

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第3619063号
(P3619063)

(45) 発行日 平成17年2月9日(2005.2.9)

(24) 登録日 平成16年11月19日(2004.11.19)

(51) Int.CI.⁷

F 1

GO6T 17/40

GO6T 17/40

F

請求項の数 12 (全 27 頁)

(21) 出願番号 特願平11-194802
 (22) 出願日 平成11年7月8日(1999.7.8)
 (65) 公開番号 特開2001-22344(P2001-22344A)
 (43) 公開日 平成13年1月26日(2001.1.26)
 審査請求日 平成13年2月13日(2001.2.13)

(73) 特許権者 000001007
 キヤノン株式会社
 東京都大田区下丸子3丁目30番2号
 (74) 代理人 100076428
 弁理士 大塚 康徳
 (74) 代理人 100112508
 弁理士 高柳 司郎
 (74) 代理人 100115071
 弁理士 大塚 康弘
 (74) 代理人 100116894
 弁理士 木村 秀二
 (72) 発明者 内山 晋二
 横浜市西区花咲町6丁目145番地 横浜
 花咲ビル 株式会社エム・アール・システム研究所内

最終頁に続く

(54) 【発明の名称】立体画像処理装置、その方法、立体視パラメータ設定装置、その方法そしてコンピュータプログラム記憶媒体

(57) 【特許請求の範囲】

【請求項1】

任意の物体の左目用画像と右目用画像とを表示して前記物体の立体視を得る立体画像処理装置において、

入力画像の奥行き方向について、観察者の視点位置に最も近い、前記物体上の物体点を検出する検出手段と、

前記物体点が二次元描画空間に描画された際に現れる左右間のズレ距離を設定するズレ距離設定手段と、

前記左目用画像と右目用画像のいずれか一方における前記物体点の画像位置と、前記ズレ距離とに基づいて、基線長の値を決定する決定手段と、

決定された基線長の値を立体視パラメータとして用いて、前記左目用画像と右目用画像を生成する生成手段とを具備することを特徴とする立体画像処理装置。

【請求項2】

前記決定手段は、

前記物体点の前記二次元描画空間上の座標から、観察者の基線長方向に、前記ズレ距離だけ離したズレ点の位置を演算するズレ演算手段と、

前記ズレ点の三次元空間における座標を演算する座標変換手段と、

前記ズレ点の三次元座標と前記物体点との距離に基づいて前記基線長の値を決定する基線長決定手段を具備することを特徴とする請求項1に記載の立体画像処理装置。

【請求項3】

所定の条件を満足すべき前記所定の物体点が前記二次元描画空間において高速に探索されるように、前記描画空間における複数の探索位置が前もって離散的分布を有するよう設定されていることを特徴とする請求項2に記載の立体画像処理装置。

【請求項4】

手動で立体視パラメータの値を設定するユーザインタフェース手段を更に具備することを特徴とする請求項1乃至3のいずれかに記載の立体画像処理装置。

【請求項5】

前記ユーザインタフェース手段と前記決定手段との競合の調停をとる手段を更に具備することを特徴とする請求項4に記載の立体画像処理装置。

【請求項6】

10

前記決定手段は、さらに、

入力画像の仮想空間の奥行き方向の広がりを求め、

観察者の左右の視点位置から見える、この仮想空間の奥行き方向の広がりの中間位置の方向に基づいて輻輳角の値を決定し、

前記生成手段は前記決定手段で決定された輻輳角の値を前記立体視パラメータの一つとして用いることを特徴とする請求項1乃至5のいずれかに記載の立体画像処理装置。

【請求項7】

前記ズレ距離設定手段は、

立体画像が表示される表示装置の画素サイズと、この表示装置の表示画面から観察者までの距離とに基づいて前もって前記ズレ距離を決定することを特徴とする請求項1に記載の立体画像処理装置。

20

【請求項8】

前記物体点は、

前記左目用画像と右目用画像のいずれか一方の画像データに、観察者の視点からの前記二次元描画空間への透視変換を施し、透視変換された画像データのなかから最も小さな奥行き値を有する画素に決定されることを特徴とする請求項1に記載の立体画像処理装置。

【請求項9】

請求項1乃至8のいずれかに記載の立体画像処理装置を具備する立体画像表示装置。

【請求項10】

任意の物体の左目用画像と右目用画像とを表示して前記物体の立体視を得る立体画像処理方法において、

30

入力画像の奥行き方向について、観察者の視点位置に最も近い、前記物体上の物体点を検出する検出工程と、

前記物体点が二次元描画空間に描画された際に現れる左右間のズレ距離を設定するズレ距離設定工程と、

前記左目用画像と右目用画像のいずれか一方における前記物体点の画像位置と、前記ズレ距離とに基づいて、基線長の値を決定する決定工程と、

決定された基線長の値を立体視パラメータとして用いて、前記左目用画像と右目用画像を生成する生成工程とを具備することを特徴とする立体画像処理方法。

【請求項11】

40

前記決定工程は、

前記物体点の前記二次元描画空間上の座標から、観察者の基線長方向に、前記ズレ距離だけ離間したズレ点の位置を演算するズレ演算工程と、

前記ズレ点の三次元空間における座標を演算する座標変換工程と、

前記ズレ点の三次元座標と前記物体点との距離に基づいて前記基線長の値を決定する基線長決定工程を具備することを特徴とする請求項10に記載の立体画像処理方法。

【請求項12】

任意の物体の左目用画像と右目用画像とを表示して前記物体の立体視を得るためにコンピュータ上で実行される立体画像処理プログラムを記憶するコンピュータプログラム記憶媒体であって、請求項10または11に記載の立体画像処理方法を実現するためのプログラ

50

ムコードを記憶するコンピュータプログラム記憶媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、例えばVRML (Virtual Reality Modeling Language) 等の三次元の仮想空間を扱う言語により記述された空間データを立体視可能な画像データに変換する立体画像処理装置、立体視パラメータ設定装置、立体画像表示装置、更にはそれらの方法に関する。

【0002】

【従来の技術】

近年の計算機の能力向上と共に、3次元コンピュータグラフィクス(CG)が急速に一般化しつつある。それと相まって、インターネット上のWWW(World Wide Web)の急速な普及と共に、3次元空間をWWW上で取り扱う言語としてVRML(Virtual Reality Modeling Language)が標準化され、広く使われてきている。

【0003】

現時点で公表されている最新版のVRML 2.0は、エレクトリック・コマース用の仮想モール構築、また、インターネットでのCAD等の3次元データの管理等で用いられている。

【0004】

一方、本発明の出願人は、観察者が直接奥行きを知覚できる立体表示装置として、リアクロスレンチrear-cross lenticular type方式の3Dディスプレイ(以下、3Dディスプレイを二次元表示装置と区別するために「立体表示装置」と略す)を開発しているが、出願人のこの表示装置以外にも多くの立体表示装置が提案されている。

【0005】

このVRML 2.0で記述された3次元空間を表示するためには、VRMLビューワ装置が必要となる。一般に普及しているVRMLビューワ装置は、記述している空間は3次元であるものの、最終的な表示は通常のモニタを用いた2次元表示(以下、「立体表示装置」との対比上「二次元表示装置」と略す)に留まっている。

【0006】

この理由は、従来のVRMLビューワ装置が、立体視用画像を効率よく生成することができず、そのために、リアルタイムの立体画像の再現に向かないこと、立体視を実現するための一部パラメータ(以下、「立体視パラメータ」と略す)がVRML 2.0でサポートされていないために、事実上、立体視を実現することができなかつたからである。

【0007】

立体視パラメータとは、基線長、輻輳角、視野角などに関連するパラメータをいう。3次元幾何モデル・仮想空間・仮想物体等といった対象を、立体表示する場合において、その対象の大きさの単位又は値が明瞭である場合は、その大きさを認識でき、且つ立体視が可能なように表示することは可能である。しかしながら、例えば基線長の設定を誤ると立体視は不可能となる。従って、立体視パラメータの初期値を正しく設定することは大きな要請である。

【0008】

この点で、本出願人は、特願平10-185135号(平成10年6月30日)において、基線長の初期値を、画像空間の大きさから求め、例えば65mmとし、又、輻輳角を0度(即ち、平行視)とすることを提案した。

【0009】

立体視パラメータは初期的に設定する時点でのみ問題となるのではない。

【0010】

例えば、仮想空間で体験者にウォークスルーシステムを提供している最中において、対象

10

20

30

40

50

を観察する視点条件（視点位置・視軸方向）は動的に変化するものである。そのために、基線長や輻輳角の初期値を適正に設定しても、ウォークスルーに伴う視点条件の変化により、基線長や輻輳角を動的に変化させる必要がある。

【0011】

この要請に対して、本出願人は、上記特願平10-185135号において、グラフィック・ユーザインタフェースを立体ディスプレイの表示画面上に設け、このユーザインタフェースを介して、体験者が上記立体視パラメータを動的に変更する途を提案した。

【0012】

【発明が解決しようとする課題】

上記特願平10-185135号は、立体視パラメータの初期値、更にはリアルタイム値を提案するものではある。10

【0013】

しかしながら、この従来技術において、基線長を例えば65mmに設定することは、取り敢えず立体視を可能とするために、その画像空間の実際の大きさを無視して、画像空間を人間の大きさに無理に合わせようとするものであることから、観察者若しくはユーザに違和感を与えることは禁じ得ない。

【0014】

又、輻輳角の初期値を0度とすることは、立体視を可能とはするが、これも本来的に望ましい立体視とは限らない。

【0015】

なお、立体視パラメータの初期値及びリアルタイム値を設定する課題はVRMLデータを扱うシステムに限定されない。およそ立体視パラメータが設定されていない三次元空間データ若しくは立体画像データ、あるいは立体視パラメータを動的に変更する必要のある仮想空間での三次元空間データであれば、あらゆる空間データにおいて発生する課題である。20

【0016】

【課題を解決するための手段】

本発明は上述の従来技術及び本出願人の提案技術に鑑みてなされたもので、その目的は、立体視パラメータの最適な初期値を自動的に設定する画像処理装置及びパラメータ設定装置を提案する。30

【0017】

本発明の他の目的は、立体視パラメータの最適リアルタイム値を自動的に更新する画像処理装置及びパラメータ設定装置を提案する。

【0018】

本発明のさらに他の目的は、ユーザ若しくは観察者の要求に応じて立体視パラメータの最適値を設定する画像処理装置及びパラメータ設定装置を提案する。

【0019】

上記課題を達成するための、本発明の請求項1にかかる、任意の物体の左目用画像と右目用画像とを表示して前記物体の立体視を得る立体画像処理装置は、入力画像の奥行き方向について、観察者の視点位置に最も近い、前記物体上の物体点を検出する検出手段と、前記物体点が二次元描画空間に描画された際に現れる左右間のズレ距離を設定するズレ距離設定手段と、前記左目用画像と右目用画像のいずれか一方における前記物体点の画像位置と、前記ズレ距離とに基づいて、基線長の値を決定する決定手段と、決定された基線長の値を立体視パラメータとして用いて、前記左目用画像と右目用画像を生成する生成手段とを具備することを特徴とする。40

【0020】

立体視パラメータとして例えば基線長を、上記ズレ距離の三次元空間での距離に設定すれば、立体視は確保される。即ち、立体視パラメータがユーザ若しくは観察者の手を煩わすことなく設定される。

【0021】

本発明の好適な一態様である請求項2に拠れば、前記決定手段は、
前記物体点の前記二次元描画空間上の座標から、観察者の基線長方向に、前記ズレ距離だけ離間したズレ点の位置を演算するズレ演算手段と、
前記ズレ点の三次元空間における座標を演算する座標変換手段と、
前記ズレ点の三次元座標と前記物体点との距離に基づいて前記基線長の値を決定する基線長決定手段を具備することを特徴とする。

【0025】

前記物体点（例えば仮想物体上の観察者に最も近い点）は二次元描画空間中において特定されねばならない。そこで、本発明の好適な一態様である請求項3に拠れば、前記物体点は所定の条件を満足すべきものとされ、満足する物体点が前記二次元描画空間において高速に探索されるように、前記描画空間における複数の探索位置が前もって離散的分布を有するように設定されている。

10

【0031】

本発明の目的は、立体視パラメータの値の最適値を自動的に設定するものであるが、その値を、別途、ユーザ自身が設定するようにしてもよい。そこで、本発明の好適な一態様である請求項4に拠れば、手動で立体視パラメータの値を設定するユーザインターフェース手段を更に具備する。

【0032】

請求項4の装置においては、立体視パラメータの値が、自動設定される場合とユーザが設定しようとする場合とで競合する。そこで、本発明の好適な一態様である請求項5に拠れば、前記ユーザインターフェース手段と前記決定手段との競合の調停をとる手段を更に具備する。後述の実施形態では、いずれのモードをとるかを指定するアイコンが表示画面上に表示される。

20

【0033】

立体視パラメータとして輻輳角を設定する場合には、請求項6の決定手段は、入力画像の仮想空間の奥行き方向の広がりを求め、観察者の左右の視点位置から見える、この仮想空間の奥行き方向の広がりの中間位置の方向に基づいて輻輳角の値を決定する。

【0035】

本発明の好適な一態様である請求項7に拠れば、前記ズレ距離設定手段は、立体画像が表示される表示装置の画素サイズと、この表示装置の表示画面から観察者までの距離に基づいて前もって前記ズレ距離を決定する。これらの要素によって決定されたズレ距離の最大値に一致するようなパラメータ、例えば基線長、は立体視を保証する。

30

【0037】

左右の視点位置での視軸が平行する場合には、仮想空間の物体の最近点が最も視差距離が大きい値を示す。そこで、本発明の好適な一態様である請求項8に拠れば、前記物体点は、前記左目用画像と右目用画像のいずれか一方の画像データに、観察者の視点からの前記二次元描画空間への透視変換を施し、透視変換された画像データのなかから最も小さな奥行き値を有する画素に決定される。

【0045】

本発明の目的は、請求項1乃至9のいずれかに記載の立体画像処理装置を具備する立体画像表示装置によっても達成できる。

40

【0046】

【発明の実施の形態】

以下、本発明を、本出願人が開発したリアクロスレンチ方式の3Dディスプレイを立体表示装置として用い、ウォークスルーライブ体験をユーザに提供するウォークスルーライブシステムに適用した実施形態を説明する。尚、本発明は、リアクロスレンチ方式の3Dディスプレイのみに適用可能なものではないこと、ウォークスルーライブ体験システムにのみに適用可能なものではないことは、後述の説明から明らかになるであろう。

【0047】

ウォークスルーライブシステム

50

第1図は、このウォークスルーシステムに適用される立体画像表示装置の構成を示す。この表示装置1000は、通常のワークステーションのように、ディスプレイ10と、ディスプレイ10に表示するデータを一時的に格納するフレームバッファ11と、描画用画像データ（の他にマスク情報や距離情報も）を記憶するフレームバッファ11と、CPU12と、キーボード13と、マウス14と、プログラムなどを格納する記憶装置15と、三次元空間データベースを記憶する大容量記憶装置16と、さらにディスプレイ10からユーザ（観察者）までの距離LGHを測定する測距センサ17と、観察者の視点の視点位置・姿勢OBを検出する視点位置センサ18とを有する。尚、観察者の位置及び視点が予め決められた範囲内でのみ許されるのであれば、測距センサや視点位置センサは不要となる。

10

【0048】

ディスプレイ10は、立体視が可能となるように、本出願人により提案されているリアクロスレンチ方式の眼鏡無し（即ち、裸眼）3Dディスプレイを用いている。

【0049】

リアクロスレンチ方式の裸眼立体ディスプレイ10は、第2図に示すようにTFTの波晶（LCD）パネル20とバックライト・パネル24との間に2枚のレンチキュラーレンズ（Hレンチキュラーレンズ21、Vレンチキュラーレンズ22）と市松状マスクパネル23を配置した構成を特徴としている。Vレンチキュラーレンズ22は、バックライト24からの照明光の指向性を走査線毎に変えて左眼方向と右眼方向とに分離する役割をもつ。Hレンチキュラーレンズ21は、市松状マスク23の1つの水平ラインの開口からの光を、LCDパネル20の一走査線上に集光させ、クロストークの発生を防ぐとともに、開口部からの光を観察者の方へ発散する光束とすることで垂直方向の立体視域を拡大している。

20

【0050】

第3図は、第2図のLCDパネル20の一部領域Aを拡大表示したものである。このディスプレイ10では、第3図に示すように、立体視用の左右ステレオ画像対を横ストライプ状に交互に並べて合成した画像をLCDパネル20に表示する。そのため、縦ストライプ画像を入力とする通常のレンチキュラ方式と比較して、波晶の縦ストライプに対応した色分離が発生しない、波晶シャッター眼鏡などで標準に使われる時分割フィールド順次方式の水平ストライプ合成画像と互換性がある、等の特長を持つ。

30

【0051】

ウォークスルすべき仮想空間の立体画像はディスプレイ10上に標示される。ウォークスルーを体験するユーザは、マウス14又はキーボード13により、進行方向及び移動距離を指定する。CPU12は、メモリ15に記憶された所定のウォークスルーアプリケーション・プログラムに従って、移動先の仮想画像を生成し、その仮想画像をディスプレイ10上に表示させることにより、仮想空間を疑似体験させる。

【0052】

今、このウォークスルー空間において、第4図のように、4つの狭領域1, 2, 3, 4が設定されている場合を想定する。異なる狭領域に跨ってウォークスルーする場合には、マウスにより移動経路5又は6などを指定すればよい。このウォークスルー空間の狭領域の夫々について、三次元空間データベースがメモリ16に記憶されている。従って、本システム、即ちアプリケーション・プログラムは、メモリ16に記憶されている各狭領域の三次元空間データにおいて立体視パラメータが設定されていない場合には、後述するよう、立体視パラメータの初期値を設定する。又、例えば、別の狭領域（空間）に移動するような場合には、視点条件（視点位置、視軸方向）が大きく変わることとなり、このために、立体視パラメータをその時点で移動先の狭空間での視点条件（視点位置、視軸方向）に合致するように変更する必要がある。

40

【0053】

このように、本実施形態のシステムは、立体視パラメータの初期最適値のみならず動的な最適値を、ダイナミック（ユーザの手を介さずに、或いは必要に応じてユーザの指示により）に設定することができる。

50

【0054】

基線長最適値の決定原理

第5図は、メモリ16に記憶された空間データが合成されて表示される様子を示す。同図において、立体視のための空間データは、右目用画像31と左目用画像32とが前もってメモリ16に記憶され、適切な基線長と輻輳角が設定されたならば第5図の33に示すように、ディスプレイ10上では、2つの画像31, 32が見かけ上のズレを持って、例えば第6図のように表示される。このズレ量が適切に設定されれば、体験者には、画像31, 32は融像した立体画像として感知される。

【0055】

本システムにおける基線長の最適値の決定は、ステレオ画像31, 32が第6図のように表示された場合において、画面上に現れる画像31と画像32の視差の最大値（以下、「最大視差量」と呼ぶ）を問題にする。ここで、視差とは、本来は1つの点（仮想物体或いは現実物体の上の点）についての、二次元空間で描画表示された画像31と画像32上における夫々の点の距離であって、そのような距離の中（仮想物体或いは現実物体の上の点は多数ある）で最大値を指すものとする。

【0056】

表示された物体画像31, 32が体験者によって立体的に知覚できるか否かは、ディスプレイ10の大きさ、特性（例えば画素ピッチ）、視点から画面までの距離などの条件の他に、上述の視差量が、密接に関係することはよく知られている。本実施形態のシステムは、この最大視差量を、

- ・ディスプレイ毎に調整する、
- ・一定量以下に調整する、
- ・自動的に調整する

ものである。

本実施形態の基線長決定の原理を第7図乃至第10図を用いて説明する。

【0057】

尚、説明を簡略化するために、物体として四角錐30を想定する。メモリ16に記憶されたこの物体30の三次元空間データは、この四角錐30を第7図に示すような配置の左右の仮想カメラで撮影すると、第6図のようなステレオ画像が得られるデータである。この仮想カメラによる撮像の様子を上方から見たとすると、物体30と左右夫々の仮想カメラの視点との関係は第8図のようになる。尚、第8図において、仮想撮像面とは、その仮想カメラの撮像面であり、その解像度はディスプレイ10の解像度に一致させており、仮想カメラの1画素分の画像データは、ディスプレイ10の表示画面の1画素に表示される。このために、スケールを考慮しなければ、仮想撮像面上の位置は、表示ディスプレイ上の表示位置に等価である。

【0058】

第8図において、四角錐30の頂点40は、右仮想カメラの撮像面上（即ち、表示面上）の点41Rに、左仮想カメラの撮像面上（即ち、表示面上）の点41Lに射影される。即ち、左右の仮想カメラの視点位置間の距離（即ち、基線長）が距離42に設定されている場合に、物体30上の点40の三次元位置と、基線長42と、射影点41L, 41R間の距離43とは、夫々、一対一に対応することになる。

【0059】

第6図に関連して説明したように、対象（この場合は物体30）が変化（変形や移動）していない場合には、基線長の変化量が物体の任意の点の仮想画面上での最大視差量に影響する。特に、第8図のように、2つの仮想カメラの視軸方向が平行である場合には、最大視差量は基線長の変化のみによって支配される。この場合、最大視差量の変化と基線長の変化は一次の関係にあるので、立体視を可能にする（即ち、最大視差量を一定値以下にする）ためには、立体視パラメータとしての基線長を調整すればよいことになる。

【0060】

一方、第7図のように仮想カメラの視軸は平行である場合には、視点に最も近い物体上の

10

20

30

40

50

点の視差量が最大となるから、このような視点に最も近い物体上の点の視差量を計算し、この視差量に基づいて基線長を計算すれば、その物体（第7図の例では四角錐30）の全ての点についての立体視が保証されることがわかる。

【0061】

従って、このウォークスルー立体視システムでは、

【0062】

- ・物体上の点のうち、仮想カメラ視点に最も近い点（或いは、最も近いと思われる点）（以下、「最近点」と略称する）を決定する処理（「最近点決定処理」と呼ぶ）と、
- ・この最近点の撮像面上でのズレ量を基にした基線長の決定処理（「基線長決定処理」と呼ぶ）、

という2つの処理が組み込まれている。

【0063】

基線長決定処理の原理

まず、最近点が既に決定されている場合において、その最近点に基づいた基線長決定処理の原理について第9図、第10図を用いて説明する。

【0064】

第9図において、視点60とは、例えば第7図の右仮想カメラ（左仮想カメラでもよい）における視点をいい、仮想撮像面61は、その仮想カメラの撮像面であり、その解像度はディスプレイ10の解像度に一致させている。即ち、仮想カメラの1画素分の画像データは、ディスプレイ10の表示画面の1画素に表示される。

【0065】

第9図において、仮想線53の長さは、ディスプレイ10上において標示されるステレオ画像を立体視として保証するための最大視差量に相当する長さに設定されているとすると、この仮想線53を仮想画面上で第9図の位置に設定することは、その右端点52は、三次元物体の最近点50の射影の筈である。すると、右端点52が最近点50の射影として設定され、仮想線53の長さが立体視として保証するための最大視差量に相当する長さとして設定されるならば、仮想線53の左端点54は、同じ最近点50の左仮想カメラの仮想撮像面上への射影点の筈である。すると、仮想線53の左端点54を、左仮想カメラではなく、右仮想カメラでもって逆透視して三次元空間に射影することにより、点55が得られる。即ち、三次元空間において点50と点55を結ぶ線分51は、仮想撮像面上の仮想線53（即ち、最大視差量を表す）の3次元空間への射影を表す。尚、点54の射影点55は、右仮想カメラによる透視変換行列を V_R とすると、その逆行列 V^{-1}_R によって得られる。

【0066】

仮想線53の長さは立体視を保証する最大視差量に相当するように設定されているから、第10図に示すように、左右の仮想カメラの距離（即ち、基線長）を線分53の長さに設定すれば、観察者には、最近点50の融像を見ることができる。仮想カメラが互いに平行視軸を持つように設定されており、仮想線53の長さが最大視差量に相当するように設定されているので、観察者は、最近点50のみならず、その物体の全ての点を、即ち、その物体を立体感を持って見ることとなる。

【0067】

かくして、本実施形態システムの基線長決定処理は、線分51の長さを決定することに他ならない。

【0068】

最近点決定処理

基線長決定処理は、その処理に先立って、最近点が決定されていることが前提である。最近点は、観察者若しくはユーザがディスプレイ10の画面を見ながらマウスなどによりマニュアルで選定してもよいが、それでは、ウォークスルーのリアルタイム性が失われてしまう。本実施形態のシステムは、最近点を、システムがユーザに替わって自動的に求めて決定する。

10

20

30

40

50

【0069】

本実施形態のシステムでは、三次元空間データは、メモリ16に、

- ・形状を記述するデータ（点群に関する三次元位置データ（X, Y, Z）と各点間のパッチリンク）と、
- ・表面の特性（色・反射率・透過率・テクスチャ）を記述するデータと、
- ・光源位置を記述するデータと、

等を記憶して構成される。尚、通常、三次元空間データは、この他にも、初期値や、数個のセットの視点位置や方向、動的に動いている物体の動作方法、音の生成方法の記述（ある物体にさわると音がするなど）、手続きの実行できるスクリプト言語の記述などが可能である。これらのうち、視点位置、方向以外は、今回の発明とは直接関係ないので、ここでは詳細な説明は省略する。10

最近点は、点群に関する三次元位置データ（X, Y, Z）と、ユーザ（即ち、仮想カメラ）の視点位置によって変化する。従って、最近点は、ユーザの視点位置を考慮して決定しなければならない。

【0070】

ユーザの視点位置から物体までの距離は、コンピュータグラフィクスの技術分野で利用されている「zバッファ値」を用いることで得ることができる。まず、zバッファおよびzバッファ値に関して簡単に説明する。

【0071】

三次元空間上の一点（ X_i , Y_i , Z_i ）は、表示のために二次元の平面へと透視変換する必要がある。第11図のようなユーザの視点位置と視体積とを想定し、その視体積を。20

【0072】

$b = bottom$, $t = top$, $l = left$, $r = right$
 $n = near$, $f = far$

【0073】

と定義すると、三次元空間状の一点（ X_i , Y_i , Z_i ）は、仮想カメラの撮像面の（ x_i , y_i , z_i ）へと次式のように変換される。ここで、この変換式では同次座標系を利用している。

【0074】

[数1]

$$\begin{bmatrix} X'_i \\ Y'_i \\ Z'_i \\ W \end{bmatrix} = V_R \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix}$$

【0075】

但し、

【0076】

[数2]

$$V_R = \begin{bmatrix} \frac{2n}{r-l} & 0 & \frac{r+l}{r-l} & 0 \\ 0 & \frac{2n}{r-b} & \frac{t+l}{t-l} & 0 \\ 0 & 0 & \frac{-(f+n)}{f-n} & \frac{-2f \cdot n}{f-n} \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

【0077】

10

20

30

40

50

[数3]

$$(x_i, y_i, z_i) = \left(\frac{x'_i}{w}, \frac{y'_i}{w}, \frac{z'_i}{w} \right)$$

【0078】

上記透視変換により得られた (x_i, y_i, z_i) のうち、 (x_i, y_i) は仮想撮像面上での 2 次元位置であり、 z_i が z バッファ値となる。この z バッファ値は、透視変換前の三次元空間中での奥行順を保持したまま、視体積の奥行方向を正規化したような値である。通常、コンピュータグラフィクスの技術分野では、この z バッファ値を利用して表示時に、面や線の隠れの判定（陰面消去・陰線消去処理）を行っている。これは、表示画面（仮想撮像面）用に必要な RGB 色バッファメモリの他に、 z バッファというフレームバッファメモリを用いることで実現している。これは、

10

- ・透視変換により得られた (x_i, y_i, z_i) があるときに、 (x_i, y_i) の位置に相当する画素位置の RGB 色バッファ及び z バッファに、色の計算結果および名を格納する、
- ・このとき、既に格納してある z バッファのその画素の値の方が小さい場合には更新を行わない、

という処理である。本来、 z バッファおよび、上記数式 1 ~ 3 による結果 z_i は、この処理のために利用されている。ここで、 z バッファは、この目的のためにのみ用意された専用ハードウェアメモリとして用意してある場合もあれば、専用のメモリではなく通常の主記憶メモリに割り当てられてソフトウェア的に実現してある場合もある。

20

この処理にあるように、 z バッファは、ユーザ視点から見えている対象物体の最も近い表面上の奥行き情報を保持しており、また、透視変換前の 3 次元空間上での奥行順も保持しているため、ユーザ視点からの「最近点」を求める際に利用できる。すなわち、最小の z バッファ値を有する点が「最近点」となる。

【0079】

ここで、 (x, y) の位置の z バッファ値を得るためにには、例えば、例えば、次のコマンド、

【0080】

```
g l R e a d P i x e l s ( x , y , 1 , 1 , G L _ D E P T H _ C O M P O N E N T ,  
G L _ F L O A T , & z )
```

30

【0081】

を用いることができる。最近点 (x_n, y_n, z_n) は、下記の式を満足する点をフレームバッファ 11 内を探索して得る。

【0082】

[数4]

$$(x_n, y_n, z_n) = \{x_i, y_i, z_i : i = n\}$$

$$n = \{z_n : \min(z_i)\}$$

【0083】

40

である。

ところで、[数3] や [数4] を用いてフレームバッファ内で最近点を探索することは、データの読み出しがかかる（特にハードウェアが通常のフレームバッファであれば特にそうである）。そこで、最近点の探索を高速化するために、探索点をサンプリングする。探索点のサンプリングは、フレームバッファ内で、即ち、第12図のように、撮像画面の二次元空間上において行う。即ち、サンプリング点 i の分布を、第13図～第18図のいずれかに前もって設定する。同図において、各点はサンプリング点を表す。要は、サンプリング点のいずれかに、実際の最近点が含まれていれば好ましい。

【0084】

第13図の例では、二次元空間 (x, y) の全てをサンプリング点とするものである。

50

この手法は最近点の探索の精度が上がるが、前述したように、探索に時間がかかるという欠点がある。

【0085】

第14図の例では撮像面の中央領域のみをサンプリングする。観察者（ユーザ）は、仮想カメラの撮像空間、即ち、ディスプレイ10の表示空間の中央領域を見つめていることが多い。このことは、中央領域で立体視が得られれば、その周辺で立体視が得られなくても問題はないことを意味する。そこで、サンプリング点を中央領域に集中して高速化を図るのである。

【0086】

第15図の例は、中央の一点に探索を集中する。高速化を極限まで達成できるが、立体視が得られない確率が向上する。例えば、遠方の点を最近点と誤認識すれば、観察者は、実際の最近点近傍の（ディスプレイ10上での）表示画像を立体として認知できないであろう。

【0087】

第16図の例では、第13図の例から、サンプリング点の密度を全体的に一様に低下させた。

【0088】

第17図の例は、ディスプレイ10上で、上下方向での立体視の確保を重要視し、左右方向での立体視性の低下を甘受することとしたサンプリングパターンの例である。

【0089】

第18図の例は、中心から、放射状に、より離間するにつれて、立体視性が低下することを甘受する例である。

【0090】

第19図は、最近点の決定のための制御手順である。

【0091】

ステップS2では、ユーザの視点位置OB（視点の三次元位置とその三軸周りの頭部の回転角度）と共に、視点の画面からの距離LGHとを入力する。ステップS4では、第13図乃至第18図のパターンの中から、ユーザの視点位置OBに適したサンプリングパターンを決定する。ステップS6では、視点位置OBや視点までの距離LGHに基づいて右仮想カメラのビューリング変換行列VRを決定する。

【0092】

ステップS8では、得られたビューリング変換行列VRを用いて、物体の三次元空間データであるベクトルX（ステップS8ではベクトルXで表す）を撮像面での二次元画像データであるベクトルx（ステップS8ではベクトルxで表す）に変換する。このベクトルxはdepthバッファ値dを有していることは前述したとおりである。

【0093】

ステップS10では、ステップS4で指定されているサンプリングパターンに従って、1つの点xiの奥行き値diをバッファから得る。ステップS12では、奥行き値diの最小値dnを探索する。全てのサンプリング点iについて探索した結果の最近点x = xnをステップS16で出力する。

【0094】

尚、複数の最近点が得られたならば、そのうちのユーザの視線方向に最も近いものを、視点位置情報OBに基づいて決定し、これを用いるものとする。

【0095】

また、ステップS4のパターン決定は、システムが、第13図～第18図に示した分布パターンに似せたアイコンをディスプレイ10上に標示し、いずれか1つのアイコンを、ユーザがキーボード13により指定するようにしてもよい。

【0096】

かくして、本実施形態の制御によれば、

【0097】

10

20

30

40

50

i : 最近点がユーザの手を煩わすことなく自動的に決定される。

i i : 最近点の決定は周知の z バッファ（若しくは depth バッファ）を利用するので、汎用性が向上する。

i i i : 最近点の探索はサンプリングポイントの分布パターンを前もって複数通り設定しておき、そのうちの最適な 1 つをシステム若しくはユーザの選択により指定することにより、最適な最近点の決定が高速処理を害することなく実現する。

i v : サンプリングポイントの分布パターンの指定を自動で行う場合には、ユーザの視線方向を考慮して行われるので、ユーザが関心ある方向にある領域が確実に立体視される。

【 0 0 9 8 】

立体視パラメータの決定の制御手順

第 20 図は、実施形態に係る立体視パラメータの決定の制御手順のフローチャートである。

【 0 0 9 9 】

まず、ステップ S 2 0 で、ディスプレイ 1 0 の各種のパラメータを入力する。パラメータとは、立体視を保証する最大視差量の値に影響するもので、例えば、ディスプレイ 1 0 の 1 画素の大きさ (width × height)、1 画面の大きさ (SIZE) などであり、これらのハード情報はシステムが既に把握しているものである。

【 0 1 0 0 】

ステップ S 2 2 では、ユーザの位置・姿勢 O B 並びにディスプレイ画面から ユーザ の視点までの距離 L G H を前記のセンサから入力する。

【 0 1 0 1 】

ステップ S 2 4 では、上記の情報に基づいて、立体視を保証する最大視差量の値を設定する。立体視を保証する最大視差量の値は、ディスプレイ画面の大きさ仕様（例えば、VGA や X VGA 等の仕様）毎に前もって値を決めておくのがよい。ステップ S 2 6 では、設定された最大視差量（例えば 10 mm の距離）を画素数で表したオフセット量 offset に変換する。これにより、立体視が保証されるには、ディスプレイ 1 0 の画面上で offset 分の視差が発生するような基線長を決定するればよいことになる。第 9 図、第 10 図を参照。

【 0 1 0 2 】

ステップ S 2 8 では、ユーザの初期位置・姿勢、視軸、画角を設定する。ステップ S 3 0 では、ステップ S 2 8 で設定したユーザの視点位置・姿勢に関する情報に基づいて、左右の夫々のカメラの各々の透視変換行列 V_L , V_R 並びに各々の逆透視変換行列 V^{-1}_L , V^{-1}_R を演算しておく。ステップ S 3 2 では、メモリ 1 6 に記憶されている三次元空間データを読み出す。

【 0 1 0 3 】

以上の処理により、立体視パラメータを決定するための土台が整ったことになる。

【 0 1 0 4 】

次に、ステップ S 3 4 では、第 19 図に関連して説明した最近点の決定処理を行う。第 19 図の制御手順は、最近点 (x_n , y_n , d_n) を出力する。

【 0 1 0 5 】

次に第 20 図のステップ S 3 6 の基線長の決定処理が実行される。この基線長の決定処理の詳細は第 21 図に示される。

【 0 1 0 6 】

第 21 図に基づいて、この処理を説明する。

【 0 1 0 7 】

ステップ S 4 0 では、最近点の決定処理が出力した最近点 (x_n , y_n , d_n) を所定のレジスタに読み出す。この最近点 (x_n , y_n , d_n) は、第 8 図の最近点 5 0 が仮想撮像画面上に射影された点 5 2 に相当する。ステップ S 4 2 では、最近点 (x_n , y_n , d_n) を正規化処理して右仮想カメラによる射影変換後の空間における画素位

10

20

30

40

50

置 (x_R , y_R , z_R) を下記の [数 5] に従って算出する。

【 0 1 0 8 】

[数 5]

$$(x_R, y_R, z_R) =$$

$$\left(2 \cdot \left(\frac{x}{\text{width}} - 0.5 \right), 2 \cdot \left(\frac{y}{\text{height}} - 0.5 \right), d \right)$$

【 0 1 0 9 】

ステップ S 4 4 では、画素位置 (x_R , y_R , z_R) から、カメラ座標系での三次元 10 空間中の点 (第 8 図の点 5 0) の座標値 $P_0 = (X_R, Y_R, Z_R)$ を求める。即ち、右仮想カメラの変換行列 V_R の逆行列を、

【 0 1 1 0 】

[数 6]

$$V_R^{-1} = \begin{bmatrix} \frac{r-\ell}{2n} & 0 & 0 & \frac{r+\ell}{2n} \\ 0 & \frac{t-b}{2n} & 0 & \frac{t+b}{2n} \\ 0 & 0 & 0 & -1 \\ 0 & 0 & \frac{-(f-n)}{2f \cdot n} & \frac{f+n}{2f \cdot n} \end{bmatrix}$$

20

【 0 1 1 1 】

とすると、座標値 P_0 は、

【 0 1 1 2 】

[数 7]

$$\begin{bmatrix} X'_R \\ Y'_R \\ Z'_R \\ W_R \end{bmatrix} = V_R^{-1} \begin{bmatrix} x_R \\ y_R \\ z_R \\ 1 \end{bmatrix}$$

30

【 0 1 1 3 】

[数 8]

$$P_0 = (X_R, Y_R, Z_R) = \left(\frac{X'_R}{W_R}, \frac{Y'_R}{W_R}, \frac{Z'_R}{W_R} \right)$$

【 0 1 1 4 】

かくして、第 9 図の点 5 0 に相当する P_0 が得られた。

次に、ステップ S 4 6 では、最近点 (x_n , y_n , d_n) を左仮想カメラにより透視すると、前述 40 の offset 数の画素分だけ y 軸方向にずれるのであるから、そのずれた画素位置 (x_L , y_L , z_L) は、下記の [数 9]

【 0 1 1 5 】

[数 9]

$$(x_L, y_L, z_L) =$$

$$\left(2 \cdot \left(\frac{x}{\text{width}} - 0.5 \right), 2 \cdot \left(\frac{y - \text{offset}}{\text{height}} - 0.5 \right), d \right)$$

【 0 1 1 6 】

50

によって得られる。この画素位置 (x_L , y_L , z_L) は第 8 図の点 54 に相当する。この点 54 の左カメラの座標系から逆透視した三次元座標位置 $P_1 = (X_L, Y_L, Z_L)$ を、左仮想カメラの逆透視変換行列 V_L^{-1} (実質的に V_R^{-1} に等しい) により、

【0 1 1 7】

[数 1 0]

$$V_L^{-1} = \begin{bmatrix} \frac{r-\ell}{2n} & 0 & 0 & \frac{r+\ell}{2n} \\ 0 & \frac{t-b}{2n} & 0 & \frac{t+b}{2n} \\ 0 & 0 & 0 & -\frac{1}{2f \cdot n} \\ 0 & 0 & \frac{-(f-n)}{2f \cdot n} & \frac{f+n}{2f \cdot n} \end{bmatrix}$$

10

とすると、座標値 P_1 は、

【0 1 1 8】

[数 1 1]

$$\begin{bmatrix} X'_L \\ Y'_L \\ Z'_L \\ W_L \end{bmatrix} = V_L^{-1} \begin{bmatrix} X_L \\ Y_L \\ Z_L \\ 1 \end{bmatrix}$$

20

【0 1 1 9】

[数 1 2]

$$P_1 = (X_L, Y_L, Z_L)$$

$$= \left(\frac{X'_L}{W_L}, \frac{Y'_L}{W_L}, \frac{Z'_L}{W_L} \right)$$

30

【0 1 2 0】

となる。次にステップ S 50 では、距離 $|P_0 - P_1|$ に従って基線長を決定する。以上が基線長の決定処理の制御手順である。

【0 1 2 1】

第 20 図の立体視パラメータの制御手順のフローチャートに戻ると、ステップ S 36 の次には、ステップ S 38 で輻輳角の決定処理が行われる。この処理は後に説明する。

【0 1 2 2】

以上のようにして、基線長と輻輳角などの立体視パラメータの最適値が、ユーザの手を煩わさずに、初期的に或いはリアルタイムで決定することができる。

【0 1 2 3】

40

ウォームスルーリーのための制御

第 22 図は、本立体視パラメータ決定システムをウォームスルーリー提供システムに組み込んだ場合の制御手順を示す。即ち、ウォームスルーリーに特有な問題として、第 4 図に示すように、ユーザが視点位置を移動していくにつれて、立体視パラメータを変更しなければならないほど、視点条件が変更される場合がある。即ち、立体視パラメータをウォームスルーリーに伴ってリアルタイムにまたはオンドマンドに変更する必要がある。第 22 図の制御手順は、初期的に立体視パラメータを自動で最適設定できるのみならず、ユーザからのオンドマンドに従って立体視パラメータを最適に設定する機能が盛り込まれている。

【0 1 2 4】

第 22 図の制御手順に従って、ステップ S 60 が実行されると、第 20 図に関連して説明

50

した「立体視パラメータの決定」処理が行われる。これにより、立体視パラメータの初期値が、その時点のユーザの位置やディスプレイ伸しように即して最適な値として設定される。ステップS62では、このようにして設定した立体視パラメータに従って左目用と右目用の画像を描画し、ステップS64ではそれらの画像を合成して表示する。

【0125】

このステップS64の時点では、ディスプレイ10の画面上には、ウォームスルーネットワークの仮想空間が表示されている。

【0126】

ユーザは、前述したように、マウスなどを用いてウォームスルーネットワーク方向を指定すると、その移動指示に従って仮想空間の表示画面が変化していく。移動指示の入力は「インタラクション操作」として入力され、そのための操作がステップS72で実行される。10

【0127】

ユーザは、ステップS66で「インタラクション操作」以外の種々のコマンド指示を入力することができる。

【0128】

その指示が終了であれば本システムは終了するであろう。

【0129】

ユーザが立体視パラメータの変更を希望するときは、キーボードの所定のキーを押下することにより、ステップS74が実行される。ステップS74では、立体視パラメータの決定処理（第20図の制御手順）を再実行する。即ち、その時点での、ユーザの視点位置や条件に即した立体視パラメータが再度設定される。換言すれば、第22図の制御によれば、仮想空間の変遷に係わらず立体視が確保できるので、異なる大きさの小空間を複数持つような仮想空間をウォームスルーネットワークするなどの用途に特に適している。20

【0130】

最適輻輳角の自動設定

第20図のステップS38の輻輳角の決定処理（自動設定）はオプションの処理ルーチンである。立体視において、輻輳角が必要か否かは、応用分野或いはその応用分野でのユーザに応じた個人差があるからである。本システムでは、そのために、第22図のウォームスルーアプリケーション・プログラムが実行された直後において、そのアプリケーション・プログラムが輻輳角の自動設定を行うか否かをユーザに問うダイアログをディスプレイ10に表示する。ユーザはそのダイアログにおいて、輻輳角の自動設定を行う（即ちステップS38を実行する）か、行わない（即ち、ステップS38をスキップする）かを選択する。ステップS38は選択されたときのみ実行される。30

【0131】

第23図、第24図は、輻輳角の自動決定の原理を説明する。即ち、ステップS38は、入力された仮想空間の三次元データに基づいて輻輳角を決定する。第23図において、例えば仮想空間に円錐60と四角錐70の仮想物体があるとする。勿論、これら2つの仮想物体以外にも仮想物体が存在することは許される。これらの仮想物体は三次元座標情報を有するから、この仮想空間の中の最近点と最遠点とが存在する。第23図の例では、最近点は四角錐70の頂点が存在する垂直平面71上有り、最遠点は円錐60の周囲が存在する垂直平面61上有り。輻輳角の最適な値は、種々の学説により、まちまちであるが、本実施形態では、第24図に示すように、垂直面61と垂直面71の中間位置にある垂直面80上有る点と、左右の仮想視点とが成す二等辺四角形の頂点の角度（の半分）であるを輻輳角とする。40

【0132】

2つの仮想カメラの視軸が平行である場合において最近点のみを考慮したのは、最近点において視差量が最大値をとるからである。しかしながら、仮想カメラが平行でない、即ち、0度でない輻輳角を有する場合には、最近点と最遠点において視差量が最大値を示すことになる。従って、0度でない輻輳角が損する場合には最近点のみならず最遠点を考慮しなければならないのである。50

【0133】

第25図は、ステップS38の詳細な処理手順を表す。

【0134】

最近点は第20図のステップS34で既に決定されているので、輻輳角の決定のために、最近点を改めて求める必要はない。そこで、ステップS38aでは最遠点を探索する。最遠点の探索処理は、第19図に示された最近点の探索処理を一部変更した処理手順を援用することができる。その変更箇所はステップS12において、最遠点を探索するために、

【0135】

[数13]

$$d_n = \max(d_i, d_n)$$

10

【0136】

の処理を行う。更にステップS38bでは、奥行き範囲Dを決定する。ステップS38cでは、D/2の値に従って、輻輳角を第24図に示された演算手法に従って決定する。本発明は、その趣旨を逸脱しない範囲で種々変形が可能である。

【0137】

第19図乃至第21図の制御手順に示された立体視パラメータの決定手順は、仮想カメラの視軸が互いに平行（輻輳角 = 0度）である場合を想定していた。輻輳角 = 0度の方がユーザが疲労しない場合があるからである。しかしながら、三次元空間データの中には、予め輻輳角に関するパラメータを有するものがある。また、上記実施形態でも、ユーザが望む場合には、輻輳角の初期値が自動決定された。従って、最初から輻輳角を有するような画像データや、事後、輻輳角を付加された画像データのような場合には、輻輳角を考慮した基線長の決定を行わなければならない。

20

【0138】

輻輳角を考慮した基線長の決定は回転処理を行うことによって実現される。第26図は、輻輳角を考慮した基線長の決定する処理が、第9図に示された輻輳角を有さない場合の基線長の決定処理の原理に対して、いかなる変更処理を付加するかを説明している。

【0139】

即ち、第26図と第9図とを比較すると、最近点52を、画面上での最大視差量に相当する長さだけ平行移動して点54を得るのは変わらないが、第26図では、更に、点54を、右視点を中心にしてy軸の周りに輻輳角（=2°）だけ回転して点57を得、この点57に相当する三次元空間座標58を求め、点58を点P₂とすると、最近点50（=P₀）から点P₂までの距離を、輻輳角を考慮した場合の基線長とするのである。

30

【0140】

第26図に示された処理は、実質的には、ステップS48（第21図）での処理に、上記回転による座標変換処理Rを逆透視変換処理V_L⁻¹に付加するのである。この回転処理を付加されたステップS48を新たにステップS48'にすると、ステップS46で、前述したように、[数9]に従って点54に相当する画素位置（x_L, y_L, z_L）を得る。更に、ステップS48では、画素位置（x_L, y_L, z_L）を座標変換して、点58に相当する座標値P₂を、

40

【0141】

[数14]

$$\begin{bmatrix} X'_L \\ Y'_L \\ Z'_L \\ W_L \end{bmatrix} = [R \cdot V_L]^{-1} \begin{bmatrix} X_L \\ Y_L \\ Z_L \\ 1 \end{bmatrix}$$

【0142】

[数15]

50

$$\begin{aligned} P_2 &= (X_L, Y_L, Z_L) \\ &= \left(\frac{X'_L}{W_L}, \frac{Y'_L}{W_L}, \frac{Z'_L}{W_L} \right) \end{aligned}$$

【0143】

に基づいて求める。ここで、行列 R は、 y 軸周りの角度 δ の回転変換であり、具体的には、

【0144】

[数16]

$$R = \begin{bmatrix} \cos 2\delta & 0 & \sin 2\delta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin 2\delta & 0 & \cos 2\delta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

10

【0145】

という構造を有する。また、 V_L は左右の仮想カメラを同じカメラパラメータを持つものとすれば、 V_R （即ち、[数2]）に等しい行列である。

上記実施形態は、本発明の立体視パラメータの決定を、ウォーカスルーに適用したものであつた。特にこの仮想空間ウォーカスルーでは、立体視パラメータを、初期値のみならず、リアルタイム値も計算していた。

20

【0146】

この変形例は、立体視パラメータの演算輻輳角を減らすために、初期値のみを演算する仮想空間ウォーカスルーの制御であり、その手順を第27図に示す。

【0147】

である。

第27図の処理は、第22図の処理からステップS74の工程が除去されたものに等しい。

30

【0148】

尚、第22図、第27図、第28図、第29図に示された各フローチャート間で、同じステップ番号を有する工程は同じ内容の処理を行う。

【0149】

上記実施形態では、ディスプレイ10に表示するための左右目のための画像は、合成の都度、描画生成（レンダリング）していた。この第3変形例は、立体視パラメータの設定処理（第20図のステップS34）が、第19図のステップS8において右目用の画像を描画生成していた点を考慮し、この右目用画像を有効に利用して計算負荷を軽減しようとするものである。

【0150】

これは、第22図のステップS62において左右目画像の描画するという処理を、第28図のステップS62'のように、左目用画像のみの描画を行う処理に変更する。このために、第28図の制御手順では、立体視パラメータの再計算キーが押されるたびに、ステップS60で立体視パラメータの再設定が行われ（併せてステップS8で右目用画像の描画が行われ）、ステップS62'で左目用画像の描画が行われ、ステップS64で左右の画像の合成が成される。一方、インターラクション処理やその他の処理では、ステップS76で、左右目用の画像の描画が行われ、それらの画像はステップS64で合成されて表示される。

40

【0151】

このように、第3変形例では、立体視パラメータの再建さんが行われるときは、右目用画像の描画（レンダリング）の工程が1つ省略されて高速処理に寄与することとなる。

50

【0152】

第4変形例は、立体視パラメータの再計算を毎回行うというものであって、その制御手順を第29図に示す。即ち、インタラクション処理やその他の処理では、ステップS60にモードって、立体視パラメータの再設定が行われ（併せてステップS8で右目用画像の描画が行われ）、ステップS62'で左目用画像の描画が行われ、ステップS64で左右の画像の合成が成される。

【0153】

この第4変形例でも、右目用画像の描画（レンダリング）の工程が1つ省略されて高速処理に寄与することとなる。

【0154】

前述の実施形態では、`depth`バッファを用いることにより、複数の仮想物体があつても、描画時に全ての仮想物体の座標変換が行われるために、内部での座標変換が不要であった。この第5変形例は、第30図に示すように、対象の三次元空間中の物体を観察するカメラ座標系（カメラ位置を原点とする三次元座標系）に変換して、最も視点位置に近い最近点を直接的に計算するようとする。この変形例の手法では、3次元対象内の物体間の座標変換を考慮する必要がある。第30図の例では、カメラと部屋は世界座標系の位置で指定されているが、机と椅子は部屋の座標系からの位置で指定されている。壺は机の座標系からの位置で指定されている。これらの座標系間の関係を考慮してカメラ座標系に変換する。

【0155】

本発明の発明者は、特願平10-185135号において立体視パラメータをユーザの欲する時点で変更することのできるユーザインタフェースを提案した。

【0156】

第31図は、第6変形例に従った、ディスプレイ10の画面に表示されたユーザ・インターフェースを表す。

【0157】

同図において、200と201は基線長bを変更するためのユーザ・インターフェース（VIEW OFFSET）であり、どちらもスライドバーのコントロールである。スライドバー200は基線長を大きく変更するとき、スライドバー201は小さく変更するときに、それぞれ使用される。

【0158】

第31図に於いて、スライドバー202は前進方向の移動速度（FWD SPEED）を変更するときに用いる。スライドバー203は後退方向の移動速度（BWD SPEED）を変更するときに用いる。

【0159】

また、スライドバー204は輻輳角（View Congestion）を変更するときに用いる。更にスライドバー205は視野角（Field of View）を変更するときに用いる。

【0160】

第16図に示されたユーザ・インターフェースはユーザがマウスを用いて操作する。マウスの移動量が操作量となる。

【0161】

尚、ユーザ・インターフェースの具体的なツールはスライドバーに限定されない。連続的に変更できるツール、例えば、ジョグダイヤルであつてもよい。

【0162】

尚、前述の実施形態は、立体視パラメータの値（初期値及びリアルタイム値）を自動設定することが特徴であるから、第31図のユーザインタフェースをさらに設けると競合する。そこで、第31図の200から205までの6つのバーのいずれかがユーザにより操作されたことが検知されると、前述の立体視パラメータの設定処理（第20図）を停止するように制御手順を変更するとよい。そして、自動設定を再開したいときには、同じくディ

10

20

30

40

50

スプレイ 10 に表示されているキーアイコン 206 をユーザが押したことを検知したならば、再開される。

【0163】

以上説明した第 1 变形例乃至第 6 变形例によれば、立体視パラメータの再計算が、ユーザアプリケーションの様に応じて、或いはユーザの希望に応じて、最適なタイミングで行われるので、使い勝手がよい。

【0164】

また、仮想カメラの視軸において輻輳角が存在する場合も考慮されているので、本発明の立体視パラメータの自動設定は適用分野は広いものとなる。

【0165】

また、立体視パラメータの再設定は時間のかかる処理であるが、レンダリングの処理を、立体視パラメータの設定と立体画像の生成とに移用することによって処理負担を軽減することができた。

【0166】

その他の変形例
上記実施形態や変形例では、一般的な仮想現実感システムに本発明の立体視パラメータ設定の自動処理を適用したものであったが、所謂複合現実感システムに本発明を適用することも可能である。

【0167】

複合現実感システムでは HMD (頭部装着型表示装置) を用いることが多いが、この場合は、表示画面とユーザの視点までの距離が固定であるので、測距センサ 17 は不要である。

【0168】

【発明の効果】

以上説明したように、本発明によれば、立体視パラメータの最適な初期値を自動的に設定することができる。

【0169】

また、立体視パラメータの最適リアルタイム値を自動的に更新することができる。

【0170】

また、立体視パラメータの再設定タイミングを自動的に或いはユーザの好みにあわせて設定することができる。

【図面の簡単な説明】

【図 1】本発明の実施形態に係る立体画像表示装置 1000 の構成を示す図。

【図 2】表示装置 1000 に用いられているリアクロスレンチ方式眼鏡なし 3D ディスプレイ 10 の構成を説明する図。

【図 3】ディスプレイ 10 の LCD 部分を拡大した図。

【図 4】実施形態の立体視パラメータの設定を仮想空間ウォークスルーシステムに適用した事例を説明する図。

【図 5】立体視を得るために左目用画像と右目用画像の合成原理を説明する図。

【図 6】最大視差量と基線長の関係を説明する図。

【図 7】基線長を決定原理を説明するために、2つの左右画像が生成される様子を説明する図。

【図 8】仮想空間の最近点が、2つの視点において視差として生じるかを説明する図。

【図 9】本実施形態の基線長を決定する原理を説明する図。

【図 10】本実施形態の基線長を決定する原理を説明する図。

【図 11】透視変換を説明する図。

【図 12】二次元空間でのサンプリングを説明する図。

【図 13】最近点の探索のためのサンプリングパターンの一例を示す図。

【図 14】最近点の探索のためのサンプリングパターンの他の例を示す図。

【図 15】最近点の探索のためのサンプリングパターンの他の例を示す図。

【図 16】最近点の探索のためのサンプリングパターンの他の例を示す図。

【図 17】最近点の探索のためのサンプリングパターンの他の例を示す図。

【図 18】最近点の探索のためのサンプリングパターンの他の例を示す図。

【図 19】最近点決定のための制御手順を説明するフローチャート。

【図 20】立体視パラメータの設定のための制御手順を説明するフローチャート。

【図 21】基線長決定のための制御手順を説明するフローチャート。

【図 22】ウォークスルーアプリケーションにおける全体制御手順を説明するフローチャート。

【図 23】立体視パラメータとしての輻輳角を決定する原理を説明する図。

【図 24】立体視パラメータとしての輻輳角を決定する原理を説明する図。 10

【図 25】輻輳角決定のための制御手順を説明するフローチャート。

【図 26】輻輳角が存在する場合における基線長を決定する原理を説明する図。

【図 27】ウォークスルーアプリケーションの変形例に係る全体制御手順を説明するフローチャート。

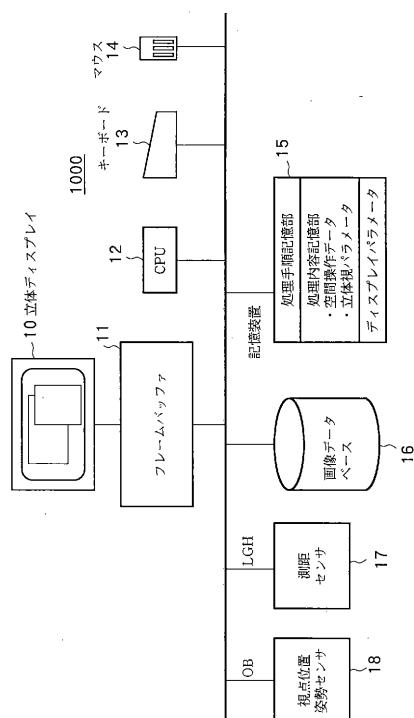
【図 28】ウォークスルーアプリケーションの他の変形例に係る全体制御手順を説明するフローチャート。

【図 29】ウォークスルーアプリケーションの更に他の変形例に係る全体制御手順を説明するフローチャート。

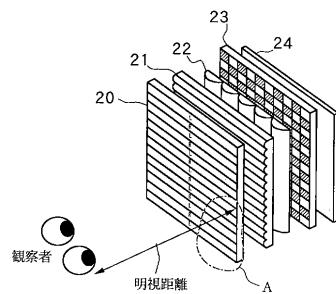
【図 30】最近点を決定する他の手法を説明する図。

【図 31】立体視パラメータの値をユーザが自由に設定するためのユーザインターフェース 20 を説明する図。

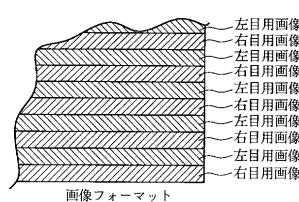
【図 1】



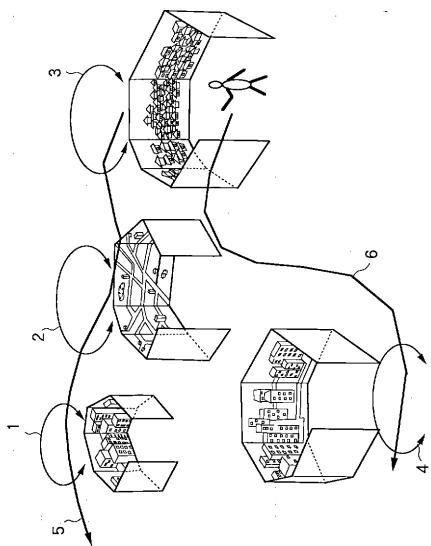
【図 2】



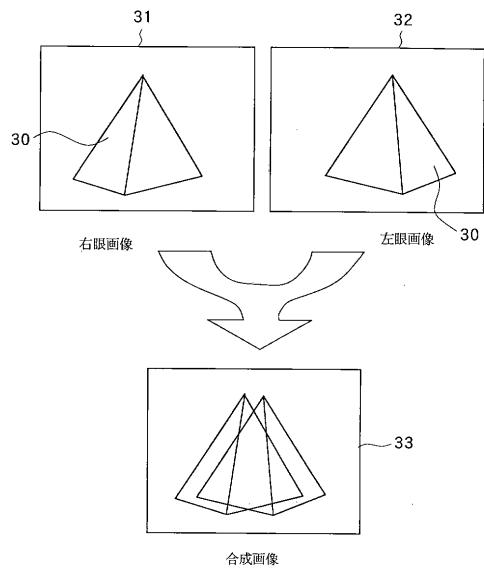
【図 3】



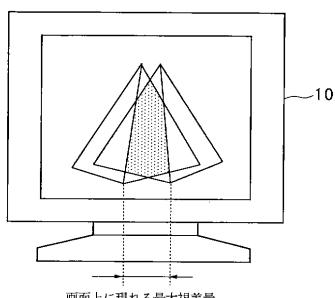
【図4】



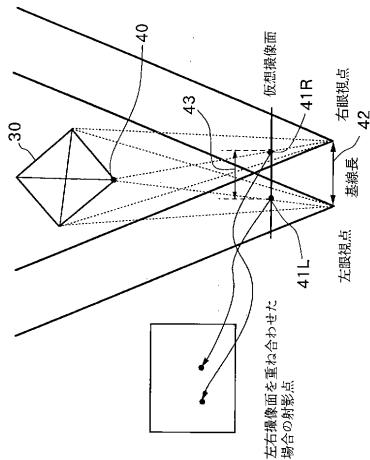
【図5】



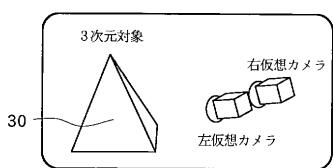
【図6】



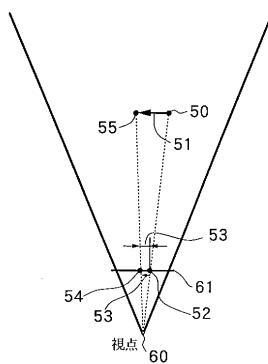
【図8】



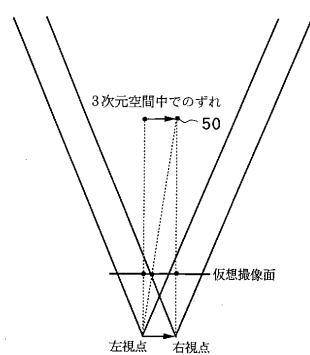
【図7】



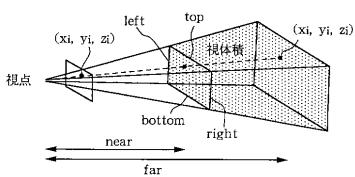
【図 9】



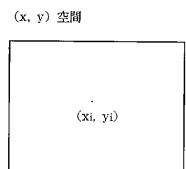
【図 10】



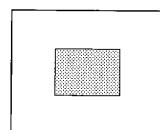
【図 11】



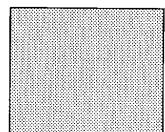
【図 12】



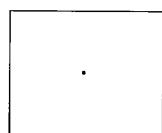
【図 14】



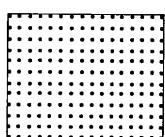
【図 13】



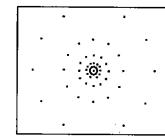
【図 15】



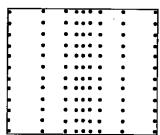
【図16】



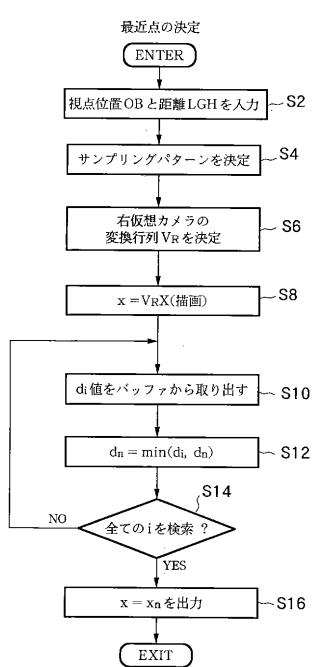
【図18】



【図17】

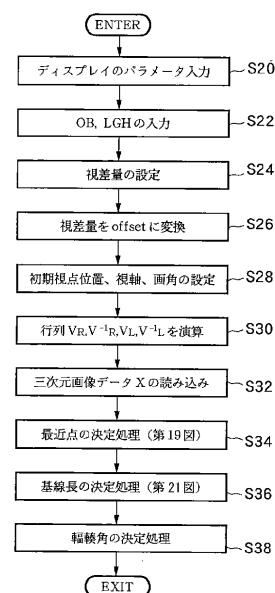


【図19】

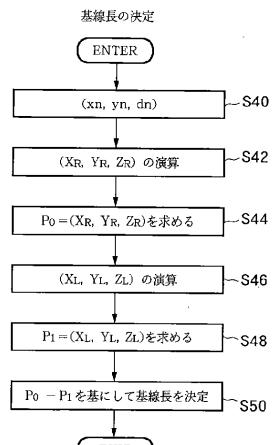


【図20】

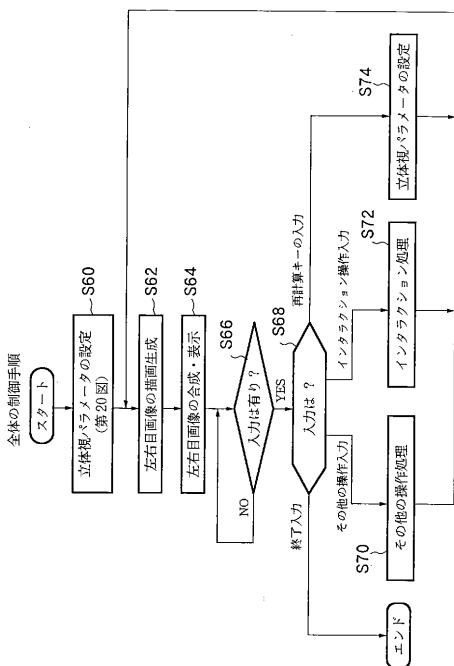
立体視パラメータの設定



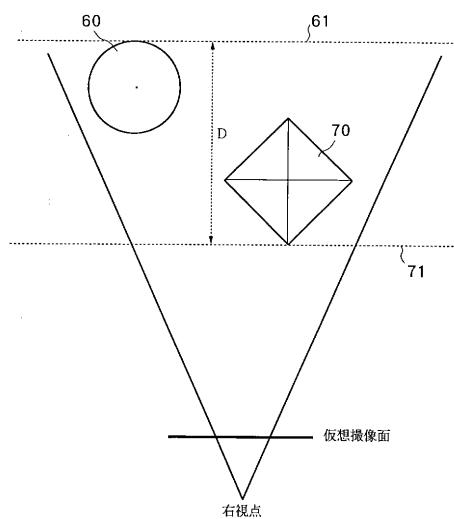
【図21】



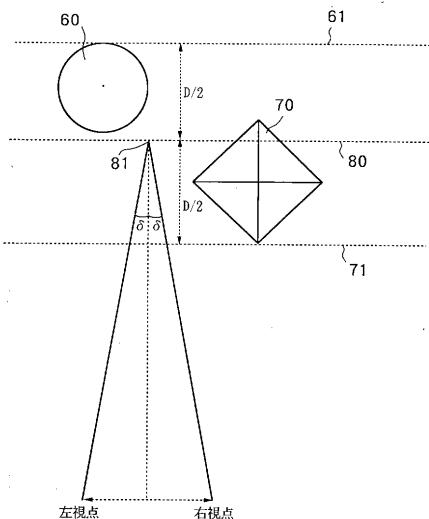
【図22】



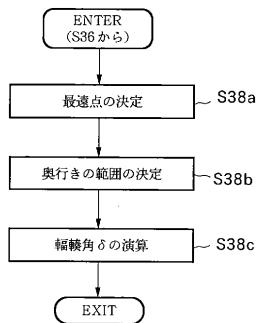
【図23】



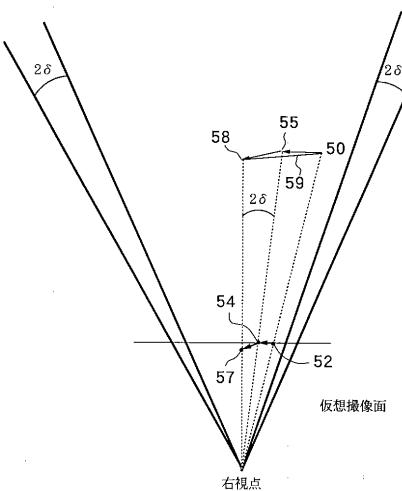
【図24】



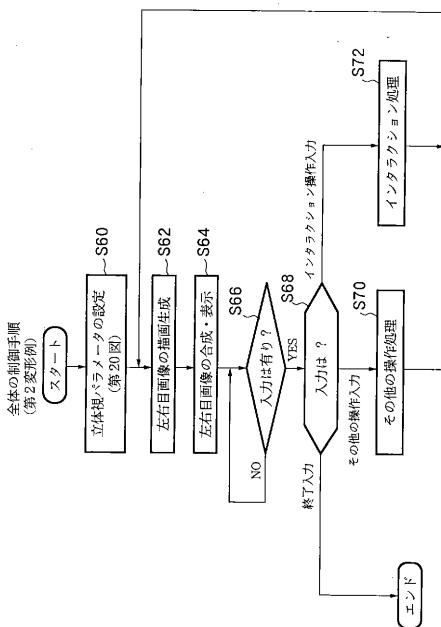
【図25】



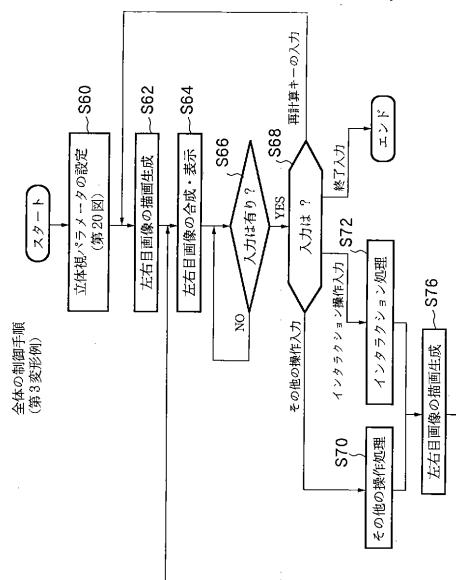
【図26】



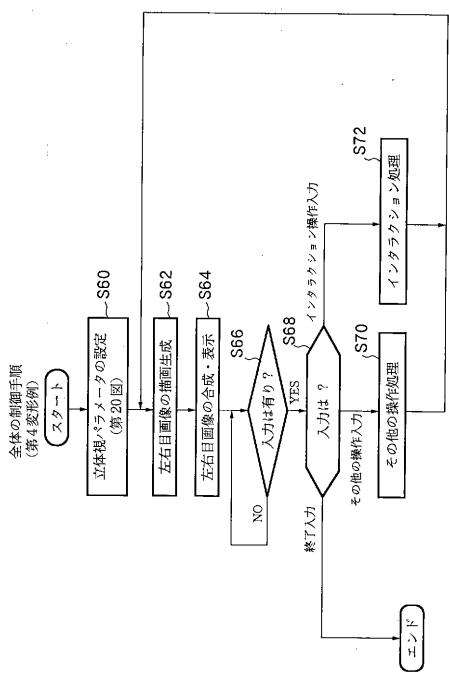
【図27】



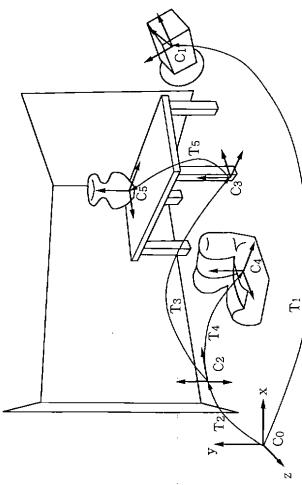
【図28】



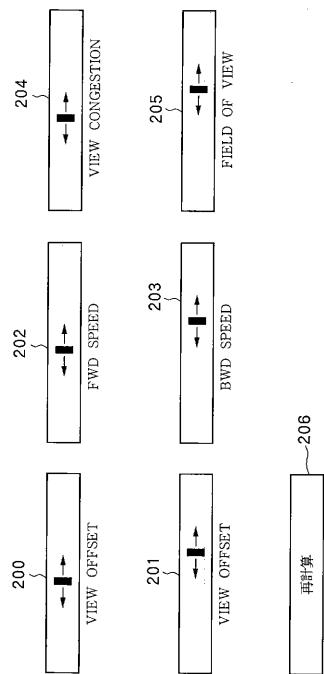
【図29】



【図30】



【図31】



フロントページの続き

審査官 村松 貴士

(56)参考文献 特開平10-188034(JP,A)
特開平11-338457(JP,A)
特開2000-20757(JP,A)

(58)調査した分野(Int.Cl.⁷, DB名)

G06T 17/40
H04N 13/00 - 17/06