

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号
特許第6093704号
(P6093704)

(45) 発行日 平成29年3月8日 (2017.3.8)

(24) 登録日 平成29年2月17日 (2017.2.17)

(51) Int.Cl.

F I

GO 6 T 13/60 (2011.01)

GO 6 T 15/00 (2011.01)

GO 6 T 13/60

GO 6 T 15/00 5 O 1

請求項の数 7 (全 19 頁)

(21) 出願番号	特願2013-533419 (P2013-533419)	(73) 特許権者	000005223
(86) (22) 出願日	平成23年9月15日 (2011.9.15)		富士通株式会社
(86) 国際出願番号	PCT/JP2011/071152		神奈川県川崎市中原区上小田中4丁目1番1号
(87) 国際公開番号	W02013/038548	(74) 代理人	100107766
(87) 国際公開日	平成25年3月21日 (2013.3.21)		弁理士 伊東 忠重
審査請求日	平成26年3月5日 (2014.3.5)	(74) 代理人	100070150
審判番号	不服2016-1062 (P2016-1062/J1)		弁理士 伊東 忠彦
審判請求日	平成28年1月25日 (2016.1.25)	(74) 代理人	100192636
			弁理士 加藤 隆夫
		(72) 発明者	渡邊 正宏
			神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内

最終頁に続く

(54) 【発明の名称】 描画装置、描画方法及び描画プログラム

(57) 【特許請求の範囲】

【請求項 1】

三次元空間における流体の流れを表す流線を、任意の視点位置から見た二次元画像として描画する描画装置において、

所定の開始位置から流線を生成する流線生成部と、
前記視点位置を前記三次元空間内に設定するビュー設定部と、
前記流線生成部が生成した着目対象の流線と、前記着目対象の流線と平行な流線とを含む各流線において、前記着目対象の流線の互いに隣り合う第1の節点と第2の節点とを結ぶ線分における前記第1の節点の位置の勾配と前記第2の節点の位置の勾配との差が予め定められた閾値より大きい場合、前記第1の節点と前記第2の節点の間の前記着目対象の流線に第3の節点を生成することで、前記着目対象の流線の前記第1の節点と、該第1の節点に対応する前記平行な流線の頂点とを結ぶ第1の辺と、前記着目対象の流線の前記第3の節点と、該第3の節点に対応する前記平行な流線の頂点とを結ぶ第2の辺とを含むように生成された多角形面が含まれる複数の多角形面の各々を、1枚ずつ前記流線の方向に連結した帯状面を、前記設定された視点位置に基づき、前記流線に沿うように形成する帯状面形成部と、

前記三次元空間の所定位置に配置された光源に対する前記帯状面の反射光を表す色のテクスチャを、前記複数の多角形面の各々に対応する前記流線の位置における前記流体の物理値に基づいて、前記複数の多角形面の各々にマッピングするテクスチャマッピング部とを有することを特徴とする描画装置。

【請求項 2】

前記物理値は前記流体の流れを表す形状の法線をシミュレートする法線ベクトルであって、

前記テクスチャマッピング部は、

前記法線ベクトルを用いることにより、前記三次元空間の所定位置に配置された光源に対する前記帯状面の反射光を表すテクスチャを導出することで、前記反射光を表すテクスチャを前記複数の多角形面の各々にマッピングすることを特徴とする請求項 1 記載の描画装置。

【請求項 3】

前記帯状面形成部は、

前記設定された視点位置から見た前記複数の多角形面の各々の面積が最大となるよう、又は、前記複数の多角形面の各々の上に存在する特定の点から延ばした法線に対して、前記設定された視点位置から下ろした垂線の長さが最小となるよう、又は、前記複数の多角形面の各々の各頂点と前記設定された視点位置間の距離の総和が最小となるように、前記複数の多角形面の各々の傾きを決定することを特徴とする請求項 1 記載の描画装置。

【請求項 4】

前記帯状面形成部は、

前記複数の多角形面において互いに隣接する第 1 の多角形面と第 2 の多角形面とが共有する辺が、前記流線と交叉するように前記帯状面を形成することを特徴とする請求項 1 記載の描画装置。

【請求項 5】

前記帯状面形成部は、

前記複数の多角形面の各々に対応する前記流線の位置における前記流体の物理値に基づいて、前記複数の多角形面の各々の形状を決定することを特徴とする請求項 1 記載の描画装置。

【請求項 6】

三次元空間における流体の流れを表す流線を、任意の視点位置から見た二次元画像として描画する描画方法において、

コンピュータが、

所定の開始位置から流線を生成し、

前記視点位置を前記三次元空間内に設定し、

生成された着目対象の流線と、前記着目対象の流線と平行な流線とを含む各流線において、前記着目対象の流線の互いに隣り合う第 1 の節点と第 2 の節点とを結ぶ線分における前記第 1 の節点の位置の勾配と前記第 2 の節点の位置の勾配との差が予め定められた閾値より大きい場合、前記第 1 の節点と前記第 2 の節点の間の前記着目対象の流線に第 3 の節点を生成することで、前記着目対象の流線の前記第 1 の節点と、該第 1 の節点に対応する前記平行な流線の頂点とを結ぶ第 1 の辺と、前記着目対象の流線の前記第 3 の節点と、該第 3 の節点に対応する前記平行な流線の頂点とを結ぶ第 2 の辺とを含むように生成された多角形面が含まれる複数の多角形面の各々を、1 枚ずつ前記流線の方向に連結した帯状面を、前記設定された視点位置に基づき、前記流線に沿うように形成し、

前記三次元空間の所定位置に配置された光源に対する前記帯状面の反射光を表す色のテクスチャを、前記複数の多角形面の各々に対応する前記流線の位置における前記流体の物理値に基づいて、前記複数の多角形面の各々にマッピングすることを特徴とする描画方法。

【請求項 7】

三次元空間における流体の流れを表す流線を、任意の視点位置から見た二次元画像として描画する描画プログラムにおいて、

コンピュータに、

所定の開始位置から流線を生成させ、

前記視点位置を前記三次元空間内に設定させ、

生成された着目対象の流線と、前記着目対象の流線と平行な流線とを含む各流線において、前記着目対象の流線の互いに隣り合う第1の節点と第2の節点とを結ぶ線分における前記第1の節点の位置の勾配と前記第2の節点の位置の勾配との差が予め定められた閾値より大きい場合、前記第1の節点と前記第2の節点の間の前記着目対象の流線に第3の節点を生成することで、前記着目対象の流線の前記第1の節点と、該第1の節点に対応する前記平行な流線の頂点とを結ぶ第1の辺と、前記着目対象の流線の前記第3の節点と、該第3の節点に対応する前記平行な流線の頂点とを結ぶ第2の辺とを含むように生成された多角形面が含まれる複数の多角形面の各々を、1枚ずつ前記流線の方向に連結した帯状面を、前記設定された視点位置に基づき、前記流線に沿うように形成させ、

前記三次元空間の所定位置に配置された光源に対する前記帯状面の反射光を表す色のテクスチャを、前記複数の多角形面の各々に対応する前記流線の位置における前記流体の物理値に基づいて、前記複数の多角形面の各々にマッピングさせることを特徴とする描画プログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、描画装置、描画方法及び描画プログラムに関する。

【背景技術】

【0002】

近年、計算機シミュレーションは、計算機の計算能力の向上により、様々な分野で利用されるようになってきた。そして、計算機シミュレーション結果のデータ量は、計算機の計算能力の向上とともに増加傾向にあり、特に高度科学技術計算（HPC: High Performance Computing）の分野においては、従来よりもデータ量が膨大なものとなっている。

【0003】

図1は、計算機シミュレーション結果を可視化した例を示している。図1は、立方体内に存在する流体の流れを示している。流体は、流体を構成している分子が時間と共にその位置を変えている。特定の流体の分子に着目し、その分子が時間の経過によって流体内部を移動する軌跡を流線と呼ぶ。計算機シミュレーションにより、1つ以上の開始点を予め定めておき、その開始点から複数の流線を計算することができる。流線自体は、幅を持たない線であるため、計算機シミュレーションのデータから流線だけを描画しても、人間が把握することが容易ではない。このため、図1に示されるように、流線をチューブ状のオブジェクトとして三次元空間上に配置する。そして、視点位置を定め、三次元空間上に配置したオブジェクトを視点位置から見える二次元グラフィック表示としてレンダリング（描画処理）を行うことにより、流線の可視化が実現できる。

【0004】

上述の流線の可視化を行う際には、様々な視点位置および視点方向からの観察が求められる。この場合、三次元形状のチューブの表面を複数のポリゴンで表現することが知られている。チューブに対して、特定の光源からのシェーディング（陰影処理）を施す。チューブの位置における物理値（例えば、流体の速度、温度等）に応じて色づけ又は模様づけをするためにテクスチャマッピングを行う。そして、定められた視点位置および視点方向から見える二次元画像を形成するために、レンダリングを行うこととなる。計算機シミュレーションのデータ量は、従来よりも膨大な量（テラバイト～ペタバイト単位のデータ量）となる。このため、1回のレンダリングに従来よりも多くの時間を要する。しかも、視点位置および視点方向を変えた場合には、レンダリングを再度行わなければならない。

【0005】

例えば、人物をVR空間において表示しようとする場合、モデルを複数の平面で捉え、視点位置及び視点方向情報に基づき、これらの複数の平面のうち、少なくとも1つの平面を選択し、選択された平面の情報にのみ基づき表示処理する技術が知られている。しかしながら、この技術では、視点位置及び視点方向に対応して、モデルに対して複数の平面を予め用意しておく必要があり、用意すべき平面の数が多くなってしまう。

【 0 0 0 6 】

また、三次元画像データによる物体の周囲に複数の投影面を設定し、各投影面に物体の見える面を平行投影し、各投影面の投影画像から物体表面の色データを持つテクスチャ画像を生成する。そして、これらの投影面毎のテクスチャ画像との対応関係を示す関連情報を付加した三次元形状データを生成する。三次元形状データで表現される物体を任意の視点方向から見た二次元画像データを生成し、そのテクスチャ関連情報に基づいて各投影面のテクスチャ画像を多重的にマッピングして物体を再現する技術が知られている。

【 0 0 0 7 】

また、三次元仮想空間に設定した物体、光源及び視点に基づいて前記空間の情景（シーン）を表す二次元画像を生成するために、物体のモデルデータを視点の座標系に変換し、視点座標系によるモデルデータを投影面に投影し、視点からの着目物体方向等を基準として、各物体のレンダリング方法を切り替える技術が知られている。

【 0 0 0 8 】

図 2 A は、ワイヤフレームで閉じられた多数のポリゴンで表現された流線を可視化したチューブ状のワイヤフレームを示している。また、図 2 B は、流線の可視化処理の流れを示している。上述のように、これらのポリゴンの数は年々増加傾向にある。そして、各空間位置には、物理値が存在している。物理値の例としては、渦度などのスカラー値、速度、応力などのベクトル値、せん断応力などのテンソル値などがある。まず、ステップ 2 1 0 で、シミュレーション結果等のデータを入力する。そして、ステップ 2 2 0 で、必要に応じて、数値データに対しフィルタ処理などの加工を施す。その後、ステップ 2 3 0 で、まず流線を計算する。図 2 A で示したように、流線をチューブ状として表現するために、多数のポリゴンによるチューブ状の形状を生成する。そして、ステップ 2 4 0 でグラフィック処理によるシェーディング等を行った後、レンダリングを行なう。物理値は、ポリゴン上に色などを用いて表現する。

【 0 0 0 9 】

このように、計算機シミュレーションによる物理値は近年増加しており、しかも従来より多数のポリゴンを有するチューブの形状に色とシェーディングを付与して二次元画像をレンダリングする必要がある。このためには、従来より膨大な計算量及び長時間を要する。

【 0 0 1 0 】

加えて、視点を遮る陰面の透過処理を行わない場合や視点を動かさない通常の三次元可視化処理においては、隠面側の視点から見えない形状については、可視化のために必ずしも計算しなくてよい。しかしながら、自由に視点を動かして観察することが必要となる計算機シミュレーションデータ等の可視化の場合には、隠面側の視点から見えない形状についても、可視化のために計算を行っておくことが必要となる。

【 先行技術文献 】

【 特許文献 】

【 0 0 1 1 】

【 特許文献 1 】 特開平 1 0 - 2 2 2 6 9 2 号公報

【 特許文献 2 】 特開平 1 0 - 2 8 3 4 8 7 号公報

【 特許文献 3 】 特開 2 0 0 3 - 1 1 5 0 5 5 号公報

【 発明の概要 】

【 発明が解決しようとする課題 】

【 0 0 1 2 】

1 つの側面では、本発明は、数値シミュレーション結果を表示する際の処理負荷の軽減を図ることを目的とする。

【 課題を解決するための手段 】

【 0 0 1 3 】

本装置の一観点によれば、三次元空間における流体の流れを表す流線を、任意の視点位置から見た二次元画像として描画する描画装置において、所定の開始位置から流線を生成

10

20

30

40

50

する流線生成部と、前記視点位置を前記三次元空間内に設定するビュー設定部と、複数の多角形面を連結した帯状面を、前記設定された視点位置に基づき、前記流線に沿うように形成する帯状面形成部と、前記三次元空間の所定位置に配置された光源に対する前記帯状面の反射光を表すテクスチャを、前記複数の多角形面の各々にマッピングするテクスチャマッピング部を有することを特徴とする描画装置が提供される。

【発明の効果】

【0014】

数値シミュレーション結果を表示する際の処理負荷の軽減を図ることができる。

【図面の簡単な説明】

【0015】

10

【図1】 計算機シミュレーション結果の可視化の例を示す図である。

【図2A】 多数のポリゴンで表現されたチューブによって流線を示す図である。

【図2B】 流線の可視化処理の流れを示す図である。

【図3】 流線を示す図である。

【図4A】 流線に沿って四辺形を連結した帯状面を形成する処理を示す図である。

【図4B】 流線に沿って四辺形を連結した帯状面を形成する処理を示す図である。

【図4C】 流線に沿って四辺形を連結した帯状面を形成する処理を示す図である。

【図4D】 流線に沿って四辺形を連結した帯状面を形成する処理を示す図である。

【図5】 法線マップを示す図である。

【図6】 反射光と物理値に対応するカラーテクスチャのマッピングを示す図である。

20

【図7】 物理値とテクスチャの色とのマッピングを示す図である。

【図8】 物理値とチューブの太さとのマッピングを示す図である。

【図9】 2つの物理値をチューブの太さと色にマッピングした例を示す図である。

【図10A】 視点Iaに対する四辺形の向きを決定を示す図である。

【図10B】 視点Ibに対する四辺形の向きを決定を示す図である。

【図10C】 視点Ibに対する四辺形の向きを決定を示す図である。

【図11A】 一実施形態の処理フローを示す図である。

【図11B】 一実施形態の処理装置を示すブロック図である。

【図12】 一実施形態のハードウェア構成を示す図である。

【図13】 四辺形の傾きを変更した場合の四辺形の修正を示す図である。

30

【図14】 四辺形の傾きを変更した場合の四辺形の修正を示す図である。

【発明を実施するための形態】

【0016】

図3は、三次元空間上に満たされた流体（不図示）の流線L300を示している。一般に、ある瞬間における各点の流体分子の速度ベクトルに沿ってできる曲線を「流線」という。また、同一の開始点から次々と流れる複数の流体分子をある瞬間に結んでできる曲線を「流脈線（流条線）」という。そして、一つの流体分子の移動経路を継続追跡して描かれる曲線を「流跡線」という。定常な流れでは、これら三者は同一となるが、非定常な流れでは異なる線となる。本明細書においては、上記の「流線」、「流脈線（流条線）」及び「流跡線」を、便宜上総称して「流線」の語を用いて説明する。本明細書に記載された複数の実施例は、これら三者のいずれにも適用できることは言うまでもない。

40

【0017】

図3において、流線L300は、各時間ステップのベクトル場を入力データとして、本出願の出願日時点で当業者に知られているRunge-Kutta法、あるいは、Dormand-Prince法により、流線L300を計算することができる。流線L300には、曲率の大きな特徴部分に点（節点）P301乃至P306を配置してもよい。

【実施例】

【0018】

図4A乃至図4Dは、流線に沿って四辺形を連結した帯状面を形成する処理を示している。なお、本実施形態では、四辺形（矩形）を用いているが、四辺形以外の多角形面を用

50

いてもよい。

【 0 0 1 9 】

図 4 A において、三次元空間上に視点 I 及び視線ベクトル V を設定する。この視点 I 及び視線ベクトル V は、ユーザによって指定されてもよい。ユーザからの指定がない場合は、予め定められた視点 I の位置及び視点ベクトル V の値を用いてもよい。また、様々な角度から、流線を観察するために、これらの値は、ユーザにより適宜変更されてもよい。指定された視点 I 及び視線ベクトル V の情報は、記憶部に記憶され、本実施例のシステムによって利用される。

【 0 0 2 0 】

開始位置 P 4 0 1 から終端位置 P e n d まで、流線 L 4 0 0 が示されている。まず、流線 L 4 0 0 の開始位置 P 4 0 1 と、隣の節点 P 4 0 2 を含むように、四辺形 M 4 1 0 を生成する。四辺形 M 4 1 0 の辺 H 4 0 1 上に開始位置 P 4 0 1 が存在し、辺 H 4 0 2 上に節点 P 4 0 2 が存在するようにしてもよい。節点を通る辺 H 4 0 1、H 4 0 2 は、予め定められた一定の長さとしてもよい。あるいは、節点における物理値（例えば、流体の速度、温度）に応じて、辺の長さを決定してもよい。四辺形 M 4 1 0 は、4 つの頂点（P 4 0 1 a、P 4 0 1 b、P 4 0 2 a、P 4 0 2 b）を持つ。四辺形 M 4 1 0 の面の傾きは、視点 I から四辺形 M 4 1 0 が見える面積が最大となるように決定される。同様に、節点 P 4 0 2 と節点 P 4 0 3 を用いて、四辺形 M 4 2 0 を生成する。四辺形 M 4 2 0 は、節点 P 4 0 2 を通る辺 H 4 0 2 と、節点 P 4 0 3 を通る辺 H 4 0 3 を持ち、4 つの頂点（P 4 0 2 a、P 4 0 2 b、P 4 0 3 a、P 4 0 3 b）を有する。そして、四辺形 M 4 1 0 と四辺形 M 4 2 0 は、辺 H 4 0 2 を共有する。なお、各々の四辺形は、必ずしも長方形又は正方形である必要はなく、平行四辺形又は台形となってもよい。あるいは四辺形以外の三角形やその他の多角形でもよい。以上の操作を繰り返し、終端位置 P e n d まで、四辺形を生成する。この操作によって、複数の四辺形により接続された帯状面 Q 4 0 0 が流線 L 4 0 0 に沿って形成される。なお、各四辺形の面の向きをどのように決定するかのアプローチの詳細は後述する。

【 0 0 2 1 】

図 4 B は、着目する流線と平行になるように新たな流線をコピーする操作を示している。この操作は、図 4 C で後述するように、四辺形の数を増やし、帯状面 Q 4 0 0 を滑らかにするための前処理として位置づけられる。四辺形を適切な数だけ生成する処理は、ここで述べる方法以外にもあり得る点に留意すべきである。例えば、図 4 A において、予め節点の数を増やしておき、四辺形を生成してもよい。流線 L 4 0 0 を挟むように、流線 L 4 0 0 に近似した近似流線 L 4 0 0 a 及び L 4 0 0 b を描く。L 4 0 0 a は、四辺形の頂点 P 4 0 1 a、P 4 0 2 a、P 4 0 3 a、...、P e n d a を通るように、流線 L 4 0 0 を平行移動して生成した曲線としてもよい。同様に、四辺形の頂点 P 4 0 1 b、P 4 0 2 b、P 4 0 3 b、...、P e n d b を通る曲線 L 4 0 0 b を生成する。なお、本実施形態では、着目する流線を挟むように 2 つの新たな曲線を生成したが、新たな流線を 1 つだけ生成してもよい。

【 0 0 2 2 】

図 4 C は、辺を追加することにより、四辺形の数を増加させる操作を示している。例えば、辺 H 4 0 1 と辺 H 4 0 2 の間に、辺 H 4 0 1 a が追加されている。同様に、辺 H 4 0 2 と辺 H 4 0 3 の間に、辺 H 4 0 2 a が追加されている。

【 0 0 2 3 】

辺の追加は、例えば以下のように行えばよい。

- (1) 各流線において隣り合う 2 つの節点を結ぶ線分の勾配（勾配ベクトル）を求める。
- (2) 2 つの節点を結ぶ線分勾配の差が、予め定められた閾値 t_h より大きい場合、例えば、その 2 つの節点の中間点に新たな節点を生成する。
- (3) 各流線において生成した新たな節点を結ぶ新たな辺を生成する。
- (4) 新たな辺と、隣接する既存の辺とで構成される四辺形を生成する。

【 0 0 2 4 】

10

20

30

40

50

以上の操作により、例えば、辺 H 4 0 1 と、新たな辺 H 4 0 1 a とを持つ新たな四辺形を形成することができる。同様に辺 H 4 0 1 a と辺 H 4 0 2 とで、新たな四辺形を形成する。この操作を繰り返すことにより、四辺形の数を増加させることができる。なお、上述のように、四辺形の増加の方法は、この実施形態に限定されない。

【 0 0 2 5 】

なお、上記 (2) の処理において、2つの節点の中間点以外に新たな節点を生成しても良い。

【 0 0 2 6 】

図 4 D は、三角形のポリゴンを形成する操作を示している。ポリゴンとは、立体形状を三次元的に表現する場合に用いられる当該立体形状の表面を分割する多角形である。計算のしやすさを考慮した場合、ポリゴンとして三角形が用いられる。本実施形態では、三角形のポリゴンを形成している。例えば、四辺形に一本の対角線 m 4 0 1 を引くことにより、三角形のポリゴン P g 4 0 1 と P g 4 0 2 が形成される。対角線 m 4 0 1 乃至 m 4 0 4 を引くことにより、ポリゴン P g 4 0 1 乃至 P g 4 0 8 が形成される。三角形のポリゴンを形成することにより、汎用的な三次元グラフィクスソフト及びレンダリングハードウェアを用いて処理を行うことが容易となる。なお、この段階で、四辺形の辺のうち、流線と交叉している辺の長さを、流線のその位置での物理値に基づいて伸縮させてもよい。辺の長さを伸縮させることにより、チューブの太さを変化させることができる。辺の長さの伸縮は、適切な他の処理段階で行ってもよいことは言うまでもない。なお、最終的なチューブ形状に見える二次元画像をレンダリングするために、ポリゴンを生成することは必須ではない。

【 0 0 2 7 】

従来技術では、流線 1 本あたりのポリゴン数がより多い。このため、処理を行う計算機の画像処理性能によっては、流線の可視化により多大な時間が必要となる。このため、入力の段階において単純にデータ量を減らす方法が知られている。シミュレーションのデータは既に時間空間で離散化されているが、再度サンプリングして、より疎なデータとする。この方法では、観察者が見たい現象を正確に再現できないというリスクがある。すなわち、この方法は、処理前に情報を削減してしまうため、流線の構造が、データの削減前のものと同じにならない可能性が高い。

【 0 0 2 8 】

これに比して、本実施例では、データ量を削減せずに、流線の 1 本あたりのポリゴン数を大幅に削減できる。

【 0 0 2 9 】

図 5 (A) は、法線マップの模式図を示している。四辺形 5 0 0 は面であるため、このままの状態ではレンダリングを行うと、単純な面として描画されてしまう。この四辺形をチューブ形状に見えるようにするために、四辺形の各領域 c 1 乃至 c 4 に、それぞれ法線ベクトル u 1 乃至 u 4 をマッピングする。この法線ベクトル u 1 乃至 u 4 は、チューブ形状の法線をシミュレートするよう設定される。図 5 (B) は、法線マップ情報をメモリ上に記憶する際のテーブルの例を示している。それぞれの位置に対応づけて法線ベクトルの値が格納される。なお、この実施形態では、4つの位置に法線ベクトルを対応付けたが、更に多くのベクトルを持つ法線マップを用いて細かい対応付けを行ってもよい。また、法線マップの代わりに、チューブ形状の高さを示す高さマップを保持してもよい。

【 0 0 3 0 】

なお、法線マップは、長方形 5 0 0 に対応して保持しておけば、平行四辺形、台形等の四辺形 (矩形) に対する法線マップを計算することができる。たとえば、四辺形 5 0 0 の四隅の頂点を台形の四隅の頂点と対応付けて、モーフィングと同様の手法で法線マップを台形にマッピングすればよい。同様に、三角形や四辺形以外の多角形に対しても、長方形 5 0 0 の法線マップを利用することができる。なお、三角形やその他の多角形用のための法線マップ情報を、ライブラリとして別途記憶しておいてもよいことは言うまでもない。

【 0 0 3 1 】

なお、上述の例では、ポリゴンとして三角形を用いているが、ポリゴンは、三次元形状のモデリングの手法として用いるものであり、法線マップのデータ構造とは別個に扱うことができる。

【 0 0 3 2 】

図 6 (A) は、反射光のテクスチャのマッピングを示している。上述の法線マップを適用して光源 6 5 0 からの光の反射を考慮し、視点 I から視線ベクトル V 方向に見た四辺形 6 0 0 をレンダリングした結果が示されている。例として、適用したテクスチャは単色である。

【 0 0 3 3 】

図 6 (B) は、それぞれの流線の位置における物理値（例えば、速度）を色にマッピングしたテクスチャを用いてレンダリングした結果を示している。図 6 (B) は、白黒の階調表示で描かれているが、実際には、色を変化させたテクスチャをマッピングしている。

【 0 0 3 4 】

このように、流線をチューブ形状によって可視化すると共に、流線の個々の位置における物理値（例えば、温度）を色によって表現することができる。更に、別の物理値（例えば、速度）に応じてチューブ形状の太さを変えることにより、チューブ形状の色と太さにより、二つの物理値（例えば、温度と速度）を同時に表現することができる（図 9 参照）。

【 0 0 3 5 】

図 7 は、物理値とテクスチャの色とのマッピングを示している。例えば物理値として流体の温度を四辺形に対して色のテクスチャとしてマッピングするための関係を示している。温度が低い場合には、青にマッピングされ、温度が高い場合には赤にマッピングされる。温度が中程度であれば黄色にマッピングされる。

【 0 0 3 6 】

図 8 は、物理値とチューブの太さとのマッピングを示している。例えば、流体の速度（流速）が小さければチューブの太さを太くする（すなわち四辺形の辺のうち、流線と交叉している辺の長さを長くする）。逆に、流体の速度が大きければチューブの太さを捕捉する（すなわち四辺形の辺のうち、流線と交叉している辺の長さを短くする）。

【 0 0 3 7 】

図 9 は、図 7 及び図 8 の関係を用いて、帯状面を形成し、テクスチャマッピングを施し、レンダリングを行った結果の例を示している。図 9 において、9 5 1 乃至 9 5 4 は、それぞれ、四辺形を表している。描かれたチューブは、9 0 1 及び 9 0 3 の領域で細くなっており、9 0 2 の領域で太くなっている。このことから、9 0 2 の領域では、他の部分に比べて流速が小さいことが分かる（図 8 参照）。また、図 9 においては、白黒となっているが、9 0 1 及び 9 0 3 の領域では、赤い色を示しており、9 0 2 の領域では黄色い色を示している。したがって、9 0 2 の領域では、他の部分に比べて温度が低いことが分かる。

【 0 0 3 8 】

図 1 0 A 及び図 1 0 B は、視点を視点 I a から視点 I b に移動させた場合に、四辺形 M 1 0 0 の面の向きを決定する処理を示している。

【 0 0 3 9 】

まず、図 1 0 A において、視点は、視点 I a に位置している。四辺形 M 1 0 0 において、2つの対向する辺は、流線 L 1 0 0 の節点 P 1 0 1 と P 1 0 2 において接している。この場合、四辺形 M 1 0 0 は、方向 D 1 2 0 又は方向 D 1 3 0 に、面の傾きを変化させることができる。四辺形 M 1 0 0 の面の傾きは、視点が視点 I a の位置にある場合には、例えば以下のように決定することができる。

【 0 0 4 0 】

[視点が視点 I a の位置にある場合の四辺形 M 1 0 0 の向きの決定]

(1) 四辺形 M 1 0 0 上に単位面積 T 1 5 0 を設定する。そして、視点 I a から見て T 1 5 0 の面積が最大に見えるように、四辺形 M 1 0 0 の傾きを決定する。又は、

10

20

30

40

50

(2) 四辺形 M 1 0 0 上の任意の点 S 1 6 0 から四辺形 M 1 0 0 に対する法線ベクトル V_s を延ばす。そして、視点 I a から法線ベクトル V_s に垂線を下ろし、その垂線の長さ L_a が最小となるように、四辺形 M 1 0 0 の傾きを決定する。

(3) 四辺形 M 1 0 0 の四隅の点から視点までの距離の総和が最小となるように、四辺形 M 1 0 0 の傾きを決定する。

【 0 0 4 1 】

以上が、四辺形 M 1 0 0 の面の傾きを決定する処理の例である。上記の決定処理以外にも、当業者であれば、M 1 0 0 の面の傾きを決定する手法は見いだせるであろう。

【 0 0 4 2 】

図 1 0 B は、視点が視点 I a から視点 I b に移動した場合の、四辺形 M 1 0 0 の面の傾きを決定する処理を示している。

10

【 0 0 4 3 】

上述のように、視点 I b から見て、四辺形 M 1 0 0 上の単位面積 T 1 5 0 が最大に見えるように、四辺形 M 1 0 0 の面の傾きを決定すればよい。又は、四辺形 M 1 0 0 上の任意の点 S 1 6 0 から四辺形 M 1 0 0 に対する法線ベクトル V_s を延ばす。そして、視点 I b から法線ベクトル V_s に垂線を下ろし、その垂線の長さ L_b が最小となるように、四辺形 M 1 0 0 の傾きを決定すればよい。

【 0 0 4 4 】

図 1 0 C は、視点が視点 I a から視点 I b に移動した場合の、四辺形 M 1 0 0 の面の傾きを決定する別の処理を示している。視点 I b と、四辺形 M 1 0 0 の各頂点 K 1 8 1、K 1 8 2、K 1 8 3、及び K 1 8 4 とを結んだ直線を、それぞれ N 1 8 1、N 1 8 2、N 1 8 3、及び N 1 8 4 とする。そして、直線 N 1 8 1、N 1 8 2、N 1 8 3、及び N 1 8 4 の長さの総和が最も小さくなるように、四辺形 M 1 0 0 の傾きを決定すればよい。

20

【 0 0 4 5 】

以上の操作を他の四辺形の面の傾きの決定にも行えば、帯状面の全体の傾きが変更される。その後、テクスチャマッピングを施し、レンダリングを行えば、新たな視点 I b から見たチューブ状の流線の二次元画像が得られる。

【 0 0 4 6 】

このように、帯状面を生成し、テクスチャマッピングを利用して、帯状面をチューブ状に見えるようレンダリングする。三次元空間上にチューブを実際に生成するよりも、帯状面を利用する方が、ポリゴン数を従来よりも減少させることができる。したがって、流線を可視化する処理を従来よりも短縮することが可能である。また、視点を変えた場合にも、上述のように四辺形の面の向きを変更し、テクスチャマッピングを施し、レンダリングを行えばよい。上述のように、少ないポリゴン数のため、視線を変更しても従来よりも高速に二次元画像が得られる。

30

【 0 0 4 7 】

なお、四辺形の面の傾きをより高速に決定するには、更に視線ベクトル及び視野を考慮してもよい(図示せず)。すなわち、視線ベクトルと視野によって、最終的にレンダリングされるべき三次元モデルの部分が特定される。この特定された三次元モデル部分だけに着目して、四辺形の傾きの変更、テクスチャマッピング、レンダリング等の処理を限定して行えばよい。

40

【 0 0 4 8 】

図 1 1 A は、一実施形態の処理フローを示している。以下に示す処理フローは、矛盾のない限り、順番を入れ替えてもよい。そして、また、プログラムに係る請求項及び、方法に係る請求項における、各ステップについても、矛盾のない限り順番を入れ替えてもよいことは言うまでもない。そして、請求項内のステップの順番を入れ替えた場合でも、その請求項の技術的範囲に包含される。また、既に記憶部にデータが記憶されており、改めてユーザからの指示を受け付ける必要がない場合がある点に留意すべきである。加えて、ユーザからの指示がない場合には、デフォルト値を用いて処理を行ってもよい。

【 0 0 4 9 】

50

ステップ 1 1 0 2 で、まず、三次元空間上の視点位置、視線方向、流線開始位置の指示をユーザから受け取る。なお、これらのデータは、予め記憶装置に記憶しておいてもよい。

【 0 0 5 0 】

ステップ 1 1 0 4 で、シミュレーションなどによって得られたベクトル場のデータを読み込む。

【 0 0 5 1 】

ステップ 1 1 0 6 で、記憶部に記憶された流線の開始位置を読み込む。

【 0 0 5 2 】

ステップ 1 1 0 8 で、流線を生成する。

10

【 0 0 5 3 】

ステップ 1 1 1 0 で、記憶部に記憶された視点位置を読み込む。

【 0 0 5 4 】

ステップ 1 1 1 2 で、流線に沿って、帯状面を生成する。ここでは、まず、複数の四辺形を生成する。次に、流線をコピーして新たな流線を生成する。そして、四辺形の両端の辺において、流線に沿った勾配ベクトルの変化量が、予め定められた閾値 t_h よりも大きい場合には、例えば中間点において、四辺形を分割する。なお、既に述べたように、流線をコピーせずに、流線における節点を適切に増加させ、隣接する節点間によって四辺形を形成させてもよい。なお、例えば、節点における物理値（例えば、速度、温度等）に基づいて、節点を含む四辺形の辺の長さを、予め定めた関数を用いて決定してもよい。

20

【 0 0 5 5 】

ステップ 1 1 1 4 で、四辺形に対角線を一本引いて、2つの三角形を生成する。この処理を、全ての四辺形に行う。この処理によって、帯状面は、三角形のポリゴンで表現されることとなる。なお、ポリゴンは、必ずしも三角形である必要はないため、四辺形のままでポリゴンとして扱ってもよい。なお、三角形のポリゴンを用いる場合には、三次元モデルの処理が容易化される。

【 0 0 5 6 】

ステップ 1 1 1 6 で、法線マップのデータを読み込む。法線マップは、長方形に対する法線マップのデータを記憶させておいてもよい。実際の四辺形がひし形、台形である場合には、長方形の法線マップをひし形、台形に適用することができる。例えば、長方形の各頂点を、台形の各頂点と対応させ、変形量を比例配分して台形の法線マップを計算すればよい。

30

【 0 0 5 7 】

ステップ 1 1 1 8 で、法線マップを利用して、反射を表現するテクスチャを生成する。また、流線の各位置における物理値（例えば、速度、温度等）に対応付けて色を加えたテクスチャを生成してもよい。

【 0 0 5 8 】

ステップ 1 1 2 0 において、未処理の流線があるか否かがチェックされる。未処理の流線がある場合には、ステップ 1 1 0 6 に戻る。なお、流線の数、流線の開始位置の個数と一致する。そして、未処理の流線がない場合には、全ての流線に対する帯状面の作成及びテクスチャマッピングが終了したこととなる。この場合には、ステップ 1 1 3 0 に進む。

40

【 0 0 5 9 】

ステップ 1 1 3 0 で、視点位置から、視線方向に見たチューブ状の流線が、レンダリングされる。レンダリングされたデータは、表示装置に表示される。そして、処理は「A」を経由してステップ 1 1 4 2 に進む。

【 0 0 6 0 】

ステップ 1 1 4 2 において、視点を変更するユーザからの指示があるか否かがチェックされる。ユーザは、視点位置を変更して、流線の状況を観察したい場合があるからである。このチェックで、視点位置を変更しない場合には、ステップ 1 1 6 2 に進む。

50

【 0 0 6 1 】

ステップ 1 1 3 2 で、視線方向を変更するユーザからの指示があるか否かがチェックされる。このチェックで視線方向の変更が指示されていることが分かった場合には、ステップ 1 1 6 4 に進む。

【 0 0 6 2 】

ステップ 1 1 6 4 では、ユーザから視線方向の指示を受け取る。この場合には、視点位置は、変更されていないため、視界の変更の指示がなされたことになる。そして、この場合には、ステップ 1 1 3 0 において指示された視点及び視線方向に対する二次元画像をレンダリングすればよい。

【 0 0 6 3 】

ステップ 1 1 4 2 のチェックで、Y E S である場合には、ステップ 1 1 4 4 に進む。

【 0 0 6 4 】

ステップ 1 1 4 4 では、視点位置、視線方向の変更指示をユーザから受け取り、記憶部に記憶する。この場合、視線方向については、変更の指示がなく、視点位置のみが変更される場合もある。

【 0 0 6 5 】

ステップ 1 1 4 6 で、受け付けた視点位置を読み込む。

【 0 0 6 6 】

ステップ 1 1 4 8 で、視点位置に基づいて、帯状面の各々の四辺形の傾きを修正する。

【 0 0 6 7 】

ステップ 1 1 5 0 で、ポリゴンの生成が行われる。この処理は、ステップ 1 1 1 4 と同様である。

【 0 0 6 8 】

ステップ 1 1 5 2 で、法線マップデータが読み込まれる。この処理は、ステップ 1 1 1 6 と同様である。

【 0 0 6 9 】

ステップ 1 1 5 4 で、法線マップを利用して、反射を表現するテクスチャを生成する。また、流線の各位置における物理値（例えば、速度、温度等）に対応付けて色を加えたテクスチャを生成してもよい。この処理は、ステップ 1 1 1 8 と同様である。

【 0 0 7 0 】

ステップ 1 1 5 6 で、未処理の流線があるか否かがチェックされる。未処理の流線がある場合には、ステップ 1 1 4 6 に戻る。そして、未処理の流線がない場合には、全ての流線に対する帯状面の各四辺形の傾きの修正及びテクスチャマッピングが終了したこととなる。この場合には、ステップ 1 1 3 0 に進み、レンダリングが行われる。

【 0 0 7 1 】

ステップ 1 1 6 2 において、視線方向を変更しない場合（N O ）には、ユーザは、視点位置及び視線方向の変更を望んでいないため、更なる処理は不要であるため、一連の処理が終了する。

【 0 0 7 2 】

図 1 1 B は、一実施形態の処理装置のブロック図を示す。まず、データ入力部 1 1 7 2 から流体の流れのシミュレーションデータ等のベクトルデータを読み込み、記憶部 1 1 9 2 に記憶する。

【 0 0 7 3 】

ビュー設定部 1 1 7 4 では、ユーザからの指示を受け付ける。例えば、視点位置、視線方向、焦点距離、並びに、三次元空間上での光源の位置等を受付け、記憶部に記憶する。

【 0 0 7 4 】

開始位置設定部 1 1 7 6 では、流体の流れを示す流線の開始位置に関するユーザの指定を受け付ける。このデータは、記憶部に記憶される。

【 0 0 7 5 】

色、チューブ形状設定部 1 1 7 8 では、チューブ形状の色をどのように設定するかに関

10

20

30

40

50

するユーザからの指定を受け付ける。例えば、流線上の各位置での流体の温度をチューブの色にマッピングする場合には、例えば図7に示したようなマッピング関係を設定する。また、チューブの太さを流体の物理値と関連付けることができる。この場合には、図8に示したように、例えば、流体の流速をチューブの太さに関連付けることができる。なお、チューブの太さは、帯状面を形成する各四辺形の辺の幅を伸縮させることによりコントロールできる。

【0076】

流線生成部1180は、開始位置設定部1176から受け付けられて記憶部に記憶された流線の開始位置と、シミュレーションデータとを用いて、流線を生成する。複数の開始位置が設定されていれば、複数の流線が生成される。

10

【0077】

帯状面形成部1182は、帯状面を形成する。面形状決定手段1184において、流線に沿って複数の四辺形が決定される。面傾き決定部1183において、この複数の四辺形の傾きが決定される。この処理を経て、帯状面が形成される。また、帯状面を三角形のポリゴンに分割する。なお、面傾き決定部1183は、視点の位置等が変更された場合にも、面の傾きを変更するために利用される。

【0078】

テクスチャマッピング部1185は、複数の四辺形の各々にテクスチャをマッピングする。色適用手段1186は上述のように、物理値と色とのマッピングから、テクスチャに色を適用する。法線マップ適用手段1187は、法線マップ1193を記憶部1192から読み出す。そして、光源の位置と法線マップの法線ベクトルを四辺形の傾きに適用し、反射をシミュレートしたテクスチャを四辺形にマッピングする。

20

【0079】

レンダリング部1189は、二次元平面に流線を可視化したチューブを描画する。

【0080】

表示部1190は、レンダリングされた画像を表示する。

【0081】

図12は、一実施形態のハードウェア構成1200を示している。ハードウェアには、CPU1202、メモリ1204、動的メモリ1206、媒体1209を読み書きするドライブ1208、通信インターフェース1210、表示装置1213等に画像信号を送るビデオインターフェース1212、I/Oインターフェース1214、I/Oとしてのキーボード1216、マウス1218、プリンタ1220が含まれる。

30

【0082】

CPU1202は、メモリ1204や動的メモリ1206に格納されている上記説明した種々の機能を実現するプログラムを実行する。

【0083】

また、プログラムに係る本発明は、機械読み取り可能な記録媒体に格納されてもよい。機械読み取り可能な記録媒体には、磁気記録媒体、光ディスク、光磁気記録媒体、半導体メモリなどがある。磁気記録媒体には、HDD、フレキシブルディスク(FD)、磁気テープ(MT)などがある。光ディスクには、DVD(Digital Versatile Disc)、DVD-RAM、CD-ROM(Compact Disc-Read Only Memory)、CD-R(Recordable)/RW(Rewritable)などがある。また、光磁気記録媒体には、MO(Magneto-Optical disk)などがある。

40

【0084】

図13は、四辺形の傾きを変更した場合の四辺形の修正を示す図である。図13(a)は、流線L1300に対し、四辺形M1310及びM1320が、節点P1320を通る辺H1350を共有している。視点(図示せず)の位置を変更した場合、これらの四辺形の傾きを変更する処理が行われ、図13(b)のように、二つの四辺形がM1311及びM1321のように傾きを変えたと仮定する。この場合、二つの四辺形M1311及びM

50

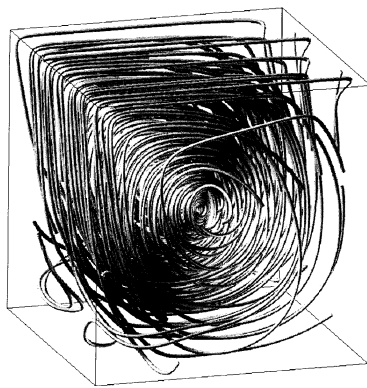
1 3 2 1 は、辺を共有しなくなってしまう場合がある。そのため、節点 P 1 3 0 2 を通り四辺形 M 1 3 1 1 及び M 1 3 2 1 のそれぞれの面が交わる直線 1 3 8 0 を求める。図 1 3 (c) に示すように、先に求めた直線 1 3 8 0 を共有する辺 H 1 3 5 2 として、四辺形 M 1 3 1 2 及び M 2 3 2 2 に示すように修正すればよい。この処理を他の四辺形と共有する辺に対しても行えばよい。

【 0 0 8 5 】

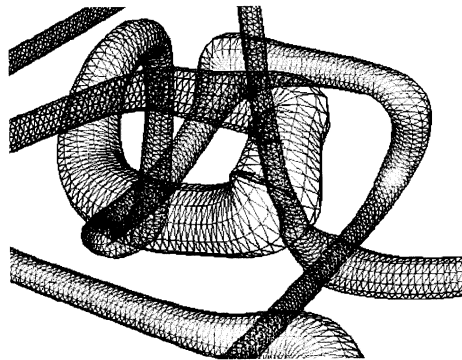
図 1 4 は、四辺形の傾きを変更した場合の四辺形の別の修正を示す図である。図 1 4 (a) は、四辺形 M 1 4 1 0、M 1 4 2 0、及び M 1 4 3 0 が、流線 L 1 4 0 0 に沿って帯状面を形成している。この 3 つの四辺形は、それぞれ隣り合う四辺形と辺を共有している。図 1 4 (b) は、視点 (図示せず) の位置が変更されたために、1 つ置きに四辺形の傾きを変更させた状態を示している。すなわち、四辺形 M 1 4 3 1 及び M 1 4 2 1 が四辺形の傾きを変えている。そして、この状態で、両四辺形の頂点 K 1 4 9 1 と K 1 4 9 4 とを結ぶ。同様に両四辺形の頂点 K 1 4 9 2 と K 1 4 9 3 とを結ぶ。図 1 4 (c) は、その結果、新たな四辺形 M 1 4 1 1 が形成された様子を示している。この処理により、簡便に四辺形の傾きを変えることができる。なお、新たに作成された四辺形 M 1 4 1 1 は、厳密に平面にならず、捻れた面となる場合があるが、平面に近似して (例えば、対角線で 2 分して、2 つの三角形ポリゴンに置き換えて) 後のレンダリング処理等を行ってもよい。

10

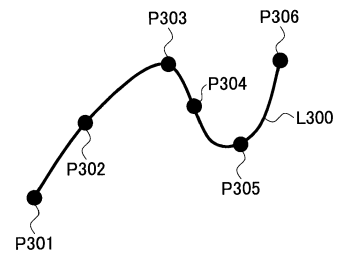
【 図 1 】



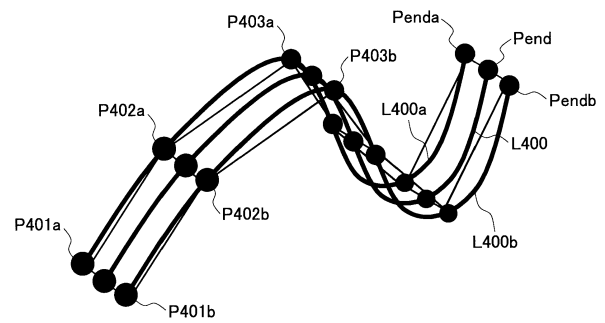
【 図 2 A 】



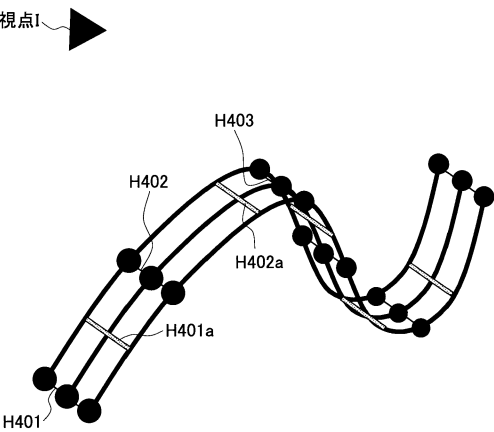
【 図 3 】



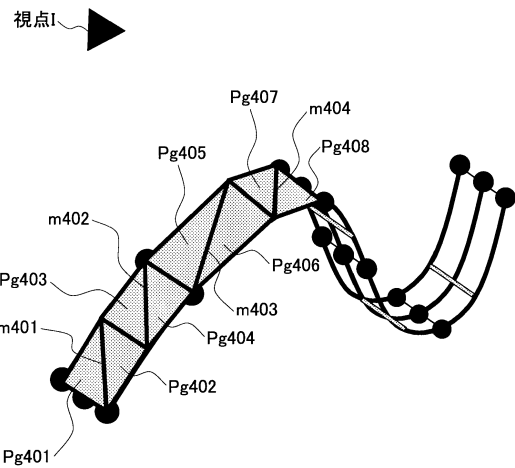
【 図 4 B 】



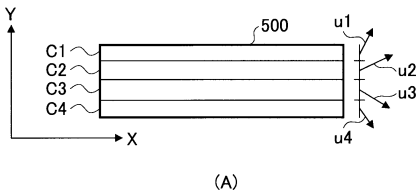
【図 4 C】



【図 4 D】



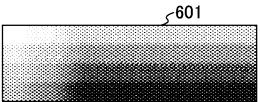
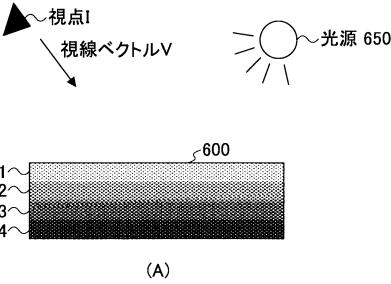
【図 5】



位置	法線ベクトル(X, Y, Z)
c1	$u1 = (0, 6, 4)$
c2	$u2 = (0, 2, 8)$
c3	$u3 = (0, -2, 8)$
c4	$u4 = (0, -6, 4)$

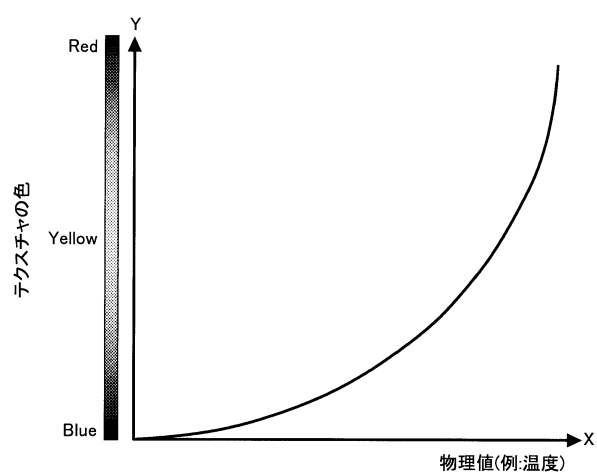
(B)

【図 6】

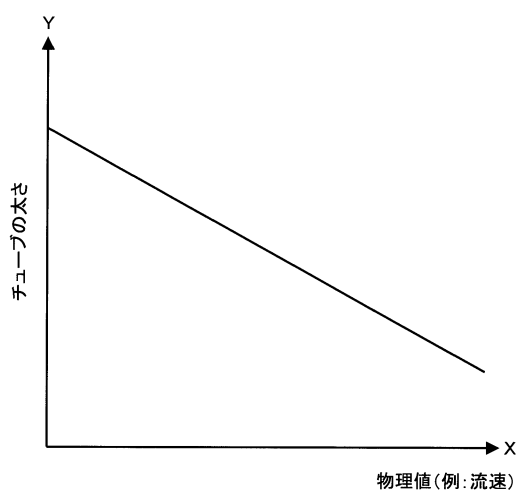


(B)

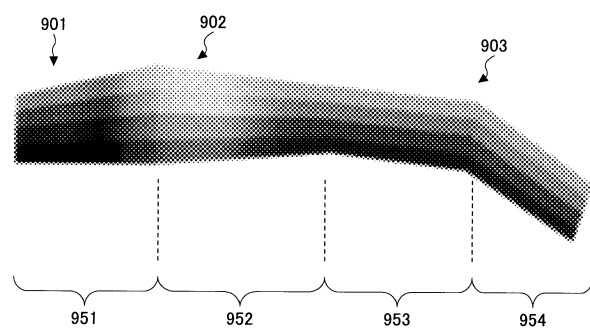
【圖 7】



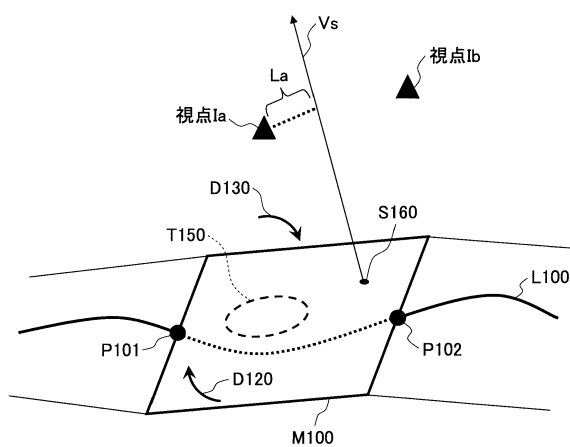
【圖 8】



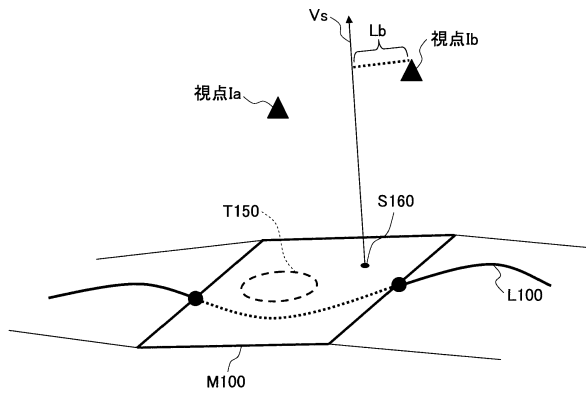
【圖 9】



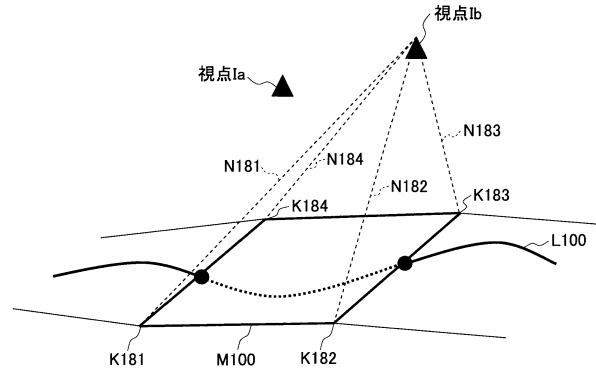
【 図 1 0 A 】



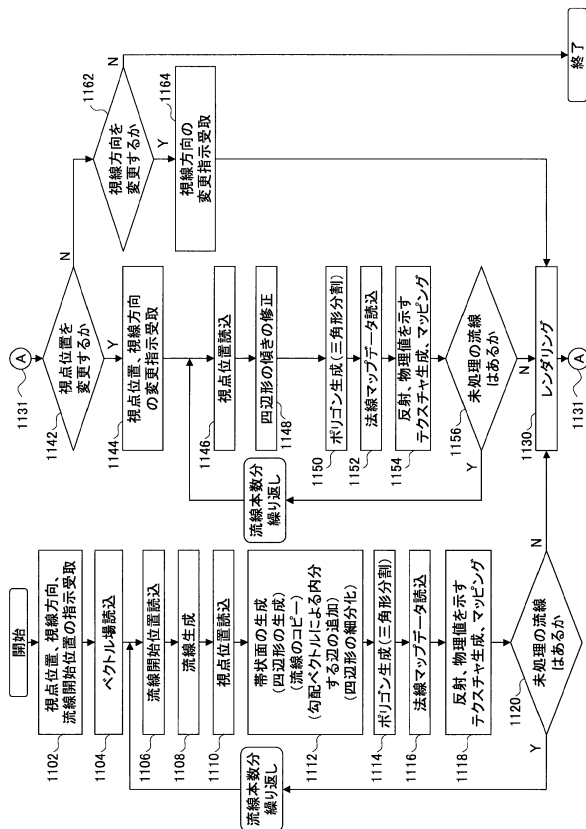
【図10B】



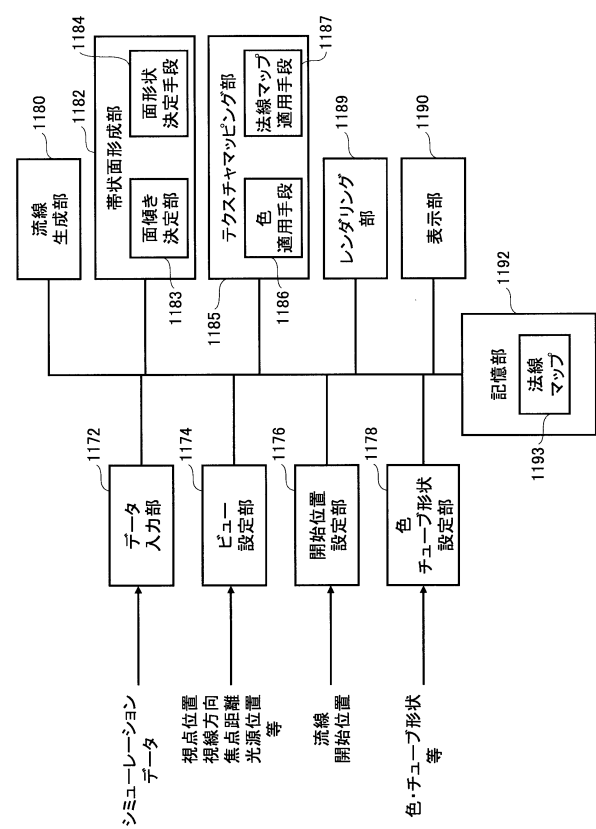
【図10C】



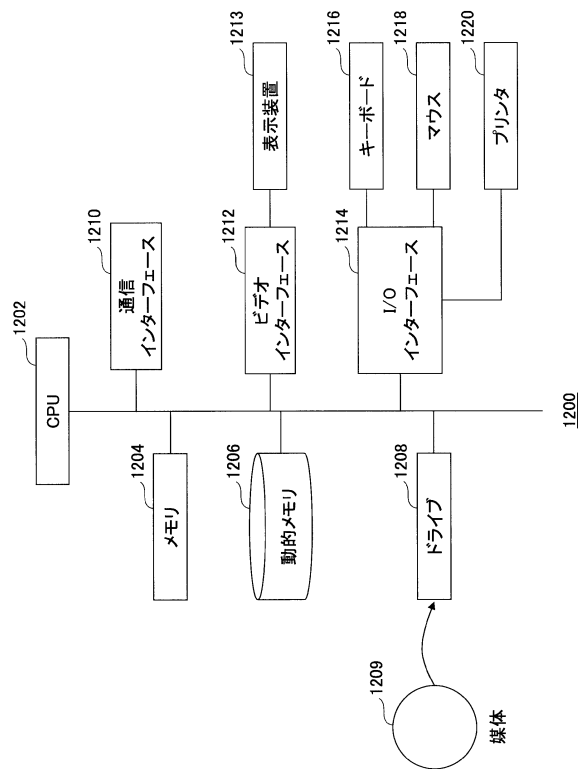
【図11A】



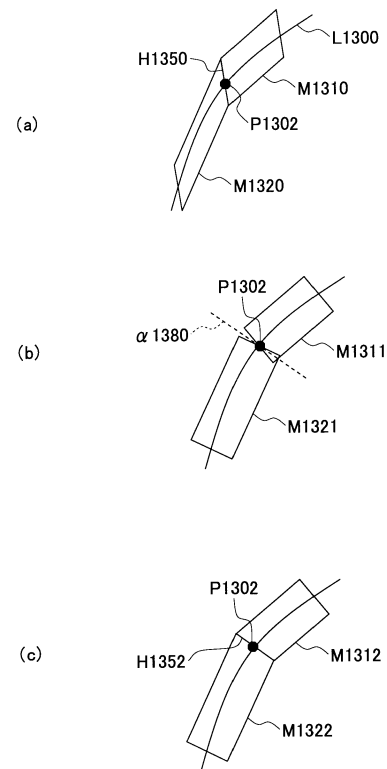
【図11B】



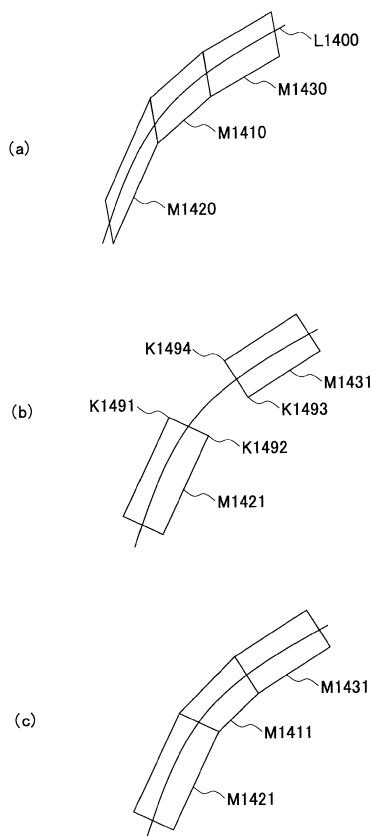
【図 1 2】



【図 1 3】



【図 1 4】



フロントページの続き

(72)発明者 中川 真智子
神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内

合議体

審判長 清水 正一

審判官 渡邊 聡

審判官 小池 正彦

(56)参考文献 特開2000-242631(JP,A)
特開2000-197769(JP,A)
特開2007-226572(JP,A)
国際公開第99/60526(WO,A1)
原田隆宏, 外3名, “粒子ベースシミュレーションの並列化”, 情報処理学会論文誌, 社団法人
情報処理学会, 2007年11月15日, 第48巻, 第11号, p. 3557-3567

(58)調査した分野(Int.Cl., DB名)

G06T13/60

G06T15/00