1 :

30

三次元空間内に与えられた散乱点群に属するすべての点は、八分木法によって木構造を生成し、ポイントリストによって整理される。

ステップ C 2 :

ステップ C 1 で生成された木構造を用いて、各点から最短距離にある点を結び、これを長さの短い順に並べ、線分リスト A を作る。さらに、空の線分リスト B 及び空の要素リスト C を用意する。

ステップ C 3 :

線分リスト A の先頭にある線分、即ち、最小の長さのもの  $l$  を選択し、選択した線分の最大角度点  $M$  を求める。線分  $l$  および  $l$  の両端点と点  $M$  を結んでできた 2 線分から構成される 3 線分のうち、線分リスト B に無いものがあれば、これを線分リスト B に加える。また、出来た三角形が要素リスト C になければ、これを要素リスト C へ加える。もし、最大角度点が複数あれば、次のような処理をする。即ち、線分  $AB$  を基準にして  $ACB = ADB = AEB = \dots$  となっている場合、もし既に、 $C, D, E, \dots$  の中に  $AXB$  が要素リスト C の中に存在するような点  $X$  があれば、線分  $AB$  はチェック済みとし、そうでないときは、 $|XAB - XBA|$  が最小になる点  $X$  を  $C, D, E, \dots$  の中から選ぶ。この段階でも、まだそのような点が複数あれば、任意の一つを既に構成された三角形要素と交差を起こさないように選ぶ。

40

ステップ C 4 :

生成した三角形を基準面として、その周囲の点の分布状況をトポロジー判断関数  $T$  によって判断する。周囲の点の  $T$  の値が定数  $a$  より小さい場合は、ステップ C 5 へと進み、そ

50

うでないならば、生成した三角形を削除すると共に選択した線分を線分リスト A から消去し、ステップ C 3 に戻る。

ステップ C 5 :

生成した三角形と共有部分（つまり、共有点或いは共有辺のいずれか）をもつ三角形を求め、そして、求めた三角形と生成した三角形との共有関係によって、図 2 2 または図 2 3 のいずれかに示された角度 A と角度 B を計算する。もし、角度 A と角度 B の両方が共に小さい場合には、交差が生成されると判断する。共有点を一つ持つ場合には、生成した三角形を削除する（つまり、新しい三角形の生成をやめる）とと共に選択した線分を線分リスト A から消去し、ステップ C 3 に戻る。一つの線分（つまり、辺）を共有する場合には、辺を共有する二つの三角形の T の平均値を求め、より大きい値を持つ方の三角形の生成は行わない。

10

ステップ C 6 :

もし、この三角形の線分が既に二つの三角形に共有されている場合は、この三角形の生成は行なわない。但し、ここでは要素として認識されていない線分のみ三角形も含む。もし、この三角形が既に完成している場合も、新しい三角形は生成しない。

ステップ C 7 :

次に、線分からのトポロジー判断法を利用する。分断面を生成し、今生成した方向とは逆の方向にある点群に対して、ステップ C 3 からステップ C 6 までのステップを実行し、もう一つの三角形を生成する。

ステップ C 8 :

線分リスト A から線分 1 を削除する。新しい線分と三角形を線分リスト A と要素リスト C へ添加する。

20

ステップ C 9 :

線分リスト A が空でなければ、ステップ C 3 に戻り、もし空であればこの一連の処理を終了する。

【 0 0 5 1 】

上述した初期メッシュ生成アルゴリズムをフローチャートに示すと、図 2 5 になる。

【 0 0 5 2 】

図 2 5 に示されるように、初期メッシュ生成アルゴリズムに基づいて本発明の三角形メッシュ生成方法では、まず、例えば三次元スキャナ等の計測機器で得られた三次元空間内の点群データ（座標値）を入力し（ステップ S 3 0 1）、入力された点群データは、八分木アルゴリズムによって整列される（ステップ S 3 0 2）。整列された点群データにおいて、各点から最短距離にある点を探索して接続する（ステップ S 3 0 3）。そして、長さの短い順に並べた線分リストを作成する（ステップ S 3 0 4）。

30

【 0 0 5 3 】

次に、線分リストの最初の線分を選択し（ステップ S 3 0 5）、選択した線分の中点を中心とした半径 R の球を設定する（ステップ S 3 0 6）。選択した線分に対して球内の点から最大角度点を選択し（ステップ S 3 0 7）、三角形要素を生成する（ステップ S 3 0 8）。次に、トポロジー判断関数 T を適用する（ステップ S 3 0 9）。

【 0 0 5 4 】

次に、球内の全ての点に対して  $T < a$  が成立するかどうかを判断する（ステップ S 3 1 0）。球内の全ての点に対して  $T < a$  が成立しない場合に、生成した三角形要素を削除すると共に（ステップ S 3 1 1）、選択した線分を線分リストから削除し（ステップ S 3 1 2）、ステップ S 3 0 5 に戻る。一方、ステップ S 3 1 0 において、球内の全ての点に対して  $T < a$  が成立すると判断された場合に、生成した三角形要素と共有部分を持つ三角形要素を選択する（ステップ S 3 1 3）。

40

【 0 0 5 5 】

そして、生成した三角形要素及び選択された共有部分を持つ三角形要素に対しての角 A と角 B が共に b より小さいかどうかを判断する（ステップ S 3 1 4）。角 A と角 B が共に b より小さい場合に、生成した三角形要素を削除すると共に（ステップ S 3 1 1）、選択

50

した線分を線分リストから削除し（ステップS312）、ステップS305に戻る。

【0056】

一方、ステップS314において、角 $A < b$ と角 $B < b$ が同時に成立しない場合に、生成した三角形要素を要素リストへ格納し（ステップS315）、新しく生成された線分を線分リストに追加する（ステップS316）。次に、選択した線分の両側で三角形要素の生成が終了したかどうかを判断する（ステップS317）。選択した線分の両側で三角形要素の生成が終了していない場合に、選択した線分の両端点と球内の点を結び、三角形を生成する（ステップS318）。そして、ステップS318で生成した三角形の法線ベクトルを計算する（ステップS319）。

【0057】

次に、球内の全ての点に対して、ステップS318からステップS319までの処理が終了したかどうかを判断する（ステップS320）。球内の全ての点に対して、ステップS318からステップS319までの処理が終了していない場合に、ステップS318に戻り、ステップS318からステップS319までの処理を繰り返す。

【0058】

一方、ステップS320において、球内の全ての点に対して、ステップS318からステップS319までの処理が終了したと判断された場合に、ステップS319で計算された全ての三角形の法線ベクトルに基づいて、平均法線ベクトルを計算する（ステップS321）。そして、選択した線分を含み、且つ、平均法線ベクトルに垂直な平面を計算する（ステップS322）。球面と平面に囲まれた領域のうち、三角形を含まない方

【0059】

一方、ステップS317において、選択した線分の両側で三角形要素の生成が終了したと判断された場合に、未チェックの線分が線分リストにあるかどうかを判断する（ステップS324）。未チェックの線分が線分リストにある場合に、選択した線分を線分リストから削除し（ステップS325）、そして、ステップS305に戻る。一方、未チェックの線分が線分リストに無い場合に、三次元空間内の点群データから初期三角形メッシュが生成され（ステップS326）、それで初期メッシュ生成アルゴリズムに基づく処理が終了する。

< 2 - 4 > 三次元空間における最大角度法を用いた三角形メッシュ生成アルゴリズム 2（以下、単に三次元空間におけるループ内三角形メッシュ生成アルゴリズム、或いは、ループ内三角形メッシュ生成アルゴリズムと称する）

本発明では、三次元空間内に与えられた散乱点群から、上述した初期メッシュ生成アルゴリズムによって初期三角形メッシュを生成し、そして、後述するループ内三角形メッシュ生成アルゴリズムによって三角形メッシュを完成する。

【0060】

初期メッシュ生成アルゴリズムで初期三角形メッシュを生成した時点で、線分リストAにあった線分は、全て要素リストCの中のある三角形の一辺になっていて、これで三角形メッシュの一部が出来たことになる。しかしながら、初期メッシュ生成アルゴリズムの段階では、交差が起きそうな部分において、三角形の生成を中止するなどの措置を取ったため、その部分に必要な応じてメッシュを補填しなければならない。そのような部分は穴のあいた状態になっていて、その穴の周囲は、例えば図26に示すような折線によるループになっている。以下では、そのループ内のメッシュ生成アルゴリズム（つまり、ループ内三角形メッシュ生成アルゴリズム）を述べる。

【0061】

初期メッシュ生成アルゴリズムで初期メッシュを生成する段階で、できた折線ループ内にメッシュを生成する、ループ内三角形メッシュ生成アルゴリズムは、以下のようになる。

ステップD1：

10

20

30

40

50



折線ループ内に一部のメッシュがあれば、ステップD4に進み、そうでなければ、ステップD2に進む。

ステップD2（折線ループ内部の三角形分割1）：

折線ループに含まれる線分のうち、最短な線分 $l$ を一つとり、この線分 $l$ に対する折線ループ内の最大角度点 $P$ を求める。線分 $l$ の両端点を $Q$ 、 $R$ とおく。折線ループで囲まれた部分から $PQR$ を取り除けば、例えば図27に示すように、元々の折線ループは、高々二つの折線ループに分割されたと考えることができる。即ち、図27(A)では、 $PQR$ と折線ループの共通部分が一辺（つまり、線分 $QR$ ）のみであり、 $PQR$ の挿入により二つの折線ループが新しく生成される。また、図27(B)では、隣接する二辺（つまり、線分 $QR$ と線分 $QP$ ）が $PQR$ と折線ループによって共有され、頂点数が一つ少なくなった折線ループが新しく生成される。

10

ステップD3：

これら新しく生成された折線ループに対して、ステップD2に戻って処理をする。対象となる折線ループがなくなれば、処理を全て終了し、そうでなければステップD1に戻る。

ステップD4（ループ内部の三角形分割2）：

例えば図28に示すように、内側のループに含まれる線分のうち、最短なものを一つとり、外側のループの頂点から最大角度点を選ぶ。これでできた三角形を要素リストに加えれば、分割されていない部分は、図28の点線に示されるように一つのループによって囲まれたことになるので、ステップD2に進む。

20

【0062】

ステップD1からステップD4までの手順により、ループ内の三角形メッシュが生成される。

【0063】

なお、図29から図34までは、少ない点数（本例では、121点）の三次元空間内の散乱点群に対して、上述した初期メッシュ生成アルゴリズムとループ内三角形メッシュ生成アルゴリズムとから構成される、本発明に係る三次元空間における最大角度法を用いた三角形メッシュ生成アルゴリズムを用いて、三角形メッシュ生成を行なった逐次的結果である。図29で初期線分群が生成され、図30から図33まで逐次的に三角形メッシュが生成され、図34で三角形メッシュが完結している。

30

【0064】

上述したループ内三角形メッシュ生成アルゴリズムをフローチャートに示すと、図35になる。

【0065】

図35に示されるように、ループ内三角形メッシュ生成アルゴリズムに基づいて本発明の三角形メッシュ生成方法では、まず、初期メッシュ生成アルゴリズムに基づいた本発明の三角形メッシュ生成方法によって生成された初期三角形メッシュを入力とし（ステップS401）、入力された初期三角形メッシュにおいて、未チェックのループ $L$ を選択する（ステップS402）。選択されたループ $L$ に囲まれた領域にメッシュが存在するかどうかを判断する（ステップS403）。選択されたループ $L$ に囲まれた領域にメッシュが存在した場合に、内側のループの最短線分を選択すると共に（ステップS404）、外側のループの節点から最大角度点を選択し（ステップS405）、そして、後述するステップS408に進む。

40

【0066】

一方、ステップS403において、選択されたループ $L$ に囲まれた領域にメッシュが存在しないと判断された場合に、ループ $L$ 内の最短線分を選択する（ステップS406）。そして、ループ $L$ の節点から最大角度点を選択し（ステップS407）、三角形要素を生成する（ステップS408）。次に、新しいループが生成されたかどうかを判断する（ステップS409）。新しいループが生成された場合に、ステップS403に戻る。一方、ステップS409において、新しいループが生成されないと判断された場合に、ループ $L$

50

の処理が終了したかどうかを判断する（ステップS410）。ループLの処理が終了していない場合に、ステップS403に戻る。

【0067】

一方、ステップS410において、ループLの処理が終了したと判断された場合に、初期三角形メッシュに未チェックのループが存在するかどうかを判断する（ステップS411）。初期三角形メッシュに未チェックのループが存在した場合に、ステップS402に戻る。一方、ステップS411において、初期三角形メッシュに未チェックのループが存在しないと判断された場合に、三角形メッシュが完結する（ステップS412）。

< 2 - 5 > 三次元空間における最大角度法を用いた三角形メッシュ生成アルゴリズムの適用例 10

ここで、本発明に係る三次元空間における最大角度法を用いた三角形メッシュ生成アルゴリズムを実際の三次元空間内の散乱点群に適用した三つの例を挙げることにする。なお、これらの例に用いるデータは、ウェブ上のメッシュデータから散乱点の座標データを抽出して利用したものである。

【0068】

まず、第1例はモアイ像である。図36は、モアイ像のモデルに対して、初期散乱点群から各点の最短距離にある点を結んでできる初期線分群を表したものである。ここで、初期散乱点の個数は10,002点である。図37は、図36に示されたモアイ像に対して、本発明の初期メッシュ生成アルゴリズムを適用した結果である。もし、本発明の初期メッシュ生成アルゴリズムを適用した段階で、交差が起きなければ、そこで三角形メッシュの生成は終わるはずである。 20

【0069】

しかし、図37及び図37に示されたモアイ像の一部の拡大図である図38に示されるように、本発明の初期メッシュ生成アルゴリズムを適用した段階で発生した交差によって、実際にはモアイ像に多くの穴が発生していることがよく分かる。そこで、こういった穴、つまり、ループへの対処アルゴリズムである、本発明のループ内三角形メッシュ生成アルゴリズムを適用すれば、図39に示されるように、モアイ像の表面の復元は終了する。このとき、三角形要素数は20,016個である。

【0070】

次に、第2例は、図40及び図41に示されるスタンフォード・バニーである。図40及び図41においては、本発明の三次元空間における最大角度法を用いた三角形メッシュ生成アルゴリズムによって、三次元空間内の35,947点の散乱点群から71,939個の三角形要素が生成されているスタンフォード・バニーを示している。図42に示したものは、図40及び図41に示されるスタンフォード・バニーの耳の内側からメッシュを見たものである。このような複雑な形状を持ったモデルに対しては、図43のオリジナルのメッシュと比べても分かるように、右下に見られるような滑らかでないメッシュが発生してしまう可能性がまだ残っている。 30

【0071】

最後に、挙げる第3例は、図44及び図45に示されるヴィーナス像である。図44及び図45においては、本発明の三次元空間における最大角度法を用いた三角形メッシュ生成アルゴリズムによって、三次元空間内の67,180点の散乱点群から134,495個の三角形要素が生成されているヴィーナス像を示している。図44及び図45に示されるヴィーナス像を生成するのにかかった所要時間は、Pentium4（登録商標）、1.6GHzのマシンで約1時間である。 40

【0072】

図46は、オリジナルデータによるヴィーナス像の一部である口元周辺の拡大図である。但し、メッシュの見易さのため、内側から見た図になっている。それに対して、図47は本発明の三次元空間における最大角度法を用いた三角形メッシュ生成アルゴリズムによって得られたものであり、図46と同じ部分を表わしている。図47からはヴィーナス像 50

の口元の傷の部分も詳細に再構成されているのが見て取れる。また、図46と図47を三角形メッシュで表現したものは、それぞれ図48と図49になる。図48に示されるように、オリジナルの三角形メッシュが幾分不均一であるのに対して、図49に示されるように、本発明の三次元空間における最大角度法を用いた三角形メッシュ生成アルゴリズムによる生成された三角形メッシュは、均一的なものであることがよく分かる。

### < 3 > 本発明のまとめ

本発明は、二次元空間内または三次元空間内の与えられた散乱点群から高精度な三角形メッシュを構成するための最大角度法及びそれを用いた三角形メッシュ生成方法に関するものである。本発明とは、散乱点群に属する各散乱点から最短距離にある点を結び、その最短線分に対する最大角度点を探索して、三角形要素を構成するというシンプルな原理（つまり、最大角度法）を基にして、散乱点群から三角形メッシュを生成する方法である。

#### 【0073】

特に、本発明に係る三次元空間における最大角度法を用いた三角形メッシュ生成方法は、射影平面をまったく用いず、三次元空間内の複雑な曲面を直接構成するのに非常に好都合である。本発明に係る三次元空間における最大角度法を用いた三角形メッシュ生成方法では、メッシュの交差の解析を行うと共にトポロジー判断関数の導入によって交差が起きそうな部分を予め探知し、それを回避することができるようにし、また、トポロジー判断関数を用いることにより、生成した三角形の妥当性も判断できるようにしている。さらに、本発明では、トポロジー判断関数を用いれば、シャープエッジやコーナーなどの鋭角的な部分の特徴をも再現することが可能になる。

#### 【図面の簡単な説明】

#### 【0074】

【図1】本発明において、二次元空間における三角形要素交差の検討を行うための模式図である。

【図2】本発明において、二次元空間における三角形要素交差の検討を行うための模式図である。

【図3】本発明において、二次元空間における三角形要素交差の検討を行うための模式図である。

【図4】本発明において、ある線分に対して複数の最大角度点が存在するのを説明するための模式図である。

【図5】本発明に係る片側アルゴリズムのフローチャートである。

【図6】本発明に係る両側アルゴリズムにおいて、線分の両側にそれぞれ生成された三角形を説明するための模式図である。

【図7】本発明に係る両側アルゴリズムのフローチャートである。

【図8】従来のドロネー三角形分割法を説明するための模式図である。

【図9】従来のドロネー三角形分割法によって得られた三角形メッシュの性質を説明するための模式図である。

【図10】従来のドロネー三角形分割法によって得られた三角形メッシュの性質を説明するための模式図である。

【図11】本発明に係る片側アルゴリズムに用いられる初期平面散乱点群の一例を示す図である。

【図12】本発明に係る片側アルゴリズムを図11に示された散乱点群に適用した際に生成された初期線分を示す図である。

【図13】本発明に係る片側アルゴリズムを図11に示された散乱点群に適用した際の三角形メッシュ生成過程を示す図である。

【図14】本発明に係る片側アルゴリズムを図11に示された散乱点群に適用した際の三角形メッシュ生成過程を示す図である。

【図15】本発明に係る片側アルゴリズムを図11に示された散乱点群に適用した際の三角形メッシュ生成過程を示す図である。

10

20

30

40

50

【図16】本発明に係る片側アルゴリズムを図11に示された散乱点群に適用した際の三角形メッシュ生成過程を示す図である。

【図17】図11に示された散乱点群から本発明に係る片側アルゴリズムに基づいて生成された三角形メッシュを示す図である。

【図18】本発明において、三次元空間における三角形要素交差のタイプを説明するための模式図である。

【図19】本発明において、三次元空間における線分から最大角度法によって決まる領域を説明するための模式図である。

【図20】本発明において、三次元空間における三角形要素の交差生成の原因を説明するための模式図である。

10

【図21】本発明におけるトポロジー判断関数を説明するための模式図である。

【図22】1つの共有辺を持つ二つの三角形の関係を説明するための模式図である。

【図23】1つの共有点を持つ二つの三角形の関係を説明するための模式図である。

【図24】本発明において、基準線分の中点を中心とする球の生成と平均法線ベクトルを説明するための模式図である。

【図25】本発明に係る初期メッシュ生成アルゴリズムのフローチャートである。

【図26】本発明に係る初期メッシュ生成アルゴリズムの段階で発生した折線ループを説明するための模式図である。

【図27】折線ループ内部の三角形分割を説明するための模式図である。

【図28】折線ループ内に一部のメッシュが存在する場合のループの三角形分割を説明するための模式図である。

20

【図29】本発明に係る三次元空間における最大角度法を用いた三角形メッシュ生成アルゴリズムによって三次元空間内の散乱点群から生成された初期線分群を示す図である。

【図30】本発明に係る三次元空間における最大角度法を用いた三角形メッシュ生成アルゴリズムを三次元空間内の散乱点群に適用した際の三角形メッシュ生成過程を示す図である。

【図31】本発明に係る三次元空間における最大角度法を用いた三角形メッシュ生成アルゴリズムを三次元空間内の散乱点群に適用した際の三角形メッシュ生成過程を示す図である。

【図32】本発明に係る三次元空間における最大角度法を用いた三角形メッシュ生成アルゴリズムを三次元空間内の散乱点群に適用した際の三角形メッシュ生成過程を示す図である。

30

【図33】本発明に係る三次元空間における最大角度法を用いた三角形メッシュ生成アルゴリズムを三次元空間内の散乱点群に適用した際の三角形メッシュ生成過程を示す図である。

【図34】図29に示された初期線分群から本発明に係る三次元空間における最大角度法を用いた三角形メッシュ生成アルゴリズムに基づいて生成された三角形メッシュを示す図である。

【図35】本発明に係るループ内三角形メッシュ生成アルゴリズムのフローチャートである。

40

【図36】三次元初期散乱点群から生成された初期線分群によって表されるモアイ像を示す図である。

【図37】図36に示されたモアイ像に対して、本発明に係る初期メッシュ生成アルゴリズムを適用した結果を示す図である。

【図38】図37に示されたモアイ像の一部の拡大図である。

【図39】図37に示されたモアイ像に対して、本発明に係るループ内三角形メッシュ生成アルゴリズムを適用した結果を示す図である。

【図40】35,947点の三次元初期散乱点群から本発明の三次元空間における最大角度法を用いた三角形メッシュ生成アルゴリズムによって生成された、71,939個の三角形要素から構成される三角形メッシュによって表したスタンフォード・バニーを示す図

50

である。

【図 4 1】35,947 点の三次元初期散乱点群から本発明の三次元空間における最大角度法を用いた三角形メッシュ生成アルゴリズムによって生成された、71,939 個の三角形要素から構成される三角形メッシュによって表したスタンフォード・バニーを示す図である。

【図 4 2】図 4 0 及び図 4 1 に示されるスタンフォード・バニーの耳の内側を表す、本発明によって生成された三角形メッシュを示す図である。

【図 4 3】スタンフォード・バニーのオリジナルの三角形メッシュを示す図である。

【図 4 4】67,180 点の三次元初期散乱点群から本発明の三次元空間における最大角度法を用いた三角形メッシュ生成アルゴリズムによって生成された、134,495 個の三角形要素から構成される三角形メッシュによって表したヴィーナス像を示す図である。

10

【図 4 5】67,180 点の三次元初期散乱点群から本発明の三次元空間における最大角度法を用いた三角形メッシュ生成アルゴリズムによって生成された、134,495 個の三角形要素から構成される三角形メッシュによって表したヴィーナス像を示す図である。

【図 4 6】オリジナルデータによるヴィーナス像の一部である口元周辺の拡大図である。

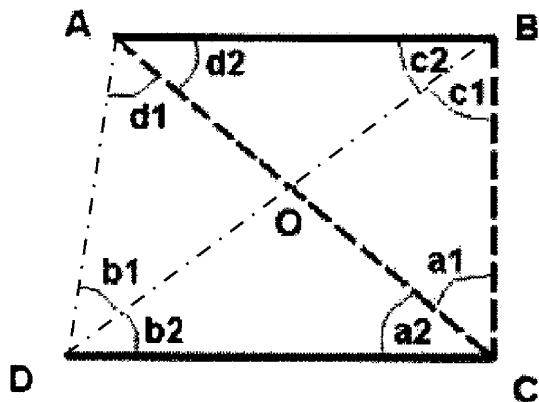
【図 4 7】本発明の三次元空間における最大角度法を用いた三角形メッシュ生成アルゴリズムによる、ヴィーナス像の一部である口元周辺の拡大図である。

【図 4 8】図 4 6 に示されたヴィーナス像の一部である口元周辺のオリジナルの三角形メッシュを示す図である。

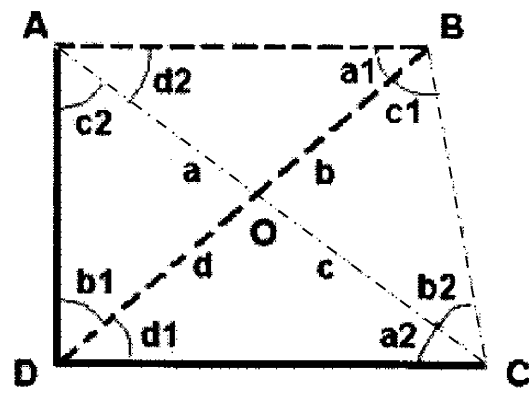
【図 4 9】図 4 7 に示されたヴィーナス像の一部である口元周辺の本発明によって生成された三角形メッシュを示す図である。

20

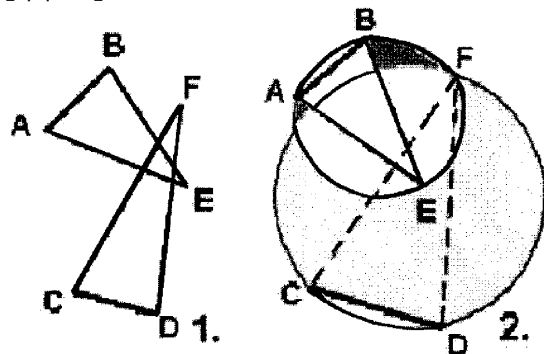
【図 1】



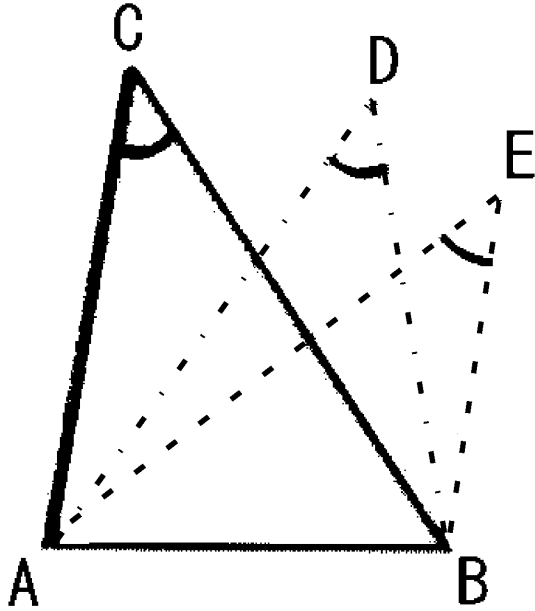
【図 2】



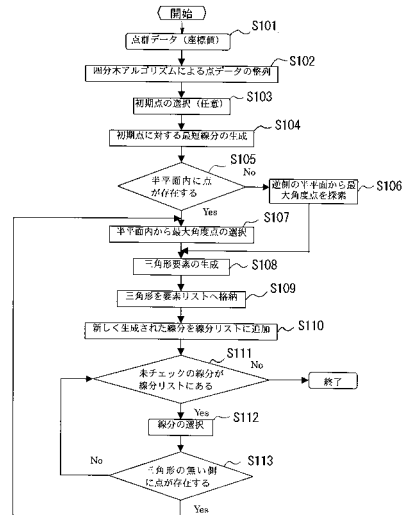
【図 3】



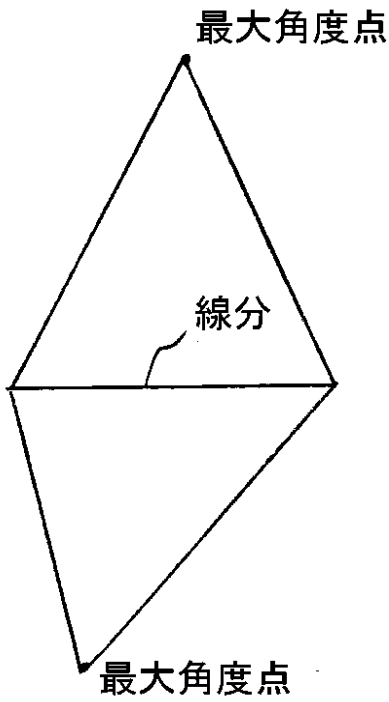
【 図 4 】



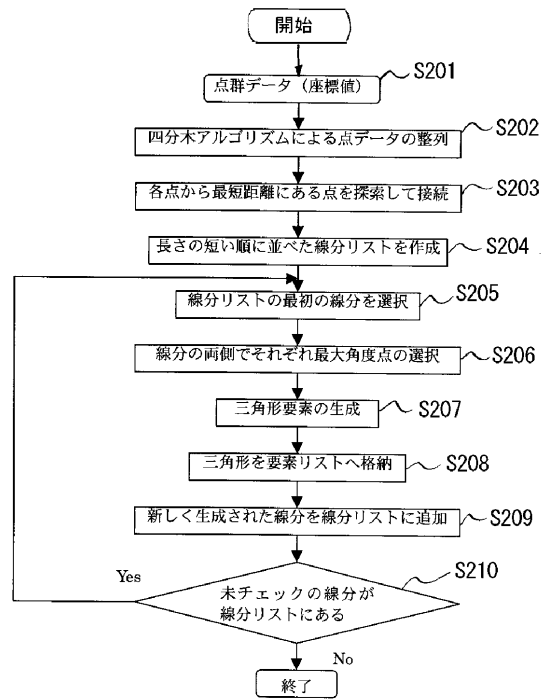
【 図 5 】



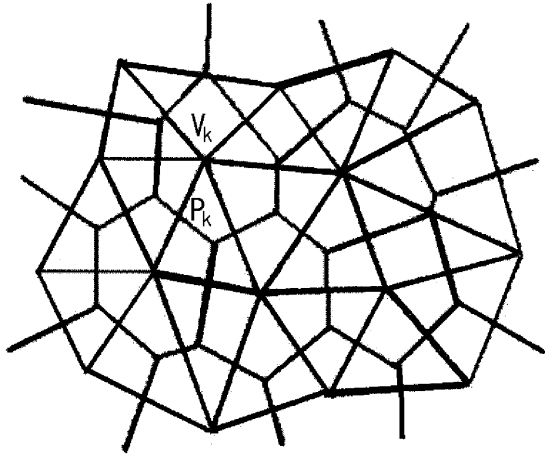
【 図 6 】



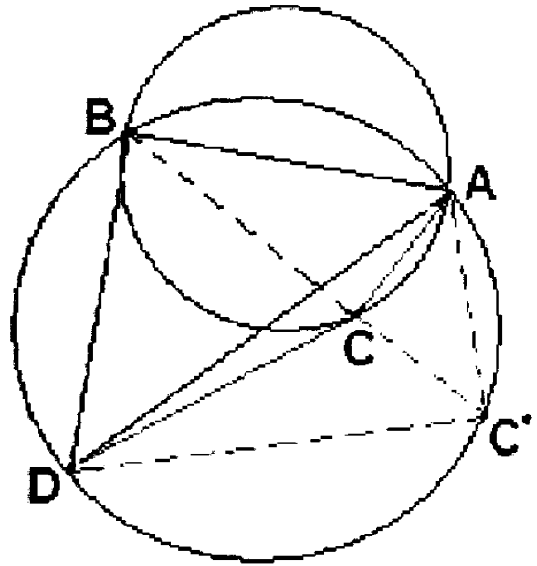
【 図 7 】



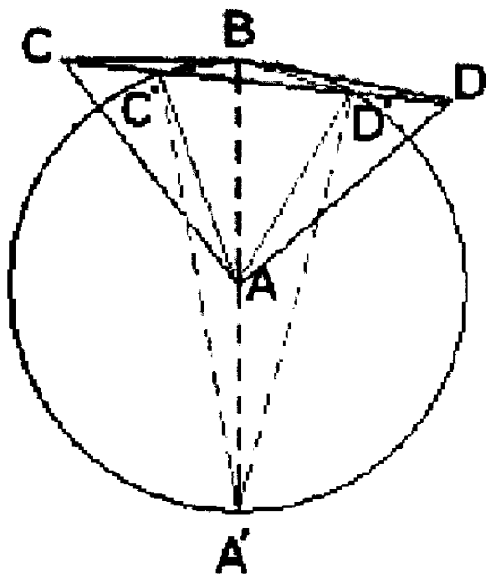
【 図 8 】



【 図 9 】



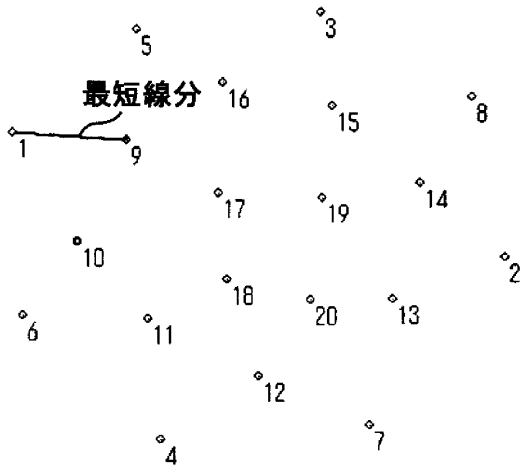
【 図 10 】



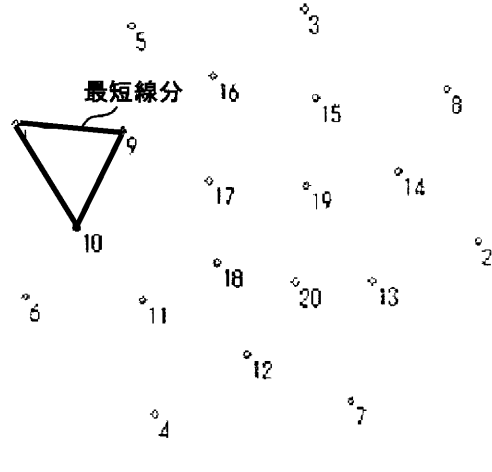
【 図 11 】



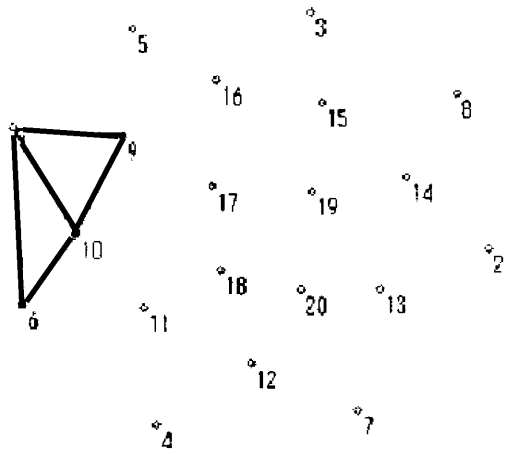
【 図 1 2 】



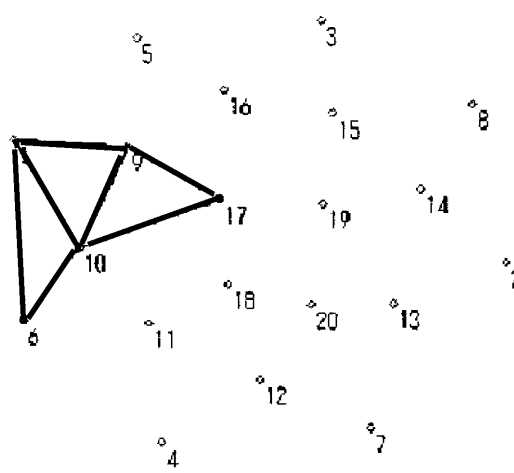
【 図 1 3 】



【 図 1 4 】

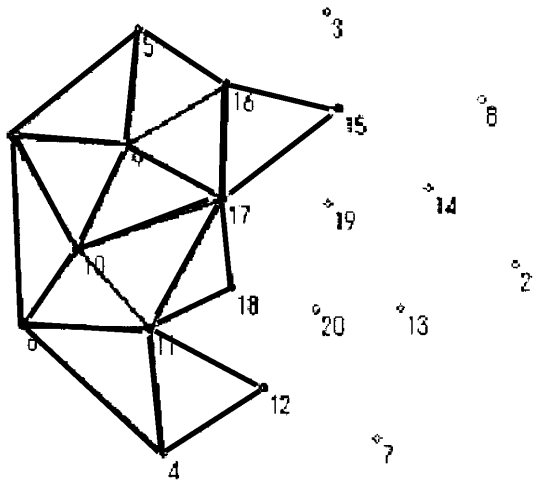


【 図 1 5 】

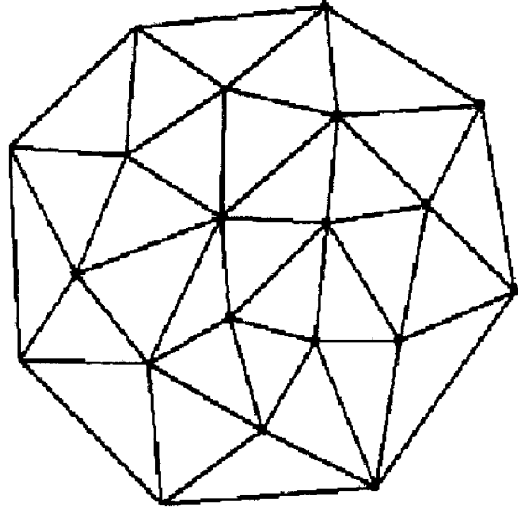




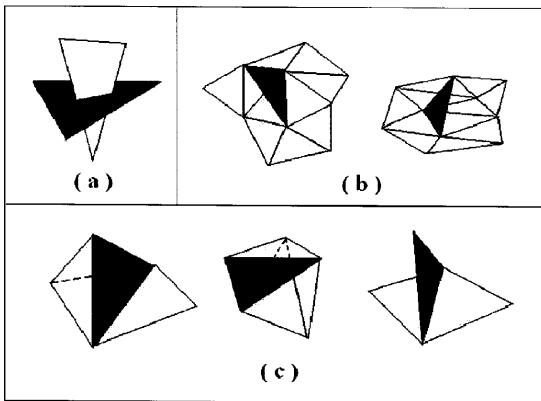
【 図 16 】



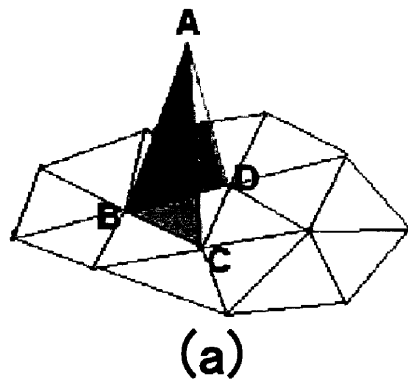
【 図 17 】



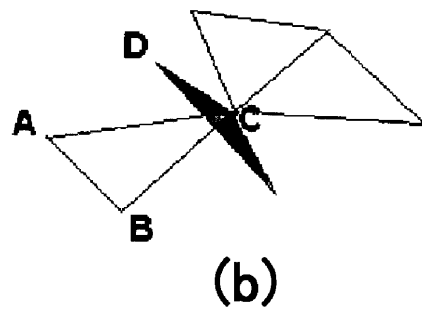
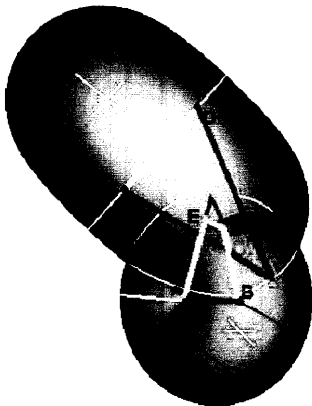
【 図 18 】



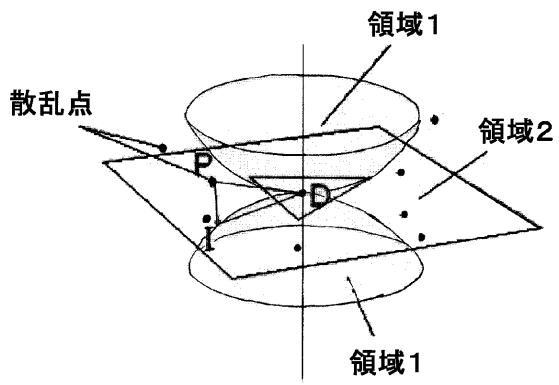
【 図 20 】



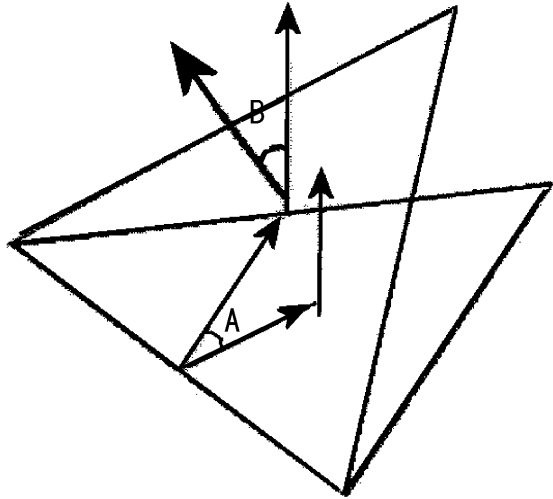
【 図 19 】



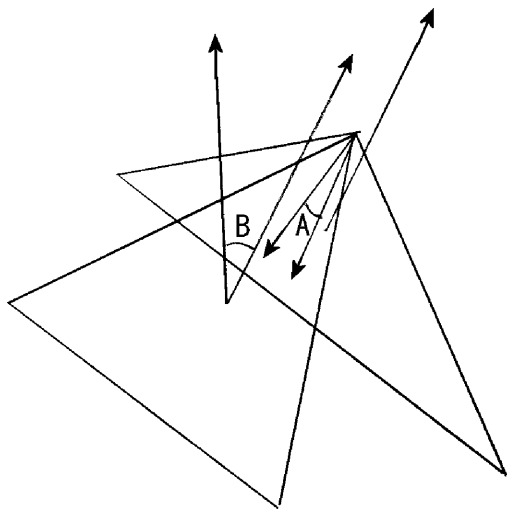
【 図 2 1 】



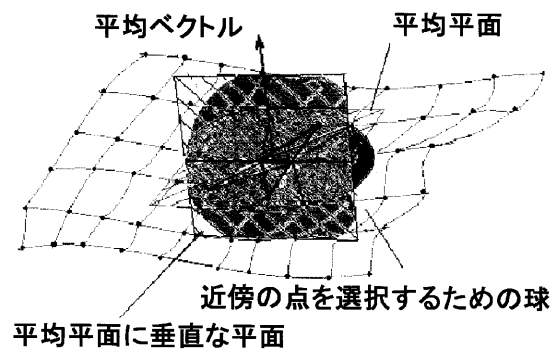
【 図 2 2 】



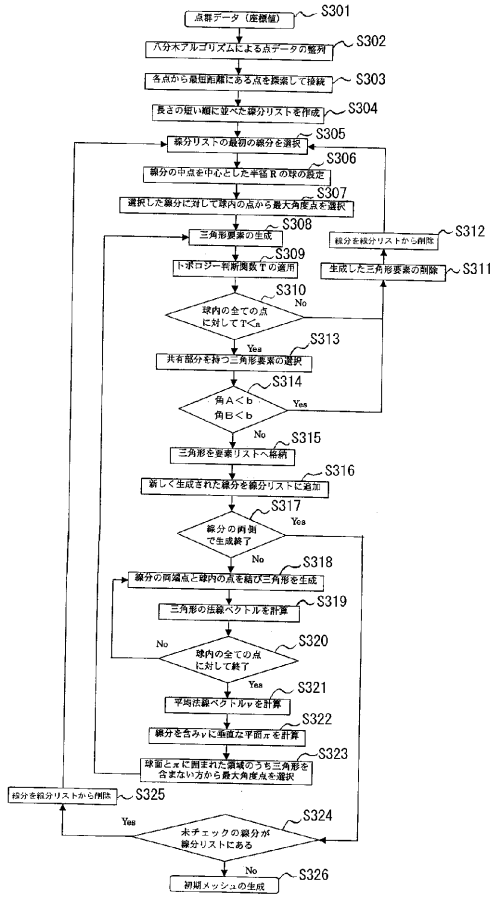
【 図 2 3 】



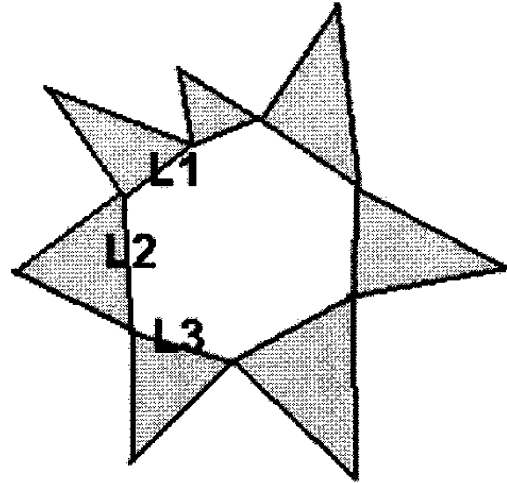
【 図 2 4 】



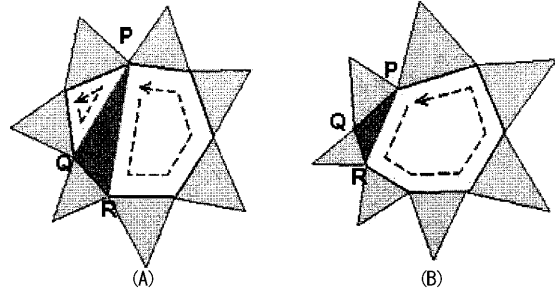
【 図 2 5 】



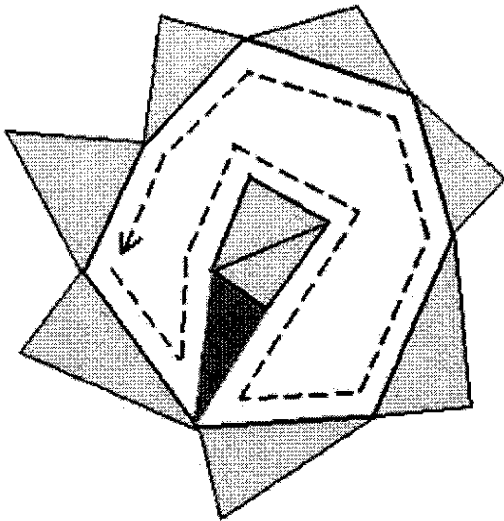
【 図 2 6 】



【 図 2 7 】



【 図 2 8 】



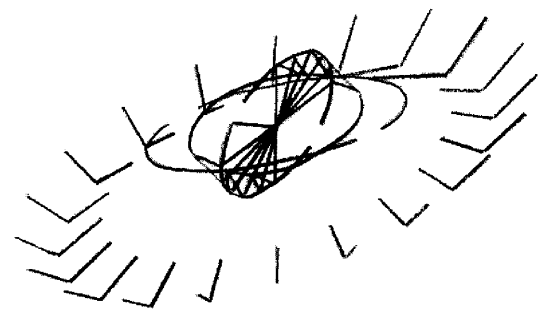
【 図 3 0 】



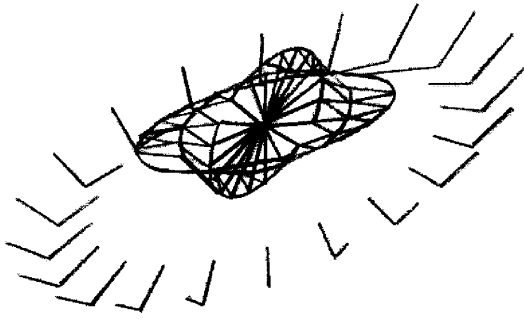
【 図 2 9 】



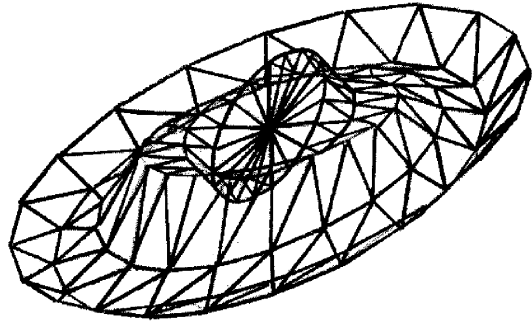
【 図 3 1 】



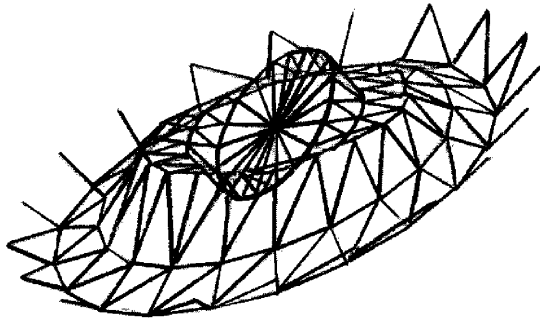
【図 3 2】



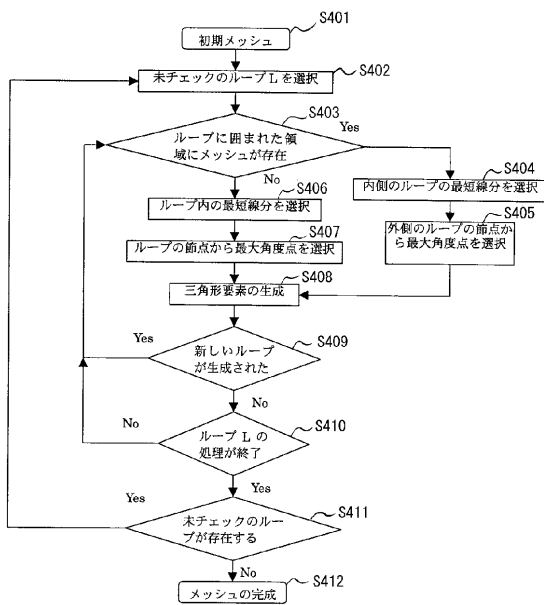
【図 3 4】



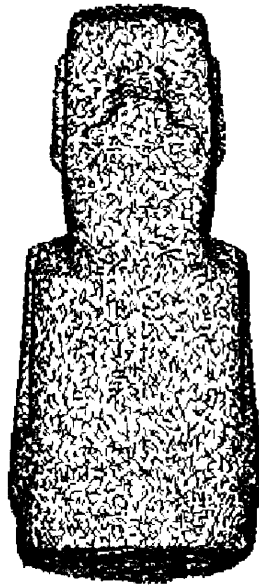
【図 3 3】



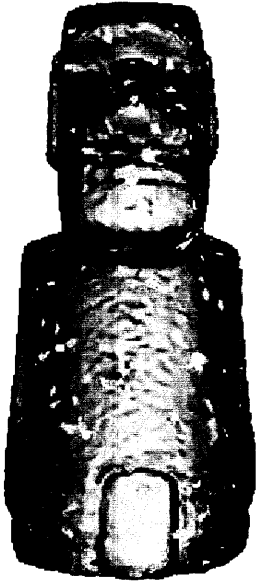
【図 3 5】



【図 3 6】



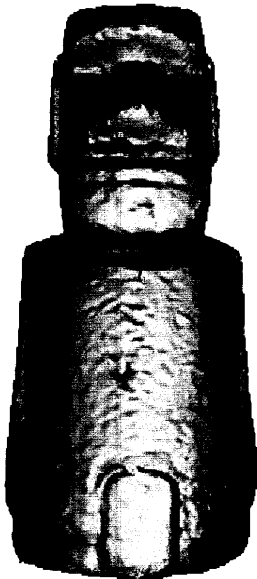
【 37 】



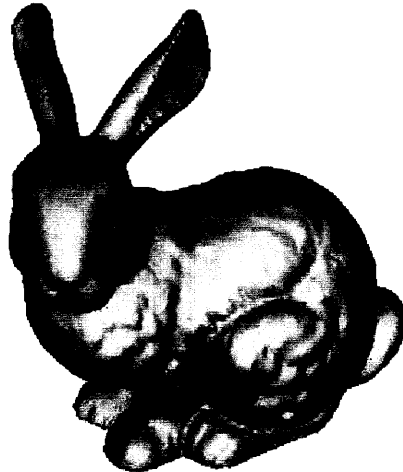
【 38 】



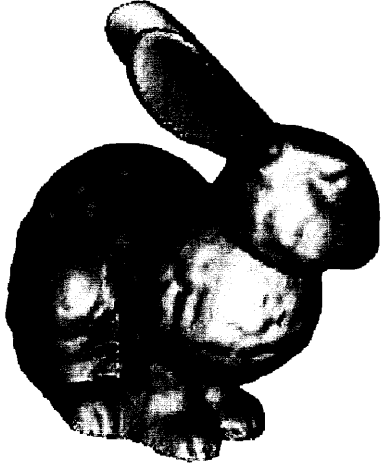
【 39 】



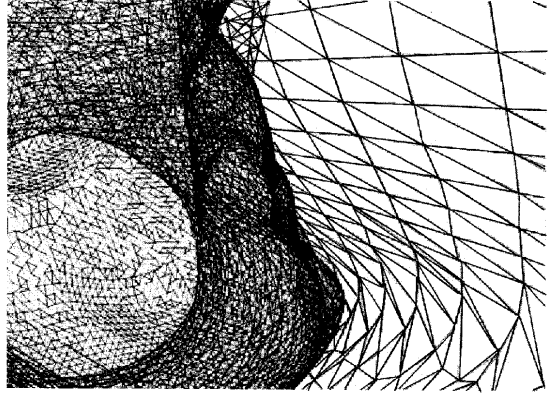
【 40 】



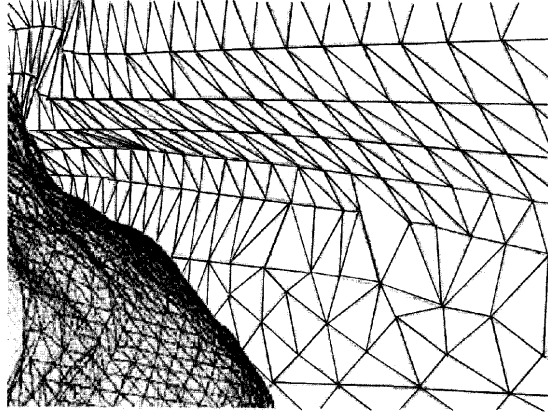
【 図 4 1 】



【 図 4 2 】



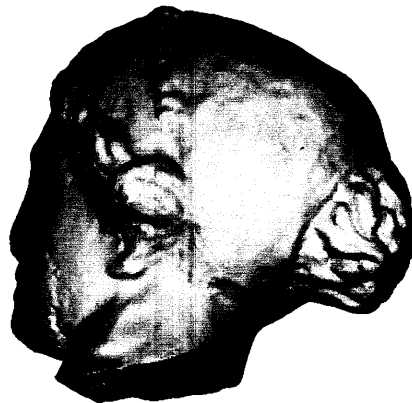
【 図 4 3 】



【 図 4 4 】



【 図 4 5 】



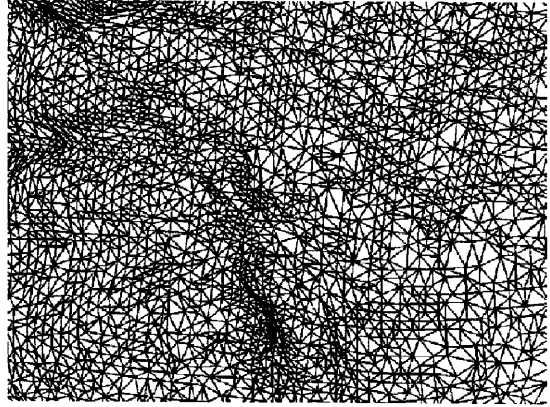
【 図 4 6 】



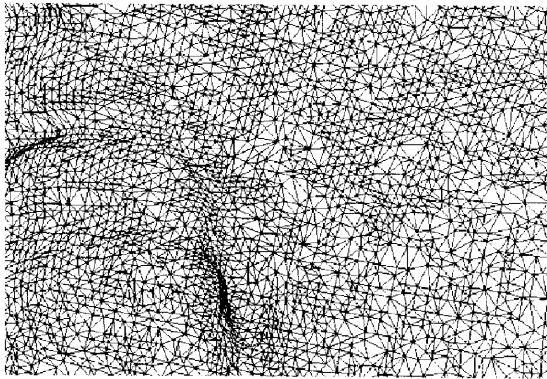
【 図 4 7 】



【 図 4 9 】



【 図 4 8 】



---

フロントページの続き

- (74)代理人 100119194  
弁理士 石井 明夫
- (74)代理人 100128679  
弁理士 星 公弘
- (72)発明者 萩原 一郎  
神奈川県鎌倉市今泉台5 - 10 - 8
- (72)発明者 程 文杰  
東京都大田区東雪谷3 - 30 - 15メゾン石川台401
- (72)発明者 篠田 淳一  
千葉県船橋市本町3 - 12 - 18

審査官 加舎 理紅子

- (56)参考文献 特開平04 - 127378 (JP, A)  
特開平05 - 314214 (JP, A)  
久保寺玄一郎 他, 半空間プリミティブによる対話的CSGモデラ - 光線追跡法におけるCSG表現生成支援システム -, 情報処理学会研究報告, 1992年 2月21日, Vol. 92, No. 13, p. 1 - 8  
河野洋一 他, 有限要素法要素分割の並列化 - ポロノイ分割の並列化 -, 情報処理学会研究報告, 1996年 3月 4日, Vol. 96, No. 22, p. 43 - 48

- (58)調査した分野(Int.Cl., DB名)  
G06F 17/50  
G06T 17/20  
JST7580 (JDream2)  
JSTPlus (JDream2)