(19)日本国特許庁(JP)　(12)公開特許公報(A)　(11)特許出願公開番号
特開2009-43245
(P2009-43245A)
(43)公開日　平成21年2月26日(2009.2.26)

(51)Int.Cl.		F I			テーマコード（参考）
G06T 15/00	(2006.01)	G06T	15/00	100A	5B050
G06T 15/70	(2006.01)	G06T	15/70	A	5B080

審査請求 未請求　請求項の数 22　ＯＬ　（全 26 頁）

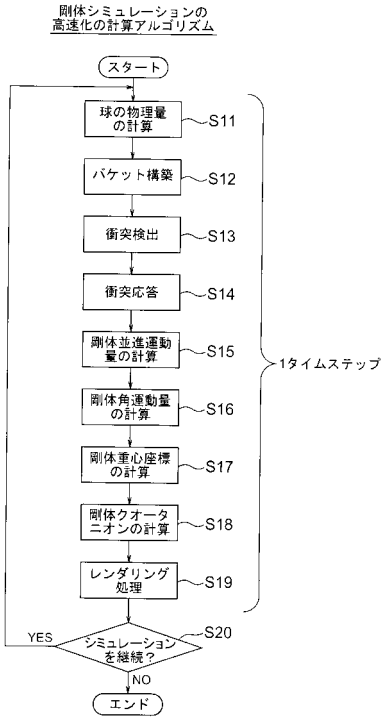
(21)出願番号　特願2008-184478 (P2008-184478)
(22)出願日　平成20年7月16日 (2008.7.16)
(31)優先権主張番号　特願2007-185978 (P2007-185978)
(32)優先日　平成19年7月17日 (2007.7.17)
(33)優先権主張国　日本国 (JP)

(71)出願人　505011534
プロメテック・ソフトウェア株式会社
東京都文京区本郷７丁目３番１号　東京大学アントレプレナープラザ３階
(74)代理人　100094020
弁理士　田宮　寛祉
(72)発明者　原田　隆宏
東京都文京区本郷七丁目３番１号　東京大学アントレプレナープラザ３階　プロメテック・ソフトウェア株式会社内

Ｆターム(参考)　5B050 AA06 BA08 BA18 CA02 EA28 EA30 FA02 FA05 FA08
5B080 AA17 CA01 CA03 DA06 FA01 FA02 FA03 GA02

(54)【発明の名称】近傍粒子探索に用いるデータ構造の構築方法、そのプログラム、およびそのプログラムを格納した記憶媒体

(57)【要約】
【課題】ＧＰＵ等上で粒子法シミュレーションの近傍粒子探索に用いるデータ構造の構築を完結させ、ＧＰＵ等を効率的に利用する近傍粒子探索に用いるデータ構造の構築方法、そのプログラム、およびそのプログラムを格納した記憶媒体を提供する。
【解決手段】近傍粒子探索のデータ構造の構築方法は、ＧＰＵ１２等のストリーミングプロセッサで実行され、３次元デジタル画像データを２次元デジタル画像データに変換することにより物理的対象物のシミュレーションを演算する。ＧＰＵを用いる場合にはバケットテクスチャを用い、このバケットテクスチャは、物理的対象物が生じる空間に対応して作られたメモリ空間を分割するバケットによって構成される。ＧＰＵの場合では、バーテックスシェーダを用いて、バケットに格納される粒子の前記粒子番号と粒子座標を読み出し、２次元画像空間に書き込む書き込みステップを有する。
【選択図】図５



【特許請求の範囲】
【請求項１】
ＧＰＵで実行され、ビデオメモリの内部で作られた画像データから構築される３次元デジタル画像データを２次元デジタル画像データに変換することにより物理的対象物のシミュレーションを演算する方法で用いられる、近傍粒子探索に用いるデータ構造の構築方法であり、
前記物理的対象物を多数の粒子で表現し、前記ビデオメモリ内における３次元画像データ構造での前記多数の粒子の相互の格納位置関係に基づいて前記物理的対象物の存在状態を定める手法が用いられ、
前記３次元画像データ構造に含まれるバケットテクスチャは、前記物理的対象物が生じる空間に対応して作られたメモリ空間を分割するバケットによって構成され、前記多数の粒子に割り付けられた粒子番号を多数の前記バケットの各々に格納し、
前記ＧＰＵで、当該ＧＰＵのバーテックスシェーダを用いて、前記バケットに格納される前記粒子の前記粒子番号と粒子座標を読み出し、２次元画像空間に書き込む書き込みステップを有し、
前記書き込みステップで、前記バケットに複数個の前記粒子が存在するときに、シミュレーションの１タイムステップで、前記複数個の粒子の各々の粒子番号と粒子座標を、複数回の処理のためのレンダリングに分けて１つずつ書き込むことを特徴とする近傍粒子探索に用いるデータ構造の構築方法。
【請求項２】
前記複数回の処理のためのレンダリングの各々で書き込む前記粒子の粒子番号と書き込むピクセルの選択、および書き込まれるピクセルにおけるチャンネルの選択は、前記ＧＰＵに用意された深度テスト、カラーマスク、ステンシルテストの各機能を用いて行われることを特徴とする請求項１記載の近傍粒子探索に用いるデータ構造の構築方法。
【請求項３】
前記バケットに４個の前記粒子が存在するとき、粒子番号が書き込まれるピクセルのＲＧＢＡから成る４つのチャンネルの各々に個別に昇順または降順に従う粒子番号の粒子からその粒子番号が順次に格納されることを特徴とする請求項２記載の近傍粒子探索に用いるデータ構造の構築方法。
【請求項４】
４個の前記粒子の粒子番号をi_{b0}，i_{b1}，i_{b2}，i_{b3}（昇順の場合には$i_{b0}<i_{b1}<i_{b2}<i_{b3}$、降順の場合には$i_{b0}>i_{b1}>i_{b2}>i_{b3}$）であるとするとき、
１パス目での前記ピクセルへの書き込みでは、深度バッファを最大値で初期化し、深度テストを用いて小さい値を持つものを合格させることにより、前記Ｒチャンネルに粒子番号i_{b0}を書き込んで１回目の処理のためのレンダリングを行うステップと、
２パス目での前記ピクセルへの書き込みでは、カラーマスクを用いて前記Ｒチャンネルへの書き込みを禁止しかつステンシルテストを用いることにより、前記Ｇチャンネルに粒子番号i_{b1}を書き込んで２回目の処理のためのレンダリングを行うステップと、
３パス目での前記ピクセルへの書き込みでは、カラーマスクを用いて前記ＲとＧの各チャンネルへの書き込みを禁止しかつステンシルテストを用いることにより、前記Ｂチャンネルに粒子番号i_{b2}を書き込んで３回目の処理のためのレンダリングを行うステップと、
４パス目での前記ピクセルへの書き込みでは、カラーマスクを用いて前記ＲとＧとＢの各チャンネルへの書き込みを禁止しかつステンシルテストを用いることにより、前記Ａチャンネルに粒子番号i_{b3}を書き込んで４回目の処理のためのレンダリングを行うステップと、
を有することを特徴とする請求項３記載の近傍粒子探索に用いるデータ構造の構築方法。
【請求項５】

前記物理的対象物は多数の剛体であり、前記多数の剛体の衝突計算に基づくシミュレーションの演算が実行されることを特徴とする請求項１記載の近傍粒子探索に用いるデータ構造の構築方法。

【請求項６】

ＧＰＵで実行され、ビデオメモリの内部で作られた画像データから構築される３次元デジタル画像データを２次元デジタル画像データに変換することにより物理的対象物のシミュレーションを演算するプログラムで用いられる、近傍粒子探索に用いるデータ構造の構築プログラムであり、

前記ＧＰＵに、

前記物理的対象物を多数の粒子で表現し、前記ビデオメモリ内における３次元画像データ構造での前記多数の粒子の相互の格納位置関係に基づいて前記物理的対象物の存在状態を定める手順と、

前記３次元画像データ構造に含まれるバケットテクスチャは、前記物理的対象物が生じる空間に対応して作られたメモリ空間を分割するバケットによって構成され、前記多数の粒子に割り付けられた粒子番号を多数の前記バケットの各々に格納する手順と、

バーテックスシェーダを用いて、前記バケットに格納される前記粒子の粒子番号と粒子座標を読み出し、２次元画像空間に書き込む書き込み手順とを実行させ、

前記書き込み手順で、前記バケットに複数個の前記粒子が存在するときに、シミュレーションの１タイムステップで、前記複数個の粒子の各々の粒子番号と粒子座標を、複数回の処理のためのレンダリングに分けて１つずつ書き込む手順を実行させることを特徴とする近傍粒子探索に用いるデータ構造の構築プログラム。

【請求項７】

前記複数回の処理のためのレンダリングの各々で書き込む前記粒子の粒子番号と書き込むピクセルの選択、および書き込まれるピクセルにおけるチャンネルの選択は、前記ＧＰＵに用意された深度テスト、カラーマスク、ステンシルテストの各機能を用いて行われることを特徴とする請求項６記載の近傍粒子探索に用いるデータ構造の構築プログラム。

【請求項８】

前記ＧＰＵに、

前記バケットに４個の前記粒子が存在するとき、粒子番号が書き込まれるピクセルのＲＧＢＡから成る４つのチャンネルの各々に個別に昇順または降順に従う粒子番号の粒子からその粒子番号が順次に格納される手順を実行させることを特徴とする請求項７記載の近傍粒子探索に用いるデータ構造の構築プログラム。

【請求項９】

前記ＧＰＵに、

４個の前記粒子の粒子番号をi_{b0}，i_{b1}，i_{b2}，i_{b3}（昇順の場合には$i_{b0}<i_{b1}<i_{b2}<i_{b3}$、降順の場合には$i_{b0}>i_{b1}>i_{b2}>i_{b3}$）であるとするとき、

１パス目での前記ピクセルへの書き込みでは、深度バッファを最大値で初期化し、深度テストを用いて小さい値を持つものを合格させることにより、前記Ｒチャンネルに粒子番号i_{b0}を書き込んで１回目の処理のためのレンダリングを行う手順と、

２パス目での前記ピクセルへの書き込みでは、カラーマスクを用いて前記Ｒチャンネルへの書き込みを禁止しかつステンシルテストを用いることにより、前記Ｇチャンネルに粒子番号i_{b1}を書き込んで２回目の処理のためのレンダリングを行う手順と、

３パス目での前記ピクセルへの書き込みでは、カラーマスクを用いて前記ＲとＧの各チャンネルへの書き込みを禁止しかつステンシルテストを用いることにより、前記Ｂチャンネルに粒子番号i_{b2}を書き込んで３回目の処理のためのレンダリングを行う手順と、

４パス目での前記ピクセルへの書き込みでは、カラーマスクを用いて前記ＲとＧとＢの各チャンネルへの書き込みを禁止しかつステンシルテストを用いることにより、前記Ａチャンネルに粒子番号i_{b3}を書き込んで４回目の処理のためのレンダリングを行う手順と、

を実行させることを特徴とする請求項８記載の近傍粒子探索に用いるデータ構造の構築プログラム。

【請求項１０】

前記物理対象物は多数の剛体であり、

前記ＧＰＵに、前記多数の剛体の衝突計算に基づくシミュレーションの演算を行わせる手順を実行させることを特徴とする請求項６記載の近傍粒子探索に用いるデータ構造の構築プログラム。

【請求項１１】

請求項６～１０のいずか１項に記載された近傍粒子探索に用いるデータ構造を構築するためのプログラムを記録したことを特徴とするコンピュータ読取り可能な記録媒体。

【請求項１２】

ストリーミングプロセッサで実行され、メモリの内部で作られたデータアレイから構築される３次元デジタル画像データを２次元デジタル画像データに変換することにより物理的対象物のシミュレーションを演算する方法で用いられる、近傍粒子探索に用いるデータ構造の構築方法であり、

前記物理的対象物を多数の粒子で表現し、前記メモリ内における３次元画像データ構造での前記多数の粒子の相互の格納位置関係に基づいて前記物理的対象物の存在状態を定める手法が用いられ、

前記３次元画像データ構造に含まれるバケットアレイは、前記物理的対象物が生じる空間に対応して作られたメモリ空間を分割するバケットによって構成され、前記多数の粒子に割り付けられた粒子番号を多数の前記バケットの各々に格納し、

前記ストリーミングプロセッサで、当該ストリーミングプロセッサの分散オペレーションを用いて、前記バケットに格納される前記粒子の前記粒子番号と粒子座標を読み出し、前記メモリにおけるアレイに書き込む書き込みステップを有し、

前記書き込みステップで、前記バケットに複数個の前記粒子が存在するときに、シミュレーションの１タイムステップで、前記複数個の粒子の各々の粒子番号と粒子座標を、複数回の処理のためのカーネルに分けて１つずつ書き込む近傍粒子探索に用いることを特徴とするデータ構造の構築方法。

【請求項１３】

前記複数回の処理のための各ステップで書き込む前記粒子の粒子番号と書き込むデータの選択、およびボクセルのために割り当てられたメモリにおける複数のメモリ箇所の選択は、前記ストリーミングプロセッサに用意されたカウンタの機能を用いて行われることを特徴とする請求項１２記載の近傍粒子探索に用いるデータ構造の構築方法。

【請求項１４】

前記バケットに４個の前記粒子が存在するとき、粒子番号が書き込まれる第１から第４の前記メモリ箇所の各々に個別に昇順または降順に従う粒子番号の粒子からその粒子番号が順次に格納される請求項１３の近傍粒子探索に用いるデータ構造の構築方法。

【請求項１５】

４個の前記粒子の粒子番号をi_{b0}，i_{b1}，i_{b2}，i_{b3}（昇順の場合には$i_{b0}<i_{b1}<i_{b2}<i_{b3}$、降順の場合には$i_{b0}>i_{b1}>i_{b2}>i_{b3}$）であるとするとき、

１パス目での前記ボクセルのために割り当てられた前記メモリへの書き込みでは、前記カウンタが０になるように初期化しかつ第１の前記メモリ箇所への最初の書き込みを受け入れることにより、前記第１のメモリ箇所に粒子番号i_{b0}を書き込んで１回目のルーチンのためのカーネルを実行するステップと、

２パス目での前記ボクセルのために割り当てられた前記メモリへの書き込みでは、前記カウンタを用いることにより、前記第１のメモリ箇所への書き込みを禁止しかつ第２の前記メモリ箇所へ次の値を書き込むことにより、前記第２のメモリ箇所に粒子番号i_{b1}を書き込んで２回目のルーチンのためのカーネルを実行するステップと、

３パス目での前記ボクセルのために割り当てられた前記メモリへの書き込みでは、前記

カウンタを用いることにより、前記第１と第２のメモリ箇所への書き込みを禁止しかつ第３の前記メモリ箇所へ次の値を書き込むことにより、前記第３のメモリ箇所に粒子番号i_{b2}を書き込んで３回目のルーチンのためのカーネルを実行するステップと、

４パス目での前記ボクセルのために割り当てられた前記メモリへの書き込みでは、前記カウンタを用いることにより、前記の第１と第２と第３のメモリ箇所への書き込みを禁止しかつ第４の前記メモリ箇所へ次の値を書き込むことにより、前記第４のメモリ箇所に粒子番号i_{b3}を書き込んで４回目のルーチンのためのカーネルを実行するステップと、

を有することを特徴とする請求項１３記載の近傍粒子探索に用いるデータ構造の構築方法。

【請求項１６】

前記物理的対象物は多数の剛体であり、前記多数の剛体の衝突計算に基づくシミュレーションの演算が実行されることを特徴とする請求項１２記載の近傍粒子探索に用いるデータ構造の構築方法。

【請求項１７】

ストリーミングプロセッサで実行され、メモリの内部で作られたデータアレイから構築される３次元デジタル画像データを２次元デジタル画像データに変換することにより物理的対象物のシミュレーションを演算するプログラムで用いられる、近傍粒子探索に用いるデータ構造の構築プログラムであり、

前記ストリーミングプロセッサに、

前記物理的対象物を多数の粒子で表現し、前記メモリ内における３次元画像データ構造での前記多数の粒子の相互の格納位置関係に基づいて前記物理的対象物の存在状態を定める手順と、

前記３次元画像データ構造に含まれるバケットアレイは、前記物理的対象物が生じる空間に対応して作られたメモリ空間を分割するバケットによって構成され、前記多数の粒子に割り付けられた粒子番号を多数の前記バケットの各々に格納する手順と、

前記ストリーミングプロセッサで、当該ストリーミングプロセッサの分散オペレーションを用いて、前記バケットに格納される前記粒子の前記粒子番号と粒子座標を読み出し、前記メモリにおけるアレイに書き込む書き込み手順とを有し、

前記書き込みステップで、前記バケットに複数個の前記粒子が存在するときに、シミュレーションの１タイムステップで、前記複数個の粒子の各々の粒子番号と粒子座標を、複数回の処理のためのカーネルに分けて１つずつ書き込む手順を有することを特徴とする近傍粒子探索に用いるデータ構造の構築プログラム。

【請求項１８】

前記複数回の処理のための各ステップで書き込む前記粒子の粒子番号と書き込むデータの選択、およびボクセルのために割り当てられたメモリにおける複数のメモリ箇所の選択は、前記ストリーミングプロセッサに用意されたカウンタの機能を用いて行われることを特徴とする請求項１７記載の近傍粒子探索に用いるデータ構造の構築プログラム。

【請求項１９】

前記バケットに４個の前記粒子が存在するとき、粒子番号が書き込まれる第１から第４の前記メモリ箇所の各々に個別に昇順または降順に従う粒子番号の粒子からその粒子番号が順次に格納される請求項１８記載の近傍粒子探索に用いるデータ構造の構築プログラム。

【請求項２０】

前記ストリーミングプロセッサに、

４個の前記粒子の粒子番号をi_{b0}，i_{b1}，i_{b2}，i_{b3}（昇順の場合には$i_{b0}<i_{b1}<i_{b2}<i_{b3}$、降順の場合には$i_{b0}>i_{b1}>i_{b2}>i_{b3}$）であるとするとき、

１パス目での前記ボクセルのために割り当てられた前記メモリへの書き込みでは、前記カウンタが０になるように初期化しかつ第１の前記メモリ箇所への最初の書き込みを受け入れることにより、前記第１のメモリ箇所に粒子番号i_{b0}を書き込んで１回目のルーチ

ンのためのカーネルを実行する手順と、
２パス目での前記ボクセルのために割り当てられた前記メモリへの書き込みでは、前記カウンタを用いることにより、前記第１のメモリ箇所への書き込みを禁止しかつ第２の前記メモリ箇所へ次の値を書き込むことにより、前記第２のメモリ箇所に粒子番号i_{b1}を書き込んで２回目のルーチンのためのカーネルを実行する手順と、
３パス目での前記ボクセルのために割り当てられた前記メモリへの書き込みでは、前記カウンタを用いることにより、前記第１と第２のメモリ箇所への書き込みを禁止しかつ第３の前記メモリ箇所へ次の値を書き込むことにより、前記第３のメモリ箇所に粒子番号i_{b2}を書き込んで３回目のルーチンのためのカーネルを実行する手順と、
４パス目での前記ボクセルのために割り当てられた前記メモリへの書き込みでは、前記カウンタを用いることにより、前記の第１と第２と第３のメモリ箇所への書き込みを禁止しかつ第４の前記メモリ箇所へ次の値を書き込むことにより、前記第４のメモリ箇所に粒子番号i_{b3}を書き込んで４回目のルーチンのためのカーネルを実行する手順と、
を実行させることを特徴とする請求項１９記載の近傍粒子探索に用いるデータ構造の構築プログラム。
【請求項２１】
前記物理対象物は多数の剛体であり、
前記ストリーミングプロセッサに、前記多数の剛体の衝突計算に基づくシミュレーションの演算を行わせる手順を実行させることを特徴とする請求項２０記載の近傍粒子探索に用いるデータ構造の構築プログラム。
【請求項２２】
請求項１７～２１のいずか１項に記載された近傍粒子探索に用いるデータ構造を構築するためのプログラムを記録したことを特徴とするコンピュータ読取り可能な記録媒体。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、粒子法シミュレーションをＧＰＵ等のストリーミングプロセッサ（Streaming processor）で完結的に実行することにより近傍粒子探索を高速化する手法を実現し、例えば多数の剛体の衝突計算等をリアルタイムで計算可能にした近傍粒子探索に用いるデータ構造の構築方法、そのプログラム、およびそのプログラムを格納した記憶媒体に関する。
【背景技術】
【０００２】
「粒子法シミュレーション」とは、対象とする物体を多数の粒子の集まり（集合）としてとらえ、多数の粒子の個々の挙動に注目して計算を行うシミュレーション手法である。コンピュータでの計算上、上記粒子には、通常、直径が一定である「球」が用いられる。物体の形状を球の集合体によって表現するとき、当該球の大きさ（直径）の決め方に依存して物体の形状の近似度または空間解像度が変化する。また、球の直径を変化させ、空間解像度を変化させることにより、コンピュータによるシミュレーション計算における計算精度と計算速度を制御することが可能となる。
【０００３】
近年、上記の粒子法シミュレーションを利用して、多数の剛体の衝突計算、流体や粉体等の挙動計算が行われ、コンピュータの表示装置の画面上で衝突や流動等のシミュレーションの研究が行われている。この研究結果は、例えば、非特許文献１に記載されている。これらのシミュレーション技術の研究は、コンピュータグラフィックス（ＣＧ）の基礎となる技術として重要である。
【０００４】
上記の粒子法シミュレーションの研究では、従来、ＣＰＵ（Central Processing Unit）を利用して計算アルゴリズムを作る方法は多く行われていた。しかし、ＧＰＵ（Graphics Processing Unit）等のストリーミングプロセッサ（Streaming processor）を積極的

に利用した粒子法シミュレーションの研究はあまり行われていなかった。例えばＧＰＵを利用した従来の或る研究によれば、近傍粒子探索をＣＰＵで実行し、ＣＰＵで得た探索データをＧＰＵに転送して画像処理するようにしていた。この技術によれば、ＣＰＵを主たる計算実行部として使用して当該ＣＰＵで近傍粒子探索の計算アルゴリズムを実行させ、ＧＰＵは従たる計算実行部として画像表示処理の計算アルゴリズムだけを実行させるように構成されていた。従って、従来の計算技術によれば、粒子法シミュレーションにおける近傍粒子探索および画像処理の計算アルゴリズムを、ＧＰＵの内部だけで完結させて実行させるようには構成できてはいなかった。

【非特許文献１】田中正幸等、「粒子法を用いた剛体計算手法の開発とコンピュータ・グラフィックスへの適用」、日本機械学会、第１９回計算力学講演会講演論文集、７０１－７０２頁、２００６。

【発明の開示】

【発明が解決しようとする課題】

【０００５】

従来の粒子法シミュレーションの計算ではすべての計算（近傍粒子探索に用いるデータ構造の構築、画像処理等）がＧＰＵ上のみで完結させるように構成されていなかったため、計算効率が良好ではなく、ＧＰＵが本来的に有している画像処理能力を十分に発揮することができず、多数の剛体の衝突計算等のリアルタイムな剛体シミュレーションや、流体シミュレーション等を行うことができなかった。そこで、ＧＰＵやその他のストリーミングプロセッサを完結的に利用して、ＧＰＵ等の上で近傍粒子探索に用いるデータ構造の構築を実行し、その画像処理能力を最大限に発揮させる手法の開発が求められている。

【０００６】

本発明の目的は、上記の課題に鑑み、ＧＰＵ等のストリーミングプロセッサ上で粒子法シミュレーションの近傍粒子探索に用いるデータ構造の構築を完結させ、ＧＰＵ等を効率的に利用し、その最大限の（画像）処理能力を発揮させ、高速計算を可能にし、例えばリアルタイムの剛体シミュレーションや流体シミュレーション等の計算処理／表示処理を実現することができる近傍粒子探索に用いるデータ構造の構築方法、そのプログラム、およびそのプログラムを格納した記憶媒体を提供することにある。

【課題を解決するための手段】

【０００７】

本発明に係る近傍粒子探索に用いるデータ構造の構築方法、そのプログラム、およびそのプログラムを格納した記憶媒体は、上記の目的を達成するために、次のように構成される。

【０００８】

本発明に係るデータ構造の構築方法は、ＧＰＵ等のごときグラフィックス演算プロセッサで実行され、ビデオメモリの内部で作られた画像データから構築される３次元デジタル画像データを２次元デジタル画像データに変換することにより物理的対象物のシミュレーションを演算する方法で用いられ、物理的対象物を多数の粒子で表現し、ビデオメモリ内における３次元画像データ構造での多数の粒子の相互の格納位置関係に基づいて物理的対象物の存在状態を定める手法が用いられる。３次元画像データ構造に含まれるバケットテクスチャは、物理的対象物が生じる空間に対応して作られたメモリ空間を分割するバケットによって構成され、多数の粒子に割り付けられた粒子番号を多数のバケットの各々に格納するものであり、グラフィックス演算プロセッサで、バーテックスシェーダを用いて、バケットに格納される粒子の前記粒子番号と粒子座標を読み出し、２次元画像空間に書き込む書き込みステップを有する。この書き込みステップでは、バケットに複数個の粒子が存在するときに、シミュレーションの１タイムステップで、複数個の粒子の各々の粒子番号と粒子座標を、複数回の書き込み（処理のためのレンダリング）に分けて１つずつ書き込むことで特徴づけられる。

【０００９】

上記のデータ構造の構築方法において、グラフィックス演算プロセッサとしてＧＰＵを

用いる場合には、複数回の処理のためのレンダリングの各々で書き込む粒子の粒子番号と粒子座標の選択、および書き込まれるピクセルにおけるチャンネルの選択は、ＧＰＵに用意された深度テスト、カラーマスク、ステンシルテストの各機能を用いて行われる。

また、ＧＰＵ等を含むストリーミングプロセッサを用いる場合には、複数回の処理サイクルの各ステップで書き込む粒子の粒子番号とメモリ箇所（Memory Location）の選択、およびボクセル（Voxel）のために割り当てられたメモリにおけるメモリ箇所の選択は、ストリーミングプロセッサに用意された複数のカウンタの各機能を用いて行うこともできる。

すなわち、ストリーミングプロセッサで実行され、メモリの内部で作られたデータアレイから構築される３次元デジタル画像データを２次元デジタル画像データに変換することにより物理的対象物のシミュレーションを演算する方法で用いられる、近傍粒子探索に用いるデータ構造の構築方法である。物理的対象物を多数の粒子で表現し、メモリ内における３次元画像データ構造での多数の粒子の相互の格納位置関係に基づいて物理的対象物の存在状態を定める手法が用いられ、３次元画像データ構造に含まれるバケットアレイは、物理的対象物が生じる空間に対応して作られたメモリ空間を分割するバケットによって構成され、多数の粒子に割り付けられた粒子番号を多数の前記バケットの各々に格納する。ストリーミングプロセッサで、バケットに格納される粒子の粒子番号と粒子座標を読み出し、メモリにおけるアレイに書き込む書き込みステップを有し、書き込みステップで、バケットに複数個の粒子が存在するときに、シミュレーションの１タイムステップで、複数個の粒子の各々の粒子番号と粒子座標を、複数回の処理のためのカーネル（a kernel）に分けて１つずつ書き込む。

【００１０】

データ構造の構築方法は、上記の方法において、バケットに最大４個の粒子が存在するとき、粒子番号が書き込まれるピクセル等のＲＧＢＡから成る４つのチャンネル等の各々に個別に昇順または降順に従う粒子番号の粒子からその粒子番号が順次に格納される。

【００１１】

上記のデータ構造の構築方法において、グラフィックス演算プロセッサとしてＧＰＵを用いる場合には、より具体的に次のステップを有する。

４個の粒子の粒子番号をi_{b0}，i_{b1}，i_{b2}，i_{b3}（昇順の場合には$i_{b0}<i_{b1}<i_{b2}<i_{b3}$、降順の場合には$i_{b0}>i_{b1}>i_{b2}>i_{b3}$）であるとするとき、

１パス目でのピクセルへの書き込みでは、深度バッファを最大値で初期化し、深度テストを用いて小さい値を持つものを合格させることにより、Ｒチャンネルに粒子番号i_{b0}を書き込んで１回目の処理のためのレンダリングを行うステップと、

２パス目でのピクセル等への書き込みでは、カラーマスクを用いてＲチャンネルへの書き込みを禁止しかつステンシルテスト等を用いることにより、Ｇチャンネルに粒子番号i_{b1}を書き込んで２回目の処理のためのレンダリングを行うステップと、

３パス目でのピクセルへの書き込みでは、カラーマスクを用いてＲとＧの各チャンネルへの書き込みを禁止しかつステンシルテストを用いることにより、Ｂチャンネルに粒子番号i_{b2}を書き込んで３回目の処理のためのレンダリングを行うステップと、

４パス目でのピクセルへの書き込みでは、カラーマスクを用いてＲとＧとＢの各チャンネルへの書き込みを禁止しかつステンシルテストを用いることにより、Ａチャンネルに粒子番号i_{b3}を書き込んで４回目の処理のためのレンダリングを行うステップと、

を有する。

一般的にストリーミングプロセッサを用いる場合には、上記のピクセル（written pixel）の代わりにボクセルに割り当てられたメモリを用い、上記のチャンネルの代わりにボクセル（Voxel）のために割り当てられたメモリにおけるメモリ箇所（Memory Location）を用い、さらに、ＧＰＵに用意された上記の深度テスト、カラーマスク、ステンシルテストの各機能の代わりに、それぞれに対応するカウンタの機能を用いる。

すなわち、４個の粒子の粒子番号をi_{b0}，i_{b1}，i_{b2}，i_{b3}（昇順の場合には

$i_{b0} < i_{b1} < i_{b2} < i_{b3}$、降順の場合には$i_{b0} > i_{b1} > i_{b2} > i_{b3}$）であるとするとき、

１パス目でのボクセル（voxel）のために割り当てられたメモリへの書き込みでは、カウンタが０になるように初期化しかつ第１のメモリ箇所への最初の書き込みを受け入れることにより、第１のメモリ箇所に粒子番号i_{b0}を書き込んで１回目のルーチンのためのカーネルを実行するステップと、

２パス目でのボクセル（voxel）のために割り当てられたメモリへの書き込みでは、カウンタを用いることにより、第１のメモリ箇所への書き込みを禁止しかつ第２のメモリ箇所へ次の値を書き込むことにより、第２のメモリ箇所に粒子番号i_{b1}を書き込んで２回目のルーチンのためのカーネルを実行するステップと、

３パス目でのボクセル（voxel）のために割り当てられたメモリへの書き込みでは、カウンタを用いることにより、第１と第２のメモリ箇所への書き込みを禁止しかつ第３のメモリ箇所へ次の値を書き込むことにより、第３のメモリ箇所に粒子番号i_{b2}を書き込んで３回目のルーチンのためのカーネルを実行するステップと、

４パス目でのボクセル（voxel）のために割り当てられたメモリへの書き込みでは、カウンタを用いることにより、第１と第２と第３のメモリ箇所への書き込みを禁止しかつ第４のメモリ箇所へ次の値を書き込むことにより、第４のメモリ箇所に粒子番号i_{b3}を書き込んで４回目のルーチンのためのカーネルを実行するステップと、を有する。

【００１２】

データ構造の構築方法は、上記の方法において、好ましくは、上記物理対象物は多数の剛体であり、物理対象物のシミュレーション画像は多数の剛体の衝突計算に基づくシミュレーション画像である。

【００１３】

データ構造の構築方法に関して、ストリーミングプロセッサの一例である上記ＧＰＵはグラフィックス演算プロセッサである。

【００１４】

また本発明に係る前述した近傍粒子探索に用いるデータ構造の構築プログラムは、ＧＰＵ等を含むストリーミングプロセッサで、上述した近傍粒子探索方法を構成する各手順等を実行させるためのプログラムである。

【００１５】

本発明によれば次の効果を奏する。

粒子法シミュレーションを適用して剛体、流体、粉体等を近似し、近傍粒子探索に用いるデータ構造の構築方法を実施することによって物理的現象をコンピュータ上でシミュレーションとして行う方法において、近傍粒子探索に用いるデータ構造の構築計算を実行するための各手順をＧＰＵ等のストリーミングプロセッサの上で完結させるように構成し、ＧＰＵ等を効率的に利用し、その最大限の画像演算処理能力を発揮させることができ、高速計算を行うことができ、これにより例えばリアルタイムの剛体シミュレーションや流体シミュレーション等を実現することができる。

また、剛体シミュレーションや流体シミュレーション等をシミュレーションＣＧとして画像表示に利用することにより水の流れ等の自然物の表現に最適なゲーム等の重要要素技術として利用することができる。

【発明を実施するための最良の形態】

【００１６】

以下に、本発明の好適な実施形態（実施例）を添付図面に基づいて説明する。

【００１７】

図１は、粒子法シミュレーションで近傍粒子探索に用いるデータ構造の構築方法が実施されるコンピュータシステムの基本的な構成を示す。

この実施形態の説明では、主たる実施形態として、グラフィックス演算プロセッサとして例えばＧＰＵを用いる例を説明する。

【００１８】

コンピュータシステム１０は、例えばＰＣ（Personal Computer）を利用して構成され、演算処理部としてのＣＰＵ１１とＧＰＵ１２を備える。またコンピュータシステム１０は、記憶装置としてメインメモリ１３Ａとビデオメモリ（ＶＲＡＭ）１３Ｂを備え、周辺装置として少なくとも入力装置１４および表示装置１５を備えている。ＣＰＵ１１とＧＰＵ１２とメインメモリ１３Ａと入力装置１４と表示装置１５はバス１６を介して相互に接続されている。入力装置１４は入力インターフェース１７を介してバス１６に接続され、表示装置１５は出力インターフェース１８を介してバス１６に接続されている。またビデオメモリ１３Ｂは、ＧＰＵ１２に直接に接続されている。ビデオメモリ１３Ｂに対してはＧＰＵ１２を通してデータの行き来が行われる。

【００１９】

ＣＰＵ（中央演算処理装置）１１は、例えばＰＣ等に装備される通常的な演算処理部である。またＧＰＵ１２は「Graphics Processing Unit（グラフィックス演算プロセッサ）」であり、画像演算処理を実行する。当該ＧＰＵ１２は、複数のプロセッサを内蔵して成り、並列計算機として用いられる。ＧＰＵ１２としては、例えばＮＶＩＤＩＡ社製の「GeForce 7900GTX」が使用される。なお、ＧＰＵ１２で使用される製品はこれに限定されず、類似した構造・機能を有する画像処理用のプロセッサまたはその他のストリーミングプロセッサを用いることができる。このＧＰＵ１２は、メインメモリ１３Ａに格納された近傍粒子探索プログラム１３Ａ－１を実行し、粒子法シミュレーションにおける近傍粒子探索演算を実施する。近傍粒子探索プログラム１３Ａ－１は、近傍粒子探索に用いるデータ構造の構築を実行する。またＧＰＵ１２は、メインメモリ１３Ａに格納された通常的な画像処理プログラム１３Ａ－２を実行し、近傍粒子探索の計算で得られかつピクセルに格納された画像データを用いて画像処理を実施し、作成した画像データを用いて表示装置１５の画面に描画（表示のためのレンダリング）する。表示装置１５の画面には、粒子法シミュレーションの計算に基づいて、ＣＧ画像が表示される。

【００２０】

本実施形態の以下の説明では、粒子法シミュレーションに基づく多数の剛体の衝突計算を実行することに基づくリアルタイムの剛体シミュレーションの演算が説明される。このシミュレーションの演算では、ＧＰＵ１２のみで、近傍粒子探索に用いるためのデータ構造の構築手法が実行される。

【００２１】

次に前述のＧＰＵ１２内の要部の基本構成を模式的に図２に示す。ＧＰＵ１２は、並列的に配置された複数のプロセッサ２１を備え、これらのプロセッサによる並列計算処理に基づいて、表示（描画）のためのレンダリングが実行される。特に本実施形態に係る近傍粒子探索に用いるデータ構造の構築（バケット構築）の方法によれば、後述するように、１つのバケットに例えば４個の粒子が存在するとき、４個の粒子の粒子番号と粒子座標を、処理のためのレンダリングを４回繰り返して１つのピクセルのＲＧＢＡチャンネルの各々に書き込む。ＧＰＵ１２における当該バケット構築の処理によって画像演算処理の高速化が実現される。より具体的に、ＧＰＵ１２は、「GeForce 7900GTX」である場合、前段に並列的に配置された８個のバーテックス（頂点）シェーダ（ＶＳ）２２と後段において並列的に配置された２４個のフラグメントシェーダ（ＦＳ）２３とを有している。バーテックス（頂点）プロセッサ（２１）で実行されるプログラムがバーテックスシェーダ２２であり、フラグメントプロセッサ（２１）で実行されるのがフラグメントシェーダ２３である。

【００２２】

上記において、バーテックスプロセッサは、並列的に配置されたプロセッサであり、多数の頂点の座標変換（入力された頂点座標を、レンダリングされる画像の空間での座標に変換すること）を一斉に処理することに特化されたプロセッサである。ＧＰＵ２１は、頂点の座標変換を、ＣＰＵに比較して高速に計算することができる。バーテックスプロセッサの動作を指定するものがバーテックスシェーダである。

【００２３】

さらにＧＰＵ１２には、深度テスト機能部２４、カラーマスク機能部２５、ステンシルテスト機能部２６が備えられている。またブロック２７はシェーダ指令分配部である。

【００２４】

本発明に係る近傍粒子探索に用いるデータ構造の構築手法を実現するための近傍粒子探索プログラム１３Ａ－１は、上記ＧＰＵ１２の内部構造を利用してＧＰＵ１２のみで完結的に実行される。

【００２５】

次に多数の剛体が衝突する剛体シミュレーションの実施形態を説明する。この剛体シミュレーションの説明を通して近傍粒子探索プログラム１３Ａ－１の実行に基づいて実施される近傍粒子探索に用いるデータ構造の構築手法の内容を明らかにする。

【００２６】

多数の剛体の例としては、例えば、多数のトーラス部材、あるいは多数のチェスの駒である。本実施形態に係る剛体シミュレーションの結果として得られたＣＧの例を図３と図４に示す。図３は、例えば１０９２２個のトーラスを落下・衝突させた物理現象に関する剛体シミュレーションの衝突計算に基づく結果画像（ＣＧ）を示し、図４は、例えば１６３８４個のチェスの駒を落下・衝突させた物理現象に関する剛体シミュレーションの衝突計算に基づく結果画像（ＣＧ）を示している。

【００２７】

上記の多数の剛体の衝突計算において、多数の剛体の各々の形状は、すべて、粒子的手法（粒子法シミュレーション）を適用することにより、同一の大きさの球（粒子）の集合体として近似的に表現されて取り扱われる。球に基づく剛体の形状の近似は、当該剛体のポリゴンモデルに基づいて行われ、例えば本発明者等によって提案されたＧＰＵを用いたボクセル化手法（「Fast solid voxelization using graphics hardware」, Transactions of JSCES, page No. 20060023, 2006）を用いて生成される。なお、用いる球の大きさ（空間解像度）は計算精度と計算速度を考慮して適切に決定される。

【００２８】

各々の剛体が球の集合体として近似的に表現されて成る多数の剛体の衝突計算においては、（１）個々の剛体の計算、（２）剛体間の衝突の検出処理（剛体を構成する球の間の距離の計算）、および（３）衝突に基づく応答処理（球に働く力とトルクの計算）が実行される。以下、これらの計算の考え方について分説する。

【００２９】

（１）剛体の計算：

剛体の計算に必要とされる物理量は、（数式１）に示すように、重心の座標、並進速度、回転量を表すクオータニオン、角速度、慣性テンソルである。

【００３０】

【数１】

$\mathbf{x}$：重心の座標

$\mathbf{v}$：並進速度

$\mathbf{q}$：回転量を表わすクオータニオン

$\mathbf{w}$：角速度

$\mathbf{I}$：慣性テンソル

【００３１】
剛体の運動の計算は、並進運動と回転運動に分けて計算する。並進運動の計算の内容は（数２）に示され、回転運動の計算の内容は（数３）に示される。
【００３２】
【数２】

剛体に働く力**F**は、(1)式に示すように衝突によって生じる**Fc**と
それ以外の外力**Fe**に分解される。

$$\mathbf{F} = \mathbf{Fc} + \mathbf{Fe} \qquad (1)$$

剛体の並進運動量**P**の時間微分は剛体に働く力**F**を用いて
(2)式に示すように計算される。

$$\frac{d\mathbf{P}}{dt} = \mathbf{F} \qquad (2)$$

並進運動量**P**を用いて、$\mathbf{P}=M\mathbf{v}$ (M:剛体の質量)より
重心の速度**v**を計算する。重心の速度**v**を用いると、
剛体の重心位置**x**の時間微分は(3)式で示される。

$$\frac{d\mathbf{x}}{dt} = \mathbf{v} \qquad (3)$$

【００３３】

【数３】

剛体に働く$\mathbf{F}$は剛体の回転運動を生み、角運動量$\mathbf{L}$を変化させる。
角運動量$\mathbf{L}$の時間微分は(4)式で表現される。$\mathbf{F}$は剛体の重心から
力の作用点までのベクトルである。

$$\frac{d\mathbf{L}}{dt} = \mathbf{r} \times \mathbf{F} \qquad (4)$$

上記の角運動量$\mathbf{L}$を用いて角速度$\mathbf{w}$は(5)式で計算される。

$$\mathbf{w} = \mathbf{I}(t)^{-1}\mathbf{L} \qquad (5)$$

(5)式で$\mathbf{I}(t)^{-1}$は時刻tでの慣性テンソルの逆行列である。
$\mathbf{I}(t)^{-1}$は初期状態での剛体の慣性テンソル$\mathbf{I}(o)$の逆行列$\mathbf{I}(o)^{-1}$を
用いて(6)式で計算される。

$$\mathbf{I}(t)^{-1} = \mathbf{R}(t)\,\mathbf{I}(o)^{-1}\mathbf{R}(t)^{T} \qquad (6)$$

(6)式で$\mathbf{R}(t)$と$\mathbf{R}(t)^{T}$は時刻tでの回転行列とその転置行列である。
(5)式で計算された角速度$\mathbf{w}$を用いてクオータニオン$\mathbf{q}$を更新する。
角速度$\mathbf{w}$からクオータニオンの変化量$d\mathbf{q}$は(7)式で計算される。

$$d\mathbf{q} = [\cos\left(\frac{\theta}{2}\right),\ a\sin\left(\frac{\theta}{2}\right)] \qquad (7)$$

ここで、回転軸aと回転角θは(8)式と(9)式で計算される。

$$a = \frac{\mathbf{w}}{|\mathbf{w}|} \qquad (8)$$

$$\theta = |\mathbf{w}dt| \qquad (9)$$

(7)式の$d\mathbf{q}$と時刻tでの$\mathbf{q}(t)$を用いてdt後のクオータニオン
$\mathbf{q}(t+dt)$は(10)式で計算される。

$$\mathbf{q}(t+dt) = d\mathbf{q} \times \mathbf{q}(t) \qquad (10)$$

【００３４】

（２）衝突の検出処理：

剛体の衝突を検出するために剛体を構成する球の間の距離を計算する。球の間の距離が球の直径よりも小さければ、２つの球は衝突しているとみなす。多数の剛体の衝突の検出では、すべての剛体の各々を構成する球に関して上記の距離計算を行うことが必要となる。計算量は球の数の２乗に比例し、剛体の数が多くなると球の数も多くなり、計算量も膨

大になる。そこで、本実施形態による衝突検出処理の構成では、計算量を減らすために、多数の剛体が存在することになる仮想的に空間（メモリにおける３次元のデータ格納空間）を「バケット（またはグリッドともいう）」と呼ばれる格子（３次元空間格子）に分割して、多数の剛体の各々を表現する多数の球をバケットに格納する。１つのバケットは１つの立方体を成し、仮にバケットの一辺の長さを球の直径と同じ長さである３次元データ構造を用いる場合には、或る球ｉと衝突している可能性のある球は、球ｉが格納されているバケットと隣接している3^3個の各バケットの内部に格納されている球に限定されることになる。上記のごとくバケットを構築することに基づいて衝突検出を行うことにより、計算量を減らし、計算コストを大幅に下げることができる。本実施形態によるバケット構築の詳細については、後述される。

【００３５】

（３）衝突に基づく応答：

衝突応答では１つの剛体を構成する球に加わる力を計算する。衝突力の計算では、接触する２つの球に関して、個別要素法で用いられる線形バネとダッシュポットを用いる。かかる衝突力の計算の内容については（数４）に示される。

【００３６】

【数４】

バネ定数κとし減衰定数ηとして、衝突している球にめり込み量に比例した反発力と相対速度に比例した減衰力を働かせる。２つの球iと球jがあり、これらの直径をdとすると、２つの球の距離$|\mathbf{r}ij|$がdより小さいときに衝突しており、(11)式によるバネによる力$\mathbf{f}s$と(12)式によるダンパによる減衰力$\mathbf{f}d$を働かせる。

$$\mathbf{f}s = -\kappa(d - |\mathbf{r}ij|)\frac{\mathbf{r}ij}{|\mathbf{r}ij|} \qquad (11)$$

$$\mathbf{f}d = -\eta(\mathbf{v}j - \mathbf{v}i) \qquad (12)$$

(11)式と(12)式で$\mathbf{r}ij = \mathbf{r}j - \mathbf{r}i$であり、$\mathbf{r}i, \mathbf{r}j$は球i, jの位置ベクトルである。
また剪断方向の摩擦力$\mathbf{f}t$は剪断方向の速度$\mathbf{v}t$に比例した力として(13)式で計算される。

$$\mathbf{f}t = \kappa t\mathbf{v}t \qquad (13)$$

球iに働く力$\mathbf{f}i$は(14)式で計算される。

$$\mathbf{f}i = \mathbf{f}s + \mathbf{f}d + \mathbf{f}t \qquad (14)$$

球iに働く力$\mathbf{f}i$を用いて、衝突によって剛体に働く力$\mathbf{F}c$とトルク$\mathbf{T}c$はそれぞれ(15)式と(16)式で計算される。

$$\mathbf{F}c = \sum_{i\in剛体}\mathbf{f}i \qquad (15)$$

$$\mathbf{T}c = \sum_{i\in剛体}\mathbf{r}'i \times \mathbf{f}i \qquad (16)$$

【００３７】
次に、ＧＰＵ１２のみを用いることに基づく近傍粒子探索に用いるためのデータ構造の構築手法の高速化について説明する。本実施形態によれば、多数の剛体の衝突に関する上記の各計算は、すべて、ＧＰＵ１２上で実装される。これによって上記計算の高速化を達成することができる。ＧＰＵ１２での近傍粒子探索プログラム１３Ａ－１に基づく計算アルゴリズムを説明する。
【００３８】
最初に、上記の計算アルゴリズムのために用意されるデータ構造を説明する。
【００３９】

前述した各計算をＧＰＵ１２で行うとき、前述した各物理量は、ビデオメモリ１３Ｂに含まれる複数枚のテクスチャ１３Ｂ－１（図１に示す）として格納され、保持される。

【００４０】

具体的に、剛体計算を行うために、各剛体の重心座標、クオータニオン、並進運動量、角運動量については、それぞれ、ビデオメモリ１３Ｂに格納された２枚のテクスチャ１３Ｂ－１を用いる。衝突計算を行うために、各球の中心座標、速度、バケット、各球に働く力、各球に働くトルクについては、それぞれ、ビデオメモリ１３Ｂに格納された１枚のテクスチャ１３Ｂ－１を用いる。

【００４１】

本実施形態の場合、粒子法シミュレーションに基づく多数の剛体の衝突計算では、３次元空間を想定し、さらに、前述したように当該３次元空間を分割する３次元空間格子すなわちバケットを用意する。しかし、現在の使用可能なＧＰＵ１２では３次元テクスチャへの書込みを行うことはできない。そこで本実施形態では、（Ｌ×Ｌ×Ｌ）個の格子数から成る３次元バケットを、所要枚数の２次元格子を（Ｄ×Ｄ）枚敷き詰めて構成される１枚の２次元テクスチャを利用して表現するようにした。この場合、３次元バケット上でのインデックスを（ｉ，ｊ，ｋ）とし、かつ２次元テクスチャ上のインデックスを（ｓ，ｔ）とするとき、これらの２つのインデックスの間は、次の（１７）式と（１８）式で関係づけられる。

$$s = i + L(k - D[k/D]) \qquad (17)$$

$$t = j + Lf[k/D] \qquad (18)$$

なおｉ，ｊ，ｋとｓ，ｔは、それぞれ、［０，Ｌ－１］と［０，Ｄ×Ｌ－１］の範囲に含まれる値を取る。

【００４２】

次に、図５に示したフローチャートを参照して、ＧＰＵ１２を用いた高速化の計算アルゴリズムを説明する。

【００４３】

表示装置１５の表示画面で描画を行うための１回の計算（シミュレーション）を行う単位時間を１タイムステップとするとき、当該１タイムステップでの計算は以下の８段階で計算される。

【００４４】

段階１　：球の物理量の計算　…（ステップＳ１１）
段階２　：バケット構築　…（ステップＳ１２）
段階３　：衝突検出　…（ステップＳ１３）
段階４　：衝突応答（球に働く力とトルクの計算）…（ステップＳ１４）
段階５　：剛体並進運動量の計算　…（ステップＳ１５）
段階６　：剛体角運動量の計算　…（ステップＳ１６）
段階７　：剛体重心座標の計算　…（ステップＳ１７）
段階８　：剛体クオータニオンの計算　…（ステップＳ１８）

【００４５】

上記の計算（シミュレーション）で最終的に得られた剛体の重心座標とクオータニオンを用いて表示のためのレンダリング処理が行われ、シミュレーションの結果画像の描画が行われる（ステップＳ１９）。次の判断ステップＳ２０ではシミュレーションを継続する否かを判断し、ＹＥＳの場合には上記のステップＳ１１に戻ってステップＳ１１～Ｓ１９を繰り返し、ＮＯの場合にはシミュレーションを終了する。上記のごとく、１回の表示のためのレンダリング処理（ステップＳ１９）に対して、上記の８段階の計算ステップＳ１１～Ｓ１８が１タイムステップの時間内で実行される。

【００４６】

上記において、ステップＳ１２で実行される「バケット構築」の計算で、近傍探索の処理計算が行われる。この計算はＧＰＵ１２のみで完結的に行われ、これにより計算の高速化が図られる。バケット構築の計算の詳細内容については以下で説明される。

【００４７】
以下に、上記のステップＳ１１～Ｓ１９の計算内容を説明する。
【００４８】
ステップＳ１１：球の物理量の計算
多数の剛体の衝突計算を粒子法で計算するには、各剛体を構成する球の中心座標と速度を計算する。球の中心座標と速度に関する計算の内容は（数５）に示す通りである。
【００４９】
【数５】

剛体jの重心座標、クオータニオン、速度、角速度をそれぞれ
$\mathbf{X}_j$, $\mathbf{Q}_j$, $\mathbf{V}_j$, $\mathbf{W}_j$とする。

回転していない状態での剛体の重心に対する球iの中心の
相対位置ベクトルを$\mathbf{r}_i$とすると、クオータニオン$\mathbf{Q}_j$で
回転している状態での相対位置、ベクトル$\mathbf{r}'_i$は(19)式で
計算される。

$$\mathbf{r}'_i = \mathbf{Q}_j \mathbf{r}_i \mathbf{Q}_j^{\mathsf{T}} \qquad (19)$$

剛体jを構成する球iの中心座標と速度をそれぞれ$\mathbf{x}_i$, $\mathbf{v}_i$とすると、
(20)式と(21)式で計算される。

$$\mathbf{x}_i = \mathbf{X}_j + \mathbf{r}'_i \qquad (20)$$

$$\mathbf{v}_i = \mathbf{V}_j + \mathbf{W}_j \times \mathbf{r}'_i \qquad (21)$$

【００５０】
ステップＳ１２：バケット構築
「バケットの構築」とは、多数の剛体の各々を表現するために用意された多数の球（粒子）のすべての番号を、前述のごとく、ビデオメモリ１３Ｂの内部のデータ構造として、空間を分割する要素として用意された３次元空間格子すなわちバケットに格納する処理を意味する。具体的には、多数の球のそれぞれに割り振られた番号を、多数のバケットのそれぞれの内部に格納する処理であり、多数の球のそれぞれに関するデータを分散させる（scattering）処理である。
【００５１】
バケットの構築に関する上記処理では、バーテックス（頂点）シェーダによる書き込み機能（処理のためのレンダリング）を用いる。当該書き込み機能は、入力とされる座標に頂点を書き込むことができるという機能である。換言すれば、バーテックスシェーダでは、ビデオメモリ１３Ｂのテクスチャにアクセスするためのバーテックステクスチャフェッチ（ＶＴＦ）という機能を有しており、バーテックスシェーダの当該機能を用いることにより、球の座標を参照することができる。その球の座標に対応するバケットの座標に頂点を配置することによって、ＧＰＵ１２において上記のデータ分散処理を行うことができる。
【００５２】
球の数をｎ（i_0，i_1，…，i_{n-1}）とするとき、上記の頂点は球の数と同じ数ｎ（p_0，p_1，…，p_{n-1}）だけ用意される。これらの頂点の各々に対しては１つの球が対応づけられる。頂点p_0，p_1，…，p_{n-1}の各々に対応する球の番号をi_0，i

$_1$，…，i_{n-1}とする。球の番号は例えば$i_0 < i_1 < \cdots < i_{n-1}$の順序（昇順）で並んでいるとする。なお球の並び方は降順であってもかまわない。

【００５３】

上記の対応関係を前提にして、バーテックスシェーダは、上記のバーテックステクスチャフェッチ（ＶＴＦ）を用いて、前回のタイムステップでのすべての球の座標を参照し、当該座標に基づきバケットテクスチャ上での座標を計算し、その座標に球の番号を出力することで、バケットテクスチャに球の番号を書き込むことが可能となる。

【００５４】

ところが、上記の処理の場合には、バケットの内部に１つの球が存在しているときのみに正しくバケット構築することができ、他方、１つのバケットの内部に複数個の球が存在しているときにはバケットを正しく構築することができない。１つのバケットの内部に複数個の球が存在する可能性のあるとき、正しいバケット構築を実現するためには、そのバケットに格納された球の数の番号を参照しながら，球の１個１個を順にバケットに格納しなければならない。ＧＰＵ１２に用いられる並列処理によれば、１回の処理のためのレンダリングでは書き込まれた球の番号を参照することができないので、当該並列処理後には各バケットには１個の番号しか書き込まれない。例えば１つのバケットに４個の球が存在するときには、４つの頂点が同一のピクセルに書き込まれるため、最終的にはそのピクセルには１個の球についての番号が書き込まれることになることになる。

【００５５】

そこで、本実施形態によるバケット構築では、１つのバケットb_jの内部には、複数の球を格納し得るように構成する。この方法を図６のフローチャートを参照して説明する。

【００５６】

ここで、バケットb_jに最大４個の球が入る場合を考える。これらの４個の球の番号をi_{b0}，i_{b1}，i_{b2}，i_{b3}として１ピクセルのＲＧＢＡチャネルのそれぞれの値を書き込む。これらの４つの球の番号は$i_{b0} < i_{b1} < i_{b2} < i_{b3}$の順（昇順）に並んでいるものとする。なお４つの球の番号の並び方は降順であってよい。そして、バケット構築では、１つのピクセルに対する書き込みを４回の球番号の書き込みに分けることにより、１個のバケットの内部に最大４個の球が存在する場合にも、ＧＰＵ１２を用いて行うことができる。換言すれば、バケットb_jに４個の球が存在している場合には、１タイムステップで４回の書き込み（処理のためのレンダリング）に分けることで、ＧＰＵ１２に基づきバケット構築を行う。

【００５７】

前述したバーテックスシェーダ（ＶＳ）２２では、頂点に対応する球の座標を用いて、その座標のバケット内での座標を計算し、球の番号をフラグメントシェーダ（ＦＳ）２３に渡す。フラグメントシェーダ（ＦＳ）２３では、球の番号を「色」と「深度」として書き出す。

【００５８】

図６を参照して、バケット構築Ｓ１２の処理内容を具体的に述べる。バケット構築Ｓ１２の工程は、順次に行われる４つのパス（pass）のステップＳ１０１～Ｓ１０４から成っている。

【００５９】

１パス目における１つのピクセルへの最初の書き込み（ステップＳ１０１）は、Ｒチャンネルに球番号i_{b0}を書き込むステップである。球番号i_{b0}は４個の球のうち最も小さい番号を持つものである。すなわち、最小の深度を有するピクセルとして書き込まれ、最初の処理のためのレンダリングが行われる。この球番号i_{b0}の書き込みの場合には、まずＲチャンネルを選択するためのカラーマスクを設定し（ステップＳ３１）、深度テストを設定し（ステップＳ３２）、次いで深度バッファを最大値で初期化し（ステップＳ３３）、設定された上記深度テストを用いて小さい値を持つものを合格させるようにし、当該書き込み（頂点のレンダリングＳ３４）を可能にしている。

【００６０】

２パス目における上記同一ピクセルへの書き込み（ステップＳ１０２）は、Ｒチャンネルに書き込まれた値を上書きしないように、Ｇチャンネルに球番号i_{b1}を書き込むステップである。この書き込みでは、Ｇチャンネルを選択するためのカラーマスクを用いることにより（ステップＳ４１）、ＲＢＡの各チャンネルへの書き込みを禁止する。深度バッファは、１パス目で用いたものをそのまま用い、かつ深度テストは大きな値を持つものを合格させるように設定する（ステップＳ４２）。しかしこれでは、上記ピクセルのＧチャンネルには最大の球番号i_{b3}が書き込まれることになるので、これを禁止するためステンシルテストを用いる（ステップＳ４３）。ステンシルテストでは、ステンシルバッファを０で初期化し（ステップＳ４４）、ステンシルテストでは値を増加させるように設定する。さらにステンシルテストでは値が１以上であるときには書き込みを失敗するように設定する（ステップＳ４３）。これによって、ピクセルに１回書き込み処理を行った後において書き込み失敗になった球番号を書き込まないようにすることができる。このようにして、同一ピクセルへの２パス目の書き込みにおいて、Ｇチャンネルに球番号i_{b1}を書き込む（ステップＳ４５）。

【００６１】

その後、同一ピクセルにおける３パス目のＢチャンネルへの球番号i_{b2}の書き込み（ステップＳ１０３）、４パス目の球番号i_{b3}の書き込み（ステップ」Ｓ１０４）の場合にも、上記と同様な処理が行われ、球番号i_{b2}および球番号i_{b3}をそれぞれＢチャンネルおよびＡチャンネルに書き込む。３パス目のステップＳ１０３は、Ｂチャンネルを選択するためのカラーマスクの設定、ステンシルバッファのクリア、頂点のレンダリング（書き込み）のステップＳ５１～Ｓ５３を有し、４パス目のステップＳ１０４は、Ａチャンネルを選択するためのカラーマスクの設定、ステンシルバッファのクリア、頂点のレンダリング（書き込み）のステップＳ６１～Ｓ６３を有している。

【００６２】

以上のように、ＧＰＵ１２を用いた近傍粒子探索に用いるデータ構造の構築、すなわちバケット構築では、バケットに複数個（例えば４個）の球が存在する場合において、これらの球の番号を１つのバケットに格納しかつピクセルに対応づけるとき、バケットへの格納処理（ピクセルのＲＧＢＡチャンネルへの球番号の書き込み処理）を複数回（例えば４回）の処理のためのレンダリングに分けて行う。１回の処理のためのレンダリングでは、前述のルールに従って、１個の球の番号をピクセルの１つのチャンネルに書き込む。各レンダリングにおける球番号の書き込み順序と書き込みチャンネルの選択に関する前述のルールは、ＧＰＵ１２の有する機能である深度テスト、カラーマスク、ステンシルテストを利用して決定される。

【００６３】

ステップＳ１３：衝突計算

この衝突計算では、或る球ｉと衝突している可能性のある近傍に存在する球の探索を行う。この探索では、球の座標テクスチャとして生成したバケットテクスチャを入力とし、フラグメントシェーダを用いて処理を行う。この処理によって、バケットを参照することで、或る球ｉが格納されているバケットを囲む3^3個のバケット内に格納されている球の番号を得て，その番号を用いて球の座標が格納されているテクスチャを参照することでその球座標（球の中心座標）を得る。各球の球座標を用いてこれらの球と球ｉとの距離計算を行い、衝突計算を行う。

【００６４】

ステップＳ１４：衝突応答（球に働く力とトルクの計算）

衝突していると判断された２つの球に関しては、２つの球の座標（位置ベクトル）と、それらの速度（速度ベクトル）と、剛体の重心に対する相対位置ベクトルとを用いて、前述の（１２）式から、球に働く力と、その力によって剛体に生じるトルクを計算することができる。

【００６５】

ステップＳ１５，Ｓ１６：剛体の並進運動量と角運動量の計算

ステップＳ１４で、球に働く力とその力によって剛体に生じるトルクとを計算したので、前述した（１５）式と（１６）式を用いてそれらを足し合わせ、多数の剛体の各々に働く力を計算する。次いで、前述の（２）式と（４）式を用いて剛体の並進運動量と角運動量を更新する。

【００６６】

ステップＳ１７，Ｓ１８：剛体の重心座標とクオータニオンの計算

剛体の重心座標は前述の（３）式を用いて計算される。本実施形態では、剛体の重心座標と重心速度テクスチャを入力として、他の１枚の剛体の重心座標テクスチャに出力を行う。また剛体のクオータニオンの更新は前述の（１０）式を用いて計算される。本実施形態では、さらに、剛体のクオータニオンと角速度テクスチャを入力として他の１枚の剛体のクオータニオンテクスチャに更新された値を書き出す。

【００６７】

ステップＳ１９：レンダリング処理（表示のためのレンダリング）

上記のごとくして計算された剛体の重心座標とクオータニオンを用いて、下記の（数６）に示される計算内容に基づいて表示装置１５の画面に描画を行う表示のためのレンダリングが行われる。

【００６８】

【数６】

頂点iの回転していない時の剛体の中心からの相対位置ベクトル
$\mathbf{r}_i$を、最初にクオータニオン$\mathbf{Q}$を用いて現在の剛体の回転量だけ
回転させ、次いで剛体の重心座標$\mathbf{X}$だけ移動させると、
頂点iの現在の相対位置ベクトル$\mathbf{r}'$は(22)式で計算される。

$$\mathbf{r}'_i = \mathbf{X} + \mathbf{Q}\mathbf{r}_i\mathbf{Q}^T \qquad (22)$$

次にモデルビュープロジェクション行列Ｍを用いて現在
レンダリングされている座標系での座標$\mathbf{r}''$に(23)式に基づいて
変換する。

$$\mathbf{r}''_i = \mathrm{M}\mathbf{r}'_i \qquad (23)$$

【００６９】

ＧＰＵ１２での計算においては、基本的に以上の８段階のアルゴリズムを繰り返すことにより、表示装１５の表示画面に、多数の剛体の衝突シミュレーションの画像をリアルタイムで表示することが可能になる。

【００７０】

なお、図５に示した剛体シミュレーションの高速化の計算アルゴリズムの変形例を図７に示す。図７の処理の流れにおいて、図５で説明したステップと実質的に同一のステップには同一の符号を付している。図７に示した計算アルゴリズムによれば、剛体並進運動量の計算Ｓ１５および剛体重心座標の計算Ｓ１６と、剛体角運動量の計算Ｓ１７および剛体クオータニオンの計算Ｓ１８とが並列的な処理によって同時に実行されるように構成することができる。これはＧＰＵ１２における並列処理を利用したものである。その他の処理の流れは図５で説明した内容と同じである。これによれば、剛体シミュレーションの計算

アルゴリズムにおける１タイムステップの時間をさらに短縮でき、より一層高速化を達成することができる。

【００７１】

図３に示した１０９２２個のトーラスの落下・衝突の剛体シミュレーション画像では１タイムステップに要する時間は１６．６ミリ秒であり、レンダリング時のフレームレートは２３ＦＰＳであった。また図４に示した１６３８４個のチェスの駒の落下・衝突の剛体シミュレーション画像では１タイムステップに要する計算時間は１２．８ミリ秒であり、レンダリング時のフレームレートは２１．２ＦＰＳであった

【００７２】

上記の実施形態の説明では剛体の向きについて「クオータニオン」という物理量を用いたが、計算上ではこれの代わりに「回転行列」を用いることもできる。「クオータニオン」も「回転行列」も「物体の姿勢」を示す物理量である。

【００７３】

前述の実施形態での粒子法シミュレーションの近傍粒子探索に用いるデータ構造の構築手法は多数の剛体の衝突に関する剛体シミュレーションに適用されたが、本発明による当該データ構造の構築手法は、これに限定されず、例えば図８と図９に示すように流体や粉体の流動的な動き等の自然物表現に対しても適用することができる。

【００７４】

図８は、水等の流体に複数の球体形状の流体（液滴）が落下したときの流体の変化状態を示すシミュレーションの結果画像を示す。図９は、ロート状容器に収容された粉体が下方に流動し、容器の下端開口部から外側に落下して、下側に配置される容器に収容される変化状態を示すシミュレーションの結果画像を示している。

【００７５】

前述した近傍粒子探索に用いるデータ構造の構築方法は、図１に示した近傍粒子探索プログラム１３Ａ－１がＧＰＵ１２で実行されることにより実現される。近傍粒子探索に用いるデータ構造の構築方法に係る近傍粒子探索プログラム１３Ａ－１は、それ自体ＣＤＲＯＭ等の記憶媒体に格納され、独立したプログラム製品として構成され得る。

【００７６】

前述した実施形態の説明では、ＧＰＵでのプログラムの実装でシェーダ（Shader）を使う例を説明した。これはＧＰＵのグラフィックス機能に特化したプログラムモデルを使用した例である。しかしながら、本発明はこれに限定されない。グラフィックス演算プロセッサとして、上記のＧＰＵを含む一般的なストリーミングプロセッサを用いることができ、さらにプログラミングモデルとして一般的なプログラミングモデルを用いて実装することができる。

【００７７】

上記のＧＰＵ１２を含むストリーミングプロセッサ（Streaming Processor）では、一般的に、上記のシェーダ（Shader）の他に、クーダ（CUDA）、CTM、Brook等のプログラミングモデルを用いて実装できる。前述した実施形態によるＧＰＵにおけるシェーダ（Shader）の実装では、計算空間におけるスライス、ボクセル、および粒子に関するデータをテクスチャ１３Ｂ－１としてビデオメモリ１３Ｂに格納した。プログラム実装としてクーダ（CUDA）等を用いる場合にはテクスチャの代わりに一般的な「配列（Array）」が用いられる。

【００７８】

前述した通り、データ構造の構築方法でグラフィックス演算プロセッサとしてＧＰＵを用いる場合には、複数回の処理のためのレンダリングの各々で書き込む粒子の粒子番号と粒子座標の選択、および書き込まれるピクセルにおけるチャンネルの選択は、ＧＰＵに用意された深度テスト、カラーマスク、ステンシルテストの各機能を用いて行われた。

【００７９】

他方、グラフィックス演算プロセッサとして、ストリーミングプロセッサを用いる場合には、一般的に、上記のピクセル（written pixel）の代わりにメモリ位置（Memory Loca

tion）を用い、上記のチャンネルの代わりに、ボクセル（Voxel）のためのメモリにおけるメモリ位置を用い、さらに、ＧＰＵに用意された上記の深度テスト、カラーマスク、ステンシルテストの各機能の代わりに、ストリーミングプロセッサに用意されたカウンタの機能を用いる。このカウンタは、上記の深度テスト、カラーマスク、ステンシルテストの代わりに機能する。

【００８０】

従って、グラフィックス演算プロセッサとして、ストリーミングプロセッサを用いる場合には、複数回の処理サイクルの各ステップで書き込む粒子の粒子番号とメモリ位置（Memory Location）の選択、およびボクセル（Voxel）のためのメモリにおけるメモリ位置の選択は、ストリーミングプロセッサに用意されたカウンタの機能を用いて行われる。

【００８１】

前述した通り、データ構造の構築方法、すなわち図６に示したバケット構築Ｓ１２の工程は、グラフィックス演算プロセッサとしてＧＰＵ１２を用いてプログラム実装を行う場合には、

４個の粒子の粒子番号をi_{b0}，i_{b1}，i_{b2}，i_{b3}（$i_{b0}<i_{b1}<i_{b2}<i_{b3}$）であるとするとき、

１パス目でのピクセルへの書き込みでは、深度バッファを最大値で初期化し、深度テストを用いて小さい値を持つものを合格させることにより、Ｒチャンネルに粒子番号i_{b0}を書き込んで１回目の処理のためのレンダリングを行うステップ（Ｓ１０１）と、

２パス目でのピクセル等への書き込みでは、カラーマスクを用いてＲチャンネルへの書き込みを禁止しかつステンシルテスト等を用いることにより、Ｇチャンネルに粒子番号i_{b1}を書き込んで２回目の処理のためのレンダリングを行うステップ（Ｓ１０２）と、

３パス目でのピクセルへの書き込みでは、カラーマスクを用いてＲとＧの各チャンネルへの書き込みを禁止しかつステンシルテストを用いることにより、Ｂチャンネルに粒子番号i_{b2}を書き込んで３回目の処理のためのレンダリングを行うステップ（Ｓ１０３）と、

４パス目でのピクセルへの書き込みでは、カラーマスクを用いてＲとＧとＢの各チャンネルへの書き込みを禁止しかつステンシルテストを用いることにより、Ａチャンネルに粒子番号i_{b3}を書き込んで４回目の処理のためのレンダリングを行うステップ（Ｓ１０４）と、

を有していた。

【００８２】

上記のバケット構築Ｓ１２の工程に関する処理の意味は、一般化的に述べると、最大４個の値をストリーミングプロセッサで書き込み時には、４回のステップ（Ｓ１０１～Ｓ１０４）に分けて書き込みを行うという意味である。４回のステップの各々で、それまで書き込まれていない値が１個書き込まれる。

上記のＧＰＵ１２を用いたプログラム実装の場合には、ＧＰＵ１２のグラフィックス特有の機能である「深度テスト」、「カラーマスク」、「ステンシルテスト」が用いられる。

「深度テスト」という機能は、「今まで書き込んだ値を除外する」という処理を行う。「カラーマスク」という機能は、「書き込み先を選択する」という処理を行う。「ステンシルテスト」という機能は、「１ステップで１つの値のみの書き込みを許す」という処理を行う。

なお、「１ステップでデータの書き込み先のメモリに粒子番号を書き込む」という処理以外にも、「深度バッファ」と「ステンシルバッファ」も更新される。

【００８３】

上記のＧＰＵ１２を含むストリーミングプロセッサを用いる場合において、一般的に、プログラム実装で、クーダ（CUDA）というプログラムモデルを用いることができる。クーダ（CUDA）を用いる場合には、ＧＰＵ１２のグラフィックス機能である「深度テスト」、「カラーマスク」、「ステンシルテスト」を用いることができない。そこで、クーダ（CU

DA）を用いる場合には、これらのグラフィック機能と同等な機能を作るため、書き込み先のメモリと、カウンタを用意する。

【００８４】

上記のカウンタによれば、カウンタによって「今まで書き込んだ値を除外する」という処理（「深度テスト」の機能）を行い、さらに「書き込み先を選択する」という処理（「カラーマスク」の機能）と「１ステップで１つの値のみの書き込みを許す」という処理（「ステンシルテスト」の機能）を行うように構成する。

さらに「１ステップでデータの書き込み先のメモリに粒子番号を書き込む」という処理以外にも、上記の「カウンタ」も更新される。ここで、「カウンタの更新」は、カウンタを１だけインクレメントすることを意味する。

【００８５】

上記のごとく、ＧＰＵ等のストリーミングプロセッサを用いかつプログラミングモデルとしてクーダ（CUDA）を実装するとき、テクスチャの代わりにビデオメモリに作成された配列を用いる。なお、テクスチャは、「ビデオメモリに作成された配列」に含まれる概念である。「ビデオメモリに作成された配列」は、一般的に「専用メモリに作成された配列」と解釈することができる。

【００８６】

ＧＰＵ等のストリーミングプロセッサにおいて、プログラミングモデルとしてシェーダの代わりにクーダ（CUDA）を用いて実装する場合、ピクセル（written pixel）の代わりにメモリ位置（Memory Location）を用い、チャンネルの代わりにボクセル（Voxel）のためのメモリにおけるメモリ位置を用い、さらに深度テスト、カラーマスク、ステンシルテストの各機能の代わりに、それぞれに対応するカウンタの機能を用いる。

【産業上の利用可能性】

【００８７】

本発明は、剛体の衝突計算に基づく剛体シミュレーションや流体等のシミュレーションに利用され、さらにゲーム等のリアルタイムアプリケーションにおける自然物等の表現手段として利用することができる。

【図面の簡単な説明】

【００８８】

【図１】本発明に係る近傍粒子探索に用いるデータ構造の構築方法が実施されるコンピュータシステムの基本的な構成を示すシステム構成図である。

【図２】コンピュータシステムに実装されたＧＰＵの内部の基本的構成を示すブロック図である。

【図３】１０９２２個のトーラスを落下・衝突させた物理現象に関する剛体シミュレーションに基づく結果画像（ＣＧ）の図である。

【図４】１６３８４個のチェスの駒を落下・衝突させた物理現象に関する剛体シミュレーションに基づく結果画像（ＣＧ）の図である。

【図５】ＧＰＵを用いた剛体シミュレーションの高速化の計算アルゴリズムを示すフローチャートである。

【図６】バケット構築の処理手順の詳細内容を示すフローチャートである。

【図７】ＧＰＵを用いた剛体シミュレーションの高速化の計算アルゴリズムの変形例を示すフローチャートである。

【図８】流体に複数の球体形状の流体が落下したときの物理現象に関する流体シミュレーションに基づく結果画像（ＣＧ）の図である。

【図９】ロート状容器に収容された粉体が下方に流動し、容器の下端開口部から外側に落下して、下側に配置される容器に収容されるときの粉体シミュレーションに基づく結果画像（ＣＧ）の図である。

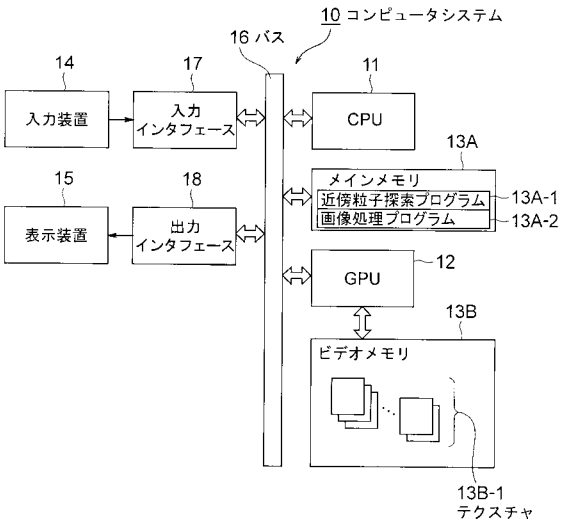
【符号の説明】

【００８９】

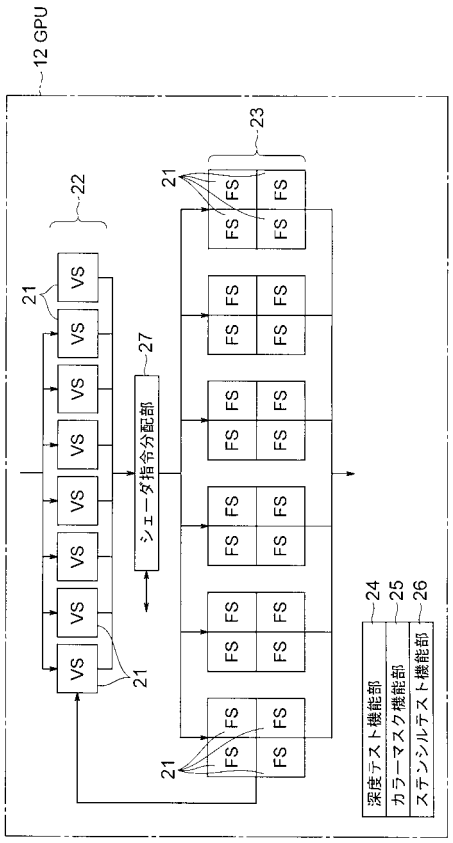
１０　　コンピュータシステム

１１　　ＣＰＵ
１２　　ＧＰＵ
１３Ａ　　メインメモリ
１３Ａ－１　　近傍粒子探索プログラム
１３Ａ－２　　画像処理プログラム
１３Ｂ　　ビデオメモリ
１３Ｂ－１　　テクスチャ

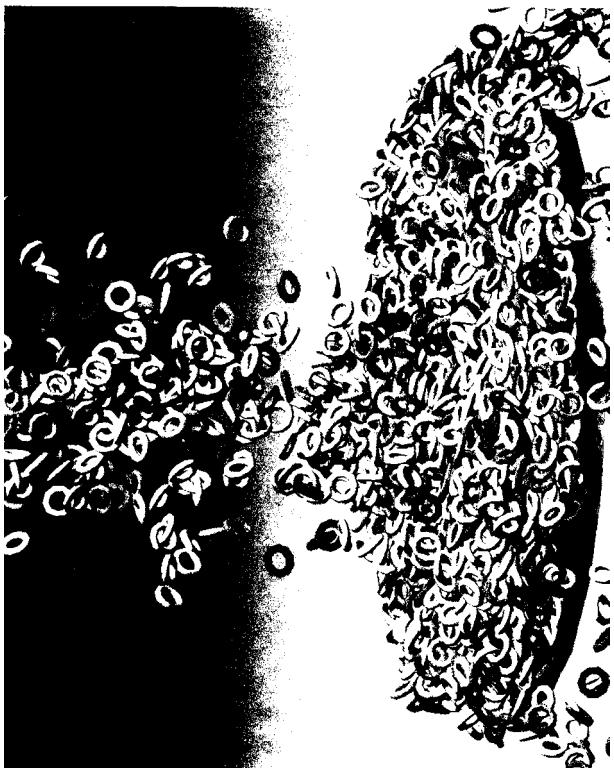
【図１】



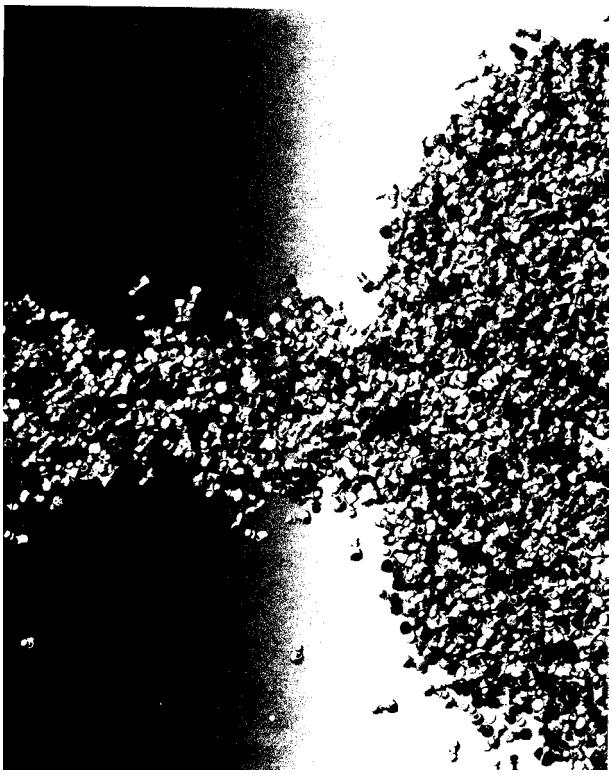
【図２】



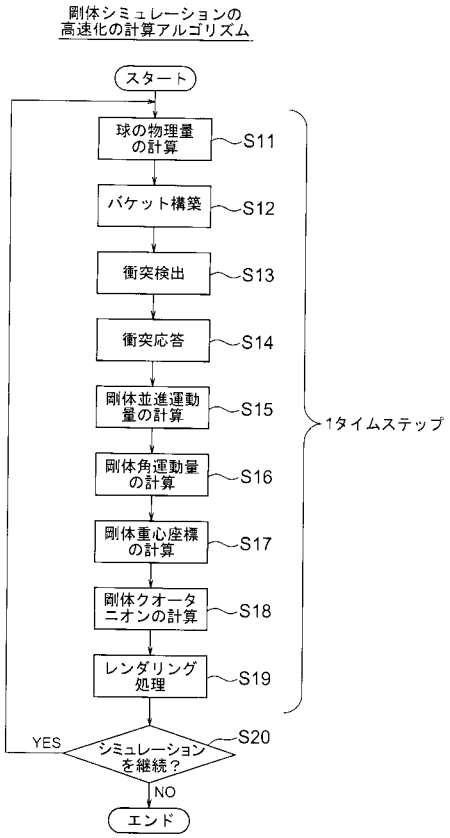
【図３】



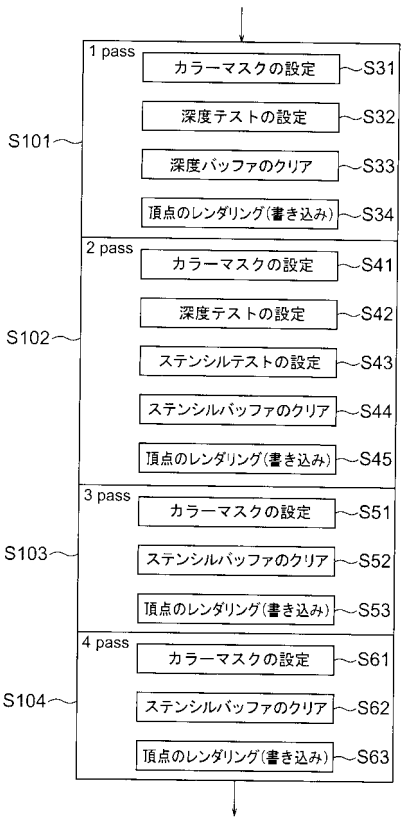
【図４】



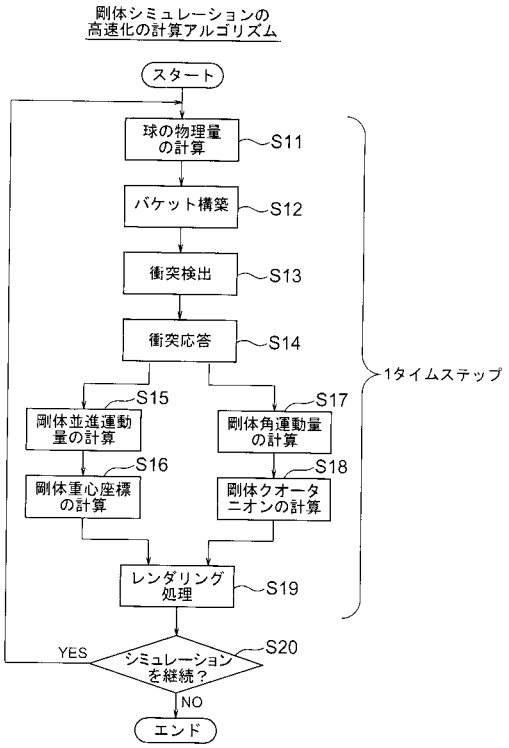
【図５】


剛体シミュレーションの
高速化の計算アルゴリズム
スタート
球の物理量の計算
S11
バケット構築
S12
衝突検出
S13
衝突応答
S14
剛体並進運動量の計算
S15
剛体角運動量の計算
S16
剛体重心座標の計算
S17
剛体クオータニオンの計算
S18
レンダリング処理
S19
1タイムステップ
YES
シミュレーションを継続？
S20
NO
エンド

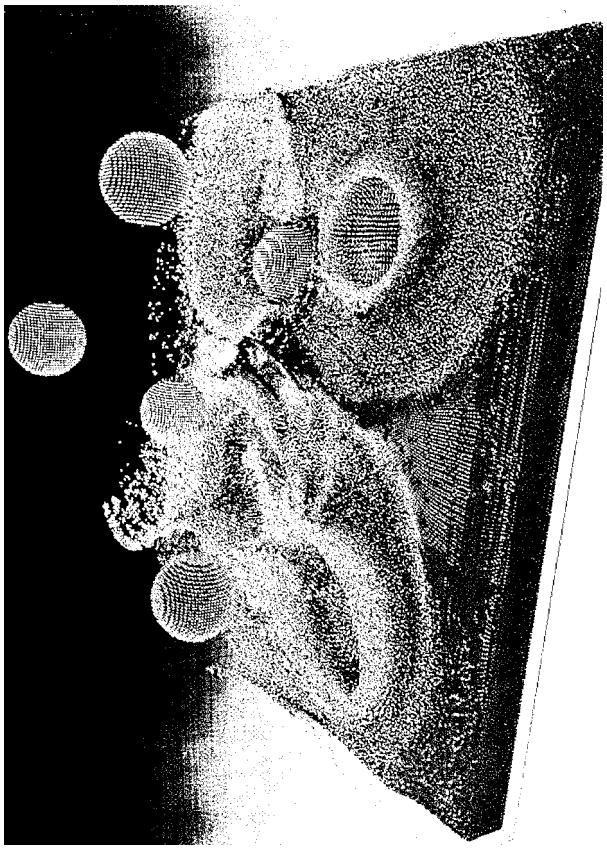
【図６】


S101
1 pass
カラーマスクの設定
S31
深度テストの設定
S32
深度バッファのクリア
S33
頂点のレンダリング(書き込み)
S34
S102
2 pass
カラーマスクの設定
S41
深度テストの設定
S42
ステンシルテストの設定
S43
ステンシルバッファのクリア
S44
頂点のレンダリング(書き込み)
S45
S103
3 pass
カラーマスクの設定
S51
ステンシルバッファのクリア
S52
頂点のレンダリング(書き込み)
S53
S104
4 pass
カラーマスクの設定
S61
ステンシルバッファのクリア
S62
頂点のレンダリング(書き込み)
S63

【図７】


剛体シミュレーションの
高速化の計算アルゴリズム
スタート
球の物理量の計算
S11
バケット構築
S12
衝突検出
S13
衝突応答
S14
S15
剛体並進運動量の計算
S16
剛体重心座標の計算
S17
剛体角運動量の計算
S18
剛体クオータニオンの計算
レンダリング処理
S19
1タイムステップ
YES
シミュレーションを継続？
S20
NO
エンド

【図８】



【図９】