

(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2007-244887

(P2007-244887A)

(43) 公開日 平成19年9月27日(2007.9.27)

(51) Int. Cl.	F I	テーマコード (参考)
A 6 1 B 6/03 (2006.01)	A 6 1 B 6/03 3 6 0 G	4 C 0 9 3
G 0 6 F 9/50 (2006.01)	G 0 6 F 9/46 4 6 5 D	4 C 0 9 6
A 6 1 B 5/055 (2006.01)	A 6 1 B 5/05 3 8 0	

審査請求 有 請求項の数 10 O L (全 27 頁)

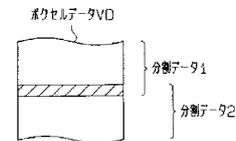
(21) 出願番号	特願2007-113406 (P2007-113406)	(71) 出願人	500109320 ザイオソフト株式会社
(22) 出願日	平成19年4月23日 (2007. 4. 23)		東京都港区三田 1 丁目 2 番 1 8 号
(62) 分割の表示	特願2002-100111 (P2002-100111) の分割	(74) 代理人	100068755 弁理士 恩田 博宣
原出願日	平成14年4月2日 (2002. 4. 2)	(74) 代理人	100105957 弁理士 恩田 誠
(31) 優先権主張番号	特願2001-369119 (P2001-369119)	(72) 発明者	松本 和彦 東京都港区三田 1 丁目 2 番 1 8 号 ザイオ ソフト 株式会社内
(32) 優先日	平成13年12月3日 (2001. 12. 3)	F ターム (参考)	4C093 AA26 CA29 FD07 FD11 FF33 FF42 FH06 4C096 AB27 AD14 AD16 DC33 DC36 DE06
(33) 優先権主張国	日本国 (JP)		

(54) 【発明の名称】 ボリュームレンダリング処理方法、ボリュームレンダリング処理システム、計算機及びプログラム

(57) 【要約】

【課題】 ボクセルデータ分割による分散処理で得られた分散レンダリング処理結果を一つに連結する際に、分割面近傍のレンダリング処理を補う計算処理が伴わずに済み、高コストパフォーマンスを満たす単一計算機では扱うことが難しい大容量ボクセルデータでも高速にボリュームレンダリング処理することができるボリュームレンダリング処理方法、ボリュームレンダリング処理システム、計算機及びプログラムを提供する。

【解決手段】 ストレージサーバーは、分散処理でボリュームレンダリング処理を行う複数のノードにボクセルデータを分割して割り当てる。このとき、ボクセルデータを、分割面で一部データが重複するように分割する。各ノードは、分割されたボクセルデータに対する分散レンダリング処理を行い、分割面近傍のレンダリング処理の補間計算については重複した部分のデータを用いて行う。



【選択図】 図 1 2

【特許請求の範囲】

【請求項 1】

複数のノードによってコンピュータネットワークが構築され、三次元以上のボクセルデータに施すべきボリュームレンダリング処理を少なくとも二のノードが分散処理で実行するボリュームレンダリング処理システムにおいて、

少なくとも一のノードに設けられ、各ノードの計算余力及びメモリ空き状況に係る計算資源使用状況情報を基に分散処理可能な余力のある少なくとも二のノードを割り当てるノード割当手段と、

少なくとも一のノードに設けられ、与えられたボクセルデータを、各割り当てノードの計算資源使用状況情報に応じた分割割合で、かつ分割面に関わるボクセルデータを重複させるように複数の分割ボクセルデータに分割するデータ分割割当部と、

10

前記分割ボクセルデータを動的に前記各割り当てノードに割り当てる負荷割当手段と、

前記割り当てノードとなる少なくとも二のノードに設けられ、割り当てられた分割ボクセルデータに対して分散レンダリング処理を行うとともに、少なくとも分割面近傍における分散レンダリング処理の補間計算を前記データの重複部分を用いて行う演算手段と、

少なくとも一のノードに設けられ、前記各割り当てノードにおける分散レンダリング処理結果を取得して一つに整合させて連結する連結手段と、

少なくとも一のノードに設けられ、前記連結されたレンダリング処理結果に基づくボリュームレンダリング画像を表示手段に表示する表示制御手段と

を備えたことを特徴とするボリュームレンダリング処理システム。

20

【請求項 2】

前記分割面に関わるボクセルデータを重複させるとは、前記分割面近傍における補間計算が可能な量を重複させることをいうことを特徴とする請求項 1 に記載のボリュームレンダリング処理システム。

【請求項 3】

前記補間計算は、グラディエント計算又はボクセル補間値計算であることを特徴とする請求項 2 に記載のボリュームレンダリング処理システム。

【請求項 4】

前記分割ボクセルデータに対する計算負荷を、前記分割ボクセルデータの割り当て先の各ノードの計算資源使用状況情報に応じた分割割合で分割する計算分割割当部を更に備え、

30

前記負荷割当手段は、前記計算分割割当部が分割した計算負荷を前記割り当て先の各ノードに割り当てることを特徴とする請求項 1 ~ 3 のいずれか一項に記載のボリュームレンダリング処理システム。

【請求項 5】

請求項 1 ~ 3 のいずれか一項に記載のボリュームレンダリング処理システムにおけるノードを構成する計算機であって、

前記ノード割当手段、前記データ分割割当部及び前記負荷割当手段を備えたことを特徴とする計算機。

【請求項 6】

複数のノードによって構築されるコンピュータネットワークを用いて、少なくとも二のノードが、三次元以上のボクセルデータに施すべきボリュームレンダリング処理を分散処理で実行するボリュームレンダリング処理システムにおける前記ノードの機能をコンピュータに実現させるプログラムであって、

40

コンピュータを、

各ノードの計算余力とメモリ空き状況に係る計算資源使用状況情報に基づき分散処理を割り当て可能な余力のある少なくとも二のノードを割り当てるノード割当手段と、

与えられたボクセルデータを、前記各割り当てノードの計算資源使用状況情報に応じた分割割合で、かつ分割面に関わるボクセルデータを重複させるように複数の分割ボクセルデータに分割するデータ分割割当部と、

前記分割ボクセルデータを動的に前記各割り当てノードに割り当てる負荷割当手段と、

50

割り当てられた分割ボクセルデータに対して分散レンダリング処理を行うとともに、少なくとも分割面近傍における分散レンダリング処理の補間計算を前記データの重複部分を用いて行う演算手段と、

前記各割り当てノードにおける分散レンダリング処理結果を取得して一つに整合させて連結する連結手段と、

前記連結されたレンダリング処理結果に基づくボリュームレンダリング画像を表示手段に表示する表示制御手段、として機能させることを特徴とするプログラム。

【請求項 7】

前記分割面に関わるボクセルデータを重複させるとは、前記分割面近傍における補間計算が可能な量を重複させることをいうことを特徴とする請求項 6 に記載のプログラム。

10

【請求項 8】

前記補間計算は、グラディエント計算又はボクセル補間値計算であることを特徴とする請求項 7 に記載のプログラム。

【請求項 9】

前記分割ボクセルデータに対する計算負荷を、前記分割ボクセルデータの割り当て先の各ノードの計算資源使用状況情報に応じた分割割合で分割する計算分割割当部としてもコンピュータを機能させ、

前記負荷割当手段は、前記計算分割割当部が分割した計算負荷を前記各割り当て先の各ノードに割り当てることを特徴とする請求項 6 ~ 8 のいずれか一項に記載のプログラム。

【請求項 10】

20

複数のノードによって構築されるコンピュータネットワークを用いて、三次元以上のボクセルデータに対して少なくとも二のノードが分散処理でボリュームレンダリング処理を施すボリュームレンダリング処理方法であって、

前記複数のノードのうち少なくとも一のノード割当手段が、各ノードの計算余力及びメモリ空き状況に係る計算資源使用状況情報に基づいて分散処理可能な余力のある少なくとも二のノードを割り当てる段階と、

少なくとも一のノードのデータ分割割当部が、与えられたボクセルデータを、各割り当てノードの計算資源使用状況情報に応じた分割割合で、かつ分割面に関わるボクセルデータを重複させるように複数の分割ボクセルデータに分割する段階と、

少なくとも一のノードの負荷割当手段が、前記分割ボクセルデータを動的に前記各割り

30

当てノードに割り当てる段階と、
前記割り当てノードの演算手段が、割り当てられた分割ボクセルデータに対して分散レンダリング処理を行うとともに、少なくとも分割面近傍における分散レンダリング処理の補間計算を前記データの重複部分を用いて行う段階と、

少なくとも一のノードの連結手段が、前記各割り当てノードにおける分散レンダリング処理結果を取得して一つに整合させて連結する段階と、

少なくとも一の表示制御部が、前記連結されたレンダリング処理結果に基づくボリュームレンダリング画像を表示手段に表示する段階と
を備えたボリュームレンダリング処理方法。

【発明の詳細な説明】

40

【技術分野】

【0001】

本発明は、複数のノード（計算機）を備えたコンピュータネットワーク上で行われる、動的負荷割り当てを用いた協調分散処理によるボリュームレンダリング処理方法、ボリュームレンダリング処理システム、計算機及びプログラムに関するものである。

【背景技術】

【0002】

近年、ネットワーク環境および計算機が低価格化・高性能化により多くの病院に普及するようになってきている。病院内の複数箇所での利用ニーズに従い、院内ネットワークに複数台の画像処理システムがある場合も珍しくなくなっている。CT, MRI など医

50

療画像診断機器で撮影された医療画像データ（ボクセルデータ）を画像処理システムの画面で見るときの画像処理にはボリュームレンダリング処理が行われる。

【0003】

CT, MRI など医療画像診断機器の発達による測定分解能の向上とともに、医療画像データ（ボクセルデータ）のデータ容量が増大している。リアルタイムでデータ本来の分解能を反映させながらデータを間引かずにボリュームレンダリング処理を行うためには、大容量のメモリと高速なCPUなど膨大な計算資源を持つ計算機が必要である。現状では上記条件を満たすような計算機は現実的なコストでは得られない。そのためボリュームレンダリング処理を行うことができるボクセルデータの容量に限界がある。すなわち、現状ではコストパフォーマンスの高い計算機1台当たりが装備可能なメモリ容量と計算能力に

10

【0004】

特に医療画像データのデータ容量の増加が著しい。この要因には時間分解能と空間分解能の向上が挙げられる。通常使用される3D医療画像データは、幅（X軸）、高さ（Y軸）が同じ2D画像（体の断面画像、1スライス）を体軸方向（Z軸）に何枚も積層することで構成される。さらに時間軸（T軸）が加わった4D医療画像データは、3D医療画像データの時系列で構成されるが、すでにデータ容量が大きい3Dデータを複数持つことになるため、膨大なデータ容量となる。診断機器の技術革新による時間分解能の向上（T軸方向）によって、3D医療画像の時系列データを臨床現場で取得できるようになってきた

20

【0005】

また、空間分解能の向上（X軸、Y軸方向）も進み、現在は1スライスの解像度は512 × 512 が主流であるがいずれ1024 × 1024 になると言われている。さらに、空間分解能の向上（Z軸方向）も挙げられ、1スライスの厚さが薄くなることで同じ撮影部位が従来よりも多くのスライス枚数から構成されるため、データ容量が増加している。

【0006】

従来、図16に示すように、1台の計算機50で完結する画像処理システムがあった。この画像処理システムでは、一台の計算機（複数CPUの場合有り）50でボリュームレンダリング処理の全工程が行われる。

30

【0007】

また図17に示すように、ボクセルデータのストレージを行うストレージサーバー52とボリュームレンダリング処理・表示を行う画像処理システム53とに分け、ストレージサーバー52と複数台の計算機50をネットワークで繋いで構築されるシステム51があった。画像処理システム53は、必要なボクセルデータをストレージサーバー52から動的に取得して、画像処理システム内のメモリに読み込んだ後、ボリュームレンダリング処理を行う。長所としては、ボリュームレンダリングで編集したい、必要なボクセルデータのみ画像処理システム53で保持すればよい。つまり、ローカル・ストレージの浪費や、ローカル・ストレージ容量による制限がなくなる。

【0008】

さらに図18に示すように、ボクセルデータのストレージとボリュームレンダリング処理を行うストレージ・計算サーバー56と、ボリュームレンダリング処理結果を表示する画像表示システム57とに分け、このサーバー56と複数台の計算機50をネットワークで繋いで構築されるシステム55があった。長所としては、(1) 処理結果を表示する計算機50は計算資源が少なくてもよく、画像表示システム57を低コストにすることができる、(2) 動的に計算資源を複数のユーザーに配分することができるなどが挙げられる。

40

【0009】

また、例えば非特許文献1には、分散処理において、レンダリング計算を画素ごとに分割するボリュームレンダリング処理方法が開示されている。また、非特許文献2には、分散処理において、ボクセルデータ（ボリューム）を分割する技術が開示されている。

50

【非特許文献1】吉岡政洋、外3名、ソフトウェアによる高速ボリュームレンダリング手法の開発と仮想化内視鏡システムへの適用、MEDICAL IMAGING TECHNOLOGY、日本、2001年11月、Vol. 19, No. 6, pp. 477 - 486

【非特許文献2】佐野健太郎、外3名、データ並列ボリュームレンダリングのためのボリューム適応分割手法、情報処理学会研究報告、日本、社団法人情報処理学会、1998年10月9日、Vol. 98, No. 93 (HPC-73), pp.7 -12

【発明の開示】

【発明が解決しようとする課題】

【0010】

しかしながら、非特許文献2に記載のように、ボクセルデータ（ボリューム）を分割してノード（計算機）に割り当てる構成を採用する場合、各ノードの分散レンダリング処理結果を一つに連結する際に、分割面近傍のレンダリング処理を補う必要があった。

【0011】

本発明の目的は、ボクセルデータ分割による分散処理で得られた分散レンダリング処理結果を一つに連結する際に、分割面近傍のレンダリング処理を補う計算処理が伴わずに済み、高コストパフォーマンスを満たす単一計算機では扱うことが難しい大容量ボクセルデータでも高速にボリュームレンダリング処理することができるボリュームレンダリング処理方法、ボリュームレンダリング処理システム、計算機及びプログラムを提供することにある。

【課題を解決するための手段】

【0012】

上記目的を達成するために請求項1に記載の発明は、複数のノードによってコンピュータネットワークが構築され、三次元以上のボクセルデータに施すべきボリュームレンダリング処理を少なくとも二のノードが分散処理で実行するボリュームレンダリング処理システムにおいて、少なくとも一のノードに設けられ、各ノードの計算余力及びメモリ空き状況に係る計算資源使用状況情報を基に分散処理可能な余力のある少なくとも二のノードを割り当てるノード割当手段と、少なくとも一のノードに設けられ、与えられたボクセルデータを、各割り当てノードの計算資源使用状況情報に応じた分割割合で、かつ分割面に関わるボクセルデータを重複させるように複数の分割ボクセルデータに分割するデータ分割割当部と、前記分割ボクセルデータを動的に前記各割り当てノードに割り当てる負荷割当手段と、前記割り当てノードとなる少なくとも二のノードに設けられ、割り当てられた分割ボクセルデータに対して分散レンダリング処理を行うとともに、少なくとも分割面近傍における分散レンダリング処理の補間計算を前記データの重複部分を用いて行う演算手段と、少なくとも一のノードに設けられ、前記各割り当てノードにおける分散レンダリング処理結果を取得して一つに整合させて連結する連結手段と、少なくとも一のノードに設けられ、前記連結されたレンダリング処理結果に基づくボリュームレンダリング画像を表示手段に表示する表示制御手段とを備えたことを要旨とする。

【0013】

この発明によれば、少なくとも一のノードにおいて、ノード割当手段が各ノードの各ノードの計算余力とメモリ空き容量とを含む計算資源使用状況情報を基に分散処理可能な余力のあるノードを、分散処理を行うノードとしてノード割り当てをする。少なくとも一のノードに設けられたデータ分割割当部は、各割り当てノードの計算資源使用状況情報に応じて、与えられたボクセルデータの分割割合を決定するとともにその分割割合でボクセルデータを分割するに当たり、分割面に関わるボクセルデータを重複させるようにボクセルデータを分割する。割り当てノードとなる少なくとも二のノードに設けられた演算手段は、割り当てられた分割ボクセルデータに対して分散レンダリング計算を行うとともに、少なくとも分割面近傍においては分散レンダリング処理をデータの重複部分も用いて行う。この結果、各割り当てノードにおける分散レンダリング処理を分割面まで行うことができる。そして、少なくとも一のノードに設けられた連結手段が、各割り当てノードにおける分散レンダリング計算結果を取得して整合させて一つに連結する。このとき、ボクセルデ

10

20

30

40

50

ータ分割による分散処理で得られた分散レンダリング処理結果を一つに連結する際に、分割面近傍のレンダリング処理を補う計算処理が伴わずに済む。そして、少なくとも一のノードに設けられた表示制御手段は、連結されたレンダリング計算結果に基づくボリュームレンダリング画像を表示手段に表示する。従って、計算資源に余力のあるノードがあればその計算資源を有効活用し、複数のノード（計算機）が動的に負荷を分担しながらボリュームレンダリング演算を協調分散処理することで、単一計算機（ノード）では扱うことが難しい大容量データでも高速にボリュームレンダリング処理することが可能となる。

【0014】

請求項2に記載の発明は、請求項1に記載の発明において、前記分割面に関わるボクセルデータを重複させるとは、前記分割面近傍における補間計算が可能な量を重複させることをいうことを要旨とする。

10

【0015】

この発明によれば、各割り当てノードには、補間計算が可能な量を重複させた分割ボクセルデータが割り当てられる。このため、各割り当てノードの演算手段が、分割面近傍における補間計算を行うことができる。

【0016】

請求項3に記載の発明は、請求項2に記載のボリュームレンダリング処理システムにおいて、前記補間計算は、グラディエント計算又はボクセル補間値計算であることを要旨とする。

【0017】

この発明によれば、請求項2に記載の発明の作用に加え、各割り当てノードは、分割ボクセルデータの分割面近傍におけるグラディエント計算又はボクセル補間値計算が可能となる。

20

【0018】

請求項4に記載の発明は、請求項1～3のいずれか一項に記載のボリュームレンダリング処理システムにおいて、前記分割ボクセルデータに対する計算負荷を、前記分割ボクセルデータの割り当て先の各ノードの計算資源使用状況情報に応じた分割割合で分割する計算分割割当部を更に備え、前記負荷割当手段は、前記計算分割割当部が分割した計算負荷を前記割り当て先の各ノードに割り当ててを要旨とする。

【0019】

この発明によれば、請求項1～3のいずれか一項に記載の発明の作用に加え、計算分割割当部は、分割ボクセルデータの割り当て先の各割り当てノードにおける計算負荷を、計算資源状況使用情報に応じた分割割合で分割する。負荷割当手段は、計算分割割当部が分割した分割レンダリング計算を各割り当てノードに割り当てて。よって、複数のノードに同じ分割ボクセルデータが割り当てられ、各ノード間で分割ボクセルデータに対するレンダリング計算が分散処理で行われる。

30

【0020】

請求項5に記載の発明は、請求項1～3のいずれか一項に記載のボリュームレンダリング処理システムにおけるノードを構成する計算機であって、前記ノード割当手段、前記データ分割割当部及び前記負荷割当手段を備えたことを要旨とする。

40

【0021】

この発明によれば、計算機（コンピュータ）は、ノード割当手段、データ分割割当部及び負荷割当手段を備えるので、この計算機を、ボリュームレンダリング処理システムに用いることで、請求項1～6のいずれか一項に記載の発明と同様の作用効果が得られる。

【0022】

請求項6に記載の発明は、複数のノードによって構築されるコンピュータネットワークを用いて、少なくとも二のノードが、三次元以上のボクセルデータに施すべきボリュームレンダリング処理を分散処理で実行するボリュームレンダリング処理システムにおける前記ノードの機能をコンピュータに実現させるプログラムであって、コンピュータを、各ノードの計算余力とメモリ空き状況に係る計算資源使用状況情報に基づき分散処理を割り当

50

て可能な余力のある少なくとも二のノードを割り当てるノード割当手段と、与えられたボクセルデータを、前記各割り当てノードの計算資源使用状況情報に応じた分割割合で、かつ分割面に関わるボクセルデータを重複させるように複数の分割ボクセルデータに分割するデータ分割割当部と、前記分割ボクセルデータを動的に前記各割り当てノードに割り当てる負荷割当手段と、割り当てられた分割ボクセルデータに対して分散レンダリング処理を行うとともに、少なくとも分割面近傍における分散レンダリング処理の補間計算を前記データの重複部分を用いて行う演算手段と、前記各割り当てノードにおける分散レンダリング処理結果を取得して一つに整合させて連結する連結手段と、前記連結されたレンダリング処理結果に基づくボリュームレンダリング画像を表示手段に表示する表示制御手段、として機能させるプログラムであることを要旨とする。

10

【0023】

この発明によれば、プログラムをコンピュータに実行させることにより、請求項1に記載のボリュームレンダリング処理システムを構成するノードとして機能させることができ、このノードでシステムを構築することにより、請求項1に記載の発明と同様の作用効果が得られる。

【0024】

請求項7に記載の発明は、請求項6に記載のプログラムにおいて、前記分割面に関わるボクセルデータを重複させるとは、補間計算が可能な量を重複させることをいうことを要旨とする。

【0025】

この発明によれば、プログラムをコンピュータに実行させることにより、請求項2に記載のボリュームレンダリング処理システムを構成するノードとして機能させることができ、このノードでシステムを構築することにより、請求項2に記載の発明と同様の作用効果が得られる。

20

【0026】

請求項8に記載の発明は、請求項7に記載のプログラムにおいて、前記補間計算は、グラディエント計算又はボクセル補間値計算であることを要旨とする。

この発明によれば、プログラムをコンピュータに実行させることにより、請求項3に記載のボリュームレンダリング処理システムを構成するノードとして機能させることができ、このノードでシステムを構築することにより、請求項3に記載の発明と同様の作用効果が得られる。

30

【0027】

請求項9に記載の発明は、請求項6～8のいずれか一項に記載のプログラムであって、前記分割ボクセルデータに対する計算負荷を、前記分割ボクセルデータの割り当て先の各ノードの計算資源使用状況情報に応じた分割割合で分割する計算分割割当部としてもコンピュータを機能させ、前記負荷割当手段は、前記計算分割割当部が分割した計算負荷を前記各割り当て先の各ノードに割り当てることを要旨とする。

【0028】

この発明によれば、プログラムをコンピュータに実行させることにより、請求項4に記載のボリュームレンダリング処理システムを構成するノードとして機能させることができ、このノードでシステムを構築することにより、請求項4に記載の発明と同様の作用効果を得ることができる。

40

【0029】

請求項10に記載の発明は、複数のノードによって構築されるコンピュータネットワークを用いて、三次元以上のボクセルデータに対して少なくとも二のノードが分散処理でボリュームレンダリング処理を施すボリュームレンダリング処理方法であって、前記複数のノードのうち少なくとも一のノード割当手段が、各ノードの計算余力及びメモリ空き状況に係る計算資源使用状況情報に基づいて分散処理可能な余力のある少なくとも二のノードを割り当てる段階と、少なくとも一のノードのデータ分割割当部が、与えられたボクセルデータを、各割り当てノードの計算資源使用状況情報に応じた分割割合で、かつ分割面に

50

関わるボクセルデータを重複させるように複数の分割ボクセルデータに分割する段階と、少なくとも一のノードの負荷割当手段が、前記分割ボクセルデータを動的に前記各割り当てノードに割り当てる段階と、前記割り当てノードの演算手段が、割り当てられた分割ボクセルデータに対して分散レンダリング処理を行うとともに、少なくとも分割面近傍における分散レンダリング処理の補間計算を前記データの重複部分を用いて行う段階と、少なくとも一のノードの連結手段が、前記各割り当てノードにおける分散レンダリング処理結果を取得して一つに整合させて連結する段階と、少なくとも一の表示制御部が、前記連結されたレンダリング処理結果に基づくボリュームレンダリング画像を表示手段に表示する段階とを備えたことを要旨とする。

【0030】

この発明によれば、請求項1に記載のシステムの発明と同様の作用効果が得られる。

【発明の効果】

【0031】

請求項1～10に記載の発明によれば、ボクセルデータを分割して複数のノードに割り当てて分散処理を行う場合に、ボクセルデータを分割面に関わるボクセルデータを重複させるように分割し、分割面近傍でも補間計算が可能となるように分割データに冗長性を持たせるので、各割り当てノードにおける分散レンダリング処理を分割面まで行うことができる。よって、ボクセルデータ分割による分散処理で得られた分散レンダリング処理結果を一つに連結する際に、分割面近傍のレンダリング処理を補う計算処理が伴わずに済み、高コストパフォーマンスを満たす単一計算機では扱うことが難しい大容量ボクセルデータでも高速にボリュームレンダリング処理することができる。

【発明を実施するための最良の形態】

【0032】

以下、本発明を具体化した一実施形態を図1～図15に従って説明する。

図1に示すボリュームレンダリング協調分散処理システム1は、例えば病院内に配備されている。病院内の複数箇所には、CT、MRIなどの医療画像を見るための複数台（ここでは3台）の画像処理システムA、B、Cが設置されている。これらの画像処理システムA、B、Cが医療画像データ（ボクセルデータ）を記憶（蓄積）するサーバー（ストレージサーバー）2と通信線を介して接続され、コンピュータネットワークとしての院内ネットワーク（LAN）が構築されている。

【0033】

画像処理システムA、B、Cは、コンピュータ（例えばワークステーション）によって構成されており、計算機（ノード）としてのコンピュータ本体3、表示装置4および入力装置5をそれぞれ備えている。ストレージサーバー2には、CT、MRIなど医療画像診断機器で撮影された医療画像データ（ボクセルデータ）が蓄積されている。医師は画像処理システムA、B、Cの画面を通してストレージサーバー2に蓄積された例えば患者の医療画像を見ることができる。

【0034】

医療画像データ（ボクセルデータ）は、CT、MRIなど医療画像診断機器の発達による測定分解能の向上とともに、そのデータ容量が増大している。医療画像データのデータ容量増大の要因として診断機器の空間分解能及び時間分解能の向上が挙げられる。空間分解能向上としては、(1)X-Y軸方向：現在は1スライスの解像度は 512×512 が主流であるが、いずれ 1024×1024 になると言われている、(2)Z軸方向：1スライスの厚さが薄くなることで、物体を構成するスライス枚数が従来よりも多くなる、などがある。また時間分解能向上としては、(1)マルチスライスCTという技術により、単位時間内で撮影できるスライス枚数が増えているため、短時間でより多くのスライス画像を得ることができる、(2)(1)と撮影時間の短縮により、心臓の拍動など器官の動きを3次元で撮影することが容易になりつつある、などがある。

【0035】

医療画像データ（ボクセルデータ）を表示装置4の画面で見るときの画像処理にはボリ

10

20

30

40

50

ュームレンダリング処理が行われる。リアルタイムでデータ本来の分解能を反映させながらデータを間引かずにポリウムレンダリング処理を行うために、複数の計算機で分散処理を行う。各計算機3は計算資源に余裕がある場合、別計算機3のプロセスに計算資源を提供することができるため、ユーザーが直接使用していない場合にも計算資源が有効に活用される。また、冗長性を持たせることによる高可用性を実現する。

【0036】

ここで、「ポリウムレンダリング」とは、ポリウムデータ(volume data, volumetric data)を2次元(2D)画像に表示する方法である。視点位置から視線方向に向かってレイキャスティングを行い、画像を生成する。通常ポリウムデータは、3次元物体をサンプリングした三次元ボクセルデータを指すことが多いが、本実施形態ではボクセルデータの次元を3次元に限定しない。特に3次元と4次元のボクセルデータ(空間軸3+時間軸1)を想定している。一般的に用いられる4Dボクセルデータは、3Dボクセルデータを時系列にしたものである。従って、4Dボクセルデータに対するポリウムレンダリング処理は、一般にはアニメーション(3Dボクセルデータに対するポリウムレンダリングを時系列に従って高速に行う)を指す場合が多い。ボクセルデータの次元としては、3次元と4次元に留まらず、さらに5次元以上であっても構わない。4Dデータのアニメーション処理は、多次元ポリウムレンダリング処理の視点から見ると、n次元データの異なる(n-1)次元断面を高速に次々表示する処理に対応する。また「ボクセルデータ」とは、3次元以上の離散的な配列データであって、2D配列データのピクセルデータに対応するものである。

10

20

【0037】

ポリウムレンダリングの種類としては、各ボクセルに不透明度を設定してレイキャスティングを行う方法(一般にポリウムレンダリングはこれを指す場合が多い)、MIP(Maximum Intensity Projection)、MinIP(Minimum Intensity Projection)、RaySum等が挙げられる。

【0038】

また、応用範囲は多岐に渡り、使用するデータがボクセルデータであればデータ取得方法には特によらない。例えば、医療分野の他、顕微鏡解析、気象解析、地質調査(石油探査など)、非破壊検査などが挙げられる。なお、5次元以上のボクセルデータには、例えば分子間力の解析(結合角)などが挙げられる。

30

【0039】

本システム1の特徴は、ボクセルデータのストレージ(保管)を行うストレージサーバー2と、ポリウムレンダリング処理・表示を行う画像処理システムA、B、Cとに分けた点にある。複数台の画像処理システムA、B、Cは互いに協調して計算を行い、互いに対等な関係にある。従来技術で述べた図17、18の手法において計算資源を提供する計算機が一台であるのに対し、本実施形態による新規手法では計算資源を複数台の計算機3が提供する。各計算機3には、協調分散処理対応の同一のプログラムがそれぞれインストールされている。

【0040】

図1に括弧()で示した数字は、ポリウムレンダリング協調分散処理の流れを示すものである。以下、処理の手順について順に説明する。なお、医師(ユーザー)が画面上で医療画像を見るために入力装置5が操作される画像処理システムが「メイン」となる。ここでは画像処理システムAをメインとする。

40

【0041】

(1) データリスト要求

メインの画像処理システムの入力装置5を操作して医療画像を選択するためのデータリストをストレージサーバー2に対して要求する。

【0042】

(2) データリスト取得

上記データリスト要求に回答してストレージサーバー2から送られてくるデータリストを

50

取得する。

【0043】

(3) ノード割り当て

データリスト取得によって画像表示の指示が予想されると、メインとなる画像処理システムAは、ネットワーク上の他の画像処理システムの計算資源利用状況を確認する。この計算資源利用状況確認のため本実施形態では、その情報としてメモリ空き容量およびCPU使用率を取得する。そしてCPU使用率から例えばCPU余力(=(1-CPU使用率)×CPUクロック周波数)を求める。そしてCPU余力とメモリ空き容量とを基に処理能力(計算資源)上の余力が必要量ある他ノードの存在の有無を判断し、該当する他ノードが存在すればそれを分散処理に加わる画像処理システムとして決定する。つまりメインノード3と共に分散処理に加わるサブノード3をその余力(処理能力の空き)に応じて決定し、分散処理を担当する各ノードを割り当てる。

10

【0044】

ここで、CPU余力とは、上記計算式からCPU使用率を用いて算出されたものに限定されず、CPUの種類・性能の違いの要素を考慮して決定されるものであってもよい。またノード割り当て前などにベンチマークを行い、各ノードの処理能力(又は余力)を計測する方法を採用しても構わない。さらにCPU性能やメモリ性能の違いを考慮してCPU余力とメモリ空き容量とから決まるトータルパフォーマンスを考慮して計算資源の余力を決める方法を採用することも可能である。同図では、画像処理システムAの他、画像処理システムB、Cが協調分散処理のノードとして選択されている。なお、本実施形態では、各ノード3からメモリ使用率の情報を通信で取得し、メモリ空き容量(=(1-メモリ使用率)×メモリ容量)を求めている。特に本実施形態では全てのノード3のメモリが同一性能(種類)・同一容量であることから、メモリ空き容量をメモリ空き率(=1-メモリ使用率)から判断している。もちろん、メモリ使用率そのものからメモリ空き容量を判断してもよい。

20

【0045】

メインとなる画像処理システムA(メインノード3)は、他の画像処理システムB、C(サブノード3)の動作状況を確認しながら各々の処理能力に応じて、協調分散処理を担当する各画像処理システムA、B、C(各割り当てノード3)のそれぞれに割り当てるべきデータ負荷と計算負荷の分担割合を決める。つまり、各ノード3のその時々々の処理能力に応じて割り当て負荷を決める動的負荷割り当てを行う。なお、メインノードはユーザーに使用されている表示対象指定元のノードであり、サブノードはユーザーが利用しておらずその計算資源の空きを利用して分散処理が行われるノードである。

30

【0046】

動的負荷割り当ての手法として、本実施形態では、データコピーモード、データ分割モード、混合モードの3モードが用意されている。データ分割モードあるいは混合モードの場合では、単にノードを割り当てるだけでなく、さらに各ノードに対するボクセルデータの分割割合を決定する。以下、これら3モードについて詳しく説明する。

【0047】

(a)「データコピーモード」では、ボクセルデータの容量がそれほど大きくなければ全ての計算機(ノード)3にボリュームレンダリングの対象となるボクセルデータを分割せずまるごとコピーする。単一計算機による処理性能の限界を突破することを目的とする。しかし、後述する(b)のモードに比べ扱えるボクセルデータの容量が制限される。

40

【0048】

(b)「データ分割モード」では、ボクセルデータの容量が大きい場合、ボリュームレンダリングの対象となるボクセルデータを分割し、各計算ノードに割り当てる。単一計算機による処理容量の限界を突破することを目的とする。しかし、処理速度は(a)のモードに比べ多少遅くなる。

【0049】

(c)「混合モード」とは、上記(a)、(b)両方の特徴を持つモードである。例えばボク

50

セルデータを2つに分割し、それぞれ2つの分割データコピーを用いれば、4台の計算機（ノード）3がボリュームレンダリング処理を行う。この場合、データ（メモリ負荷）と計算処理（計算負荷）の両方が分割され、2データ分割×2データコピーによって4台の計算機（ノード）3に負荷が割り当てられる。

【0050】

(4) データ選択

データ分割モード、混合モードの場合は、上記(3)で決定したボクセルデータの割り当ての情報も転送される。サーバー2は割り当ての情報を解釈し、ボクセルデータの分割転送を行う機能を持っている。

【0051】

(5) データ転送・メモリ読み込み

データコピーモードでは、全ての割り当てノード3にボクセルデータ全体を転送する。データ分割モードでは、分割したボクセルデータを対応する各ノード3に転送する。このモードでは、各ノード3に割り当てられる分割データがそれぞれ異なる。混合モードでは、分割したボクセルデータを対応する各ノード3に転送するが、複数のノード3に同じ分割データが重複して割り当てられる。

【0052】

(6) 分散ボリュームレンダリング処理要求

前記(3)で割り当てたノード3に対し前記(4)で決定したモードに応じて割り当てた動的負荷の計算を行わせる分散ボリュームレンダリング処理を要求する。

【0053】

(7) 分散ボリュームレンダリング処理

各ノード3において分散ボリュームレンダリング処理が実行される。

(8) 処理結果転送

各サブノード3で行われた分散ボリュームレンダリング処理の処理結果がメインノード3に転送される。

【0054】

(9) 分散処理結果の連結

各ノードから転送された分散ボリュームレンダリング処理の処理結果を連結する処理が行われる。

【0055】

(10) 表示

前記(9)で連結された処理結果に基づいて画像データを生成し、表示装置4の画面に表示させる。

【0056】

(6)～(10)の処理がループで行われる。ボクセルデータの割り当ては処理開始時の他、各画像処理システムA、B、Cの利用状況が変わった場合に実行される。

図2は、ボリュームレンダリング協調分散処理システムの機能ブロック図を示す。計算機3には、CPU、メモリ及びプログラムによって、同図に示す各機能部10～13が構築されている。すなわち、要求処理部10、割当処理部11、分散ボリュームレンダリング処理部12およびデータ連結処理部13を備えている。またメモリにより構成されるボクセルデータ格納部14および画像メモリ15が備えられている。また画像メモリ15に格納された画像データを基に表示装置4の画面に画像を表示させる表示制御部16が備えられている。なお、要求処理部10により要求手段が構成される。

【0057】

割当処理部11は、ノード割当部17と負荷割当部18とから構成されている。負荷割当部18はさらにモード決定部19、データ分割割当部20および計算分割割当部21を備えている。割当処理部11は他のノード3との通信によって他のノード3のCPU使用率およびメモリ空き容量を含む計算資源利用状況情報を取得する。

【0058】

10

20

30

40

50

要求処理部 10 は、入力装置 5 の操作によりデータリスト要求を受け付けると、ストレージサーバ 2 に対しデータリストを要求し、データリストを取得すると、表示制御部 16 を介して表示装置 4 の画面上にリストを表示させる。リストデータにはその一覧中の医療画像に関するボクセルデータのデータ容量を示す情報が含まれている。画面上に表示されたリスト中からユーザー（医師）が入力装置 5 を操作して例えば患者の医療画像（例えば心臓 CT 画像等）を選択すると、要求処理部 10 は、リストから選択指定された医療画像のボクセルデータをストレージサーバ 2 に対し要求する処理を行う。このボクセルデータ要求処理に先立ち要求処理部 10 はリストから選ばれた医療画像に関するボクセルデータのデータ量情報を割当処理部 11 に送る。そして割当処理部 11 では、ノード割当部 17 がボクセルデータ量情報に基づき協調分散処理の必要性判断を行い、協調分散処理が必要と判断したときには、ノード割り当てが可能であるかどうかを判断する。そしてノード割当部 17 は、協調分散処理必要かつノード割り当て可能である場合、ボクセルデータ量情報と各ノード 3 における計算資源利用状況情報（CPU 使用率、メモリ空き容量）とに基づき、協調分散処理を担当するノードを割り当てるノード割当処理を実行し、割り当てノードを決定する。また割り当てノードが決定すると、次に負荷割当部 18 が各ノード 3 に対して負荷（データ負荷・計算負荷）を割り当てる負荷割当処理が実行され、各ノード 3 に対する負荷割り当て（データ負荷割当・計算負荷割当）を決定する。ノード割り当て・負荷割り当ての決定情報は要求処理部 10 に渡され、要求処理部 10 はこれらの情報を付してストレージサーバ 2 に対しボクセルデータ要求を出す。このように要求処理部 10 は、図 1 における「(1) データリスト選択」および「(4) データ選択」を司る。

【0059】

ノード割当部 17 は、他の計算機（ノード）から通信によって入手した他の各計算機 3 の CPU 使用率およびメモリ空き容量の情報に基づき協調分散処理を行うノードを割り当てる。ノード割当部 17 は、図 1 における「(3) ノード割り当て」を司る。

【0060】

負荷割当部 18 は、協調分散処理のため選択された各ノードに対しリストから指定されたボクセルデータの容量や計算負荷（表示条件等から決まる計算負荷）に応じて、各ノード 3 の CPU 使用率およびメモリ空き容量を考慮しつつ各ノード毎のデータ負荷および計算負荷を割り当てる。すなわちモード決定部 19 が動的負荷割り当て手法の違いによって 3 モードで設定された、「データコピーモード」、「データ分割モード」、「混合モード」のうちいずれを採用するかを決定する。

【0061】

データ分割割当部 20 は、データ分割モードまたは混合モードにおいてボクセルデータの分割割合を決定する。また計算分割割当部 21 は、データコピーモードまたは混合モードにおいて計算負荷分割処理を行う。混合モードでは、データ分割割当部 20 と計算分割割当部 21 との協働によって、データ分割処理と計算負荷分割処理が行われる。また、要求処理部 10 は割り当てノード 3 に対して割り当て負荷内容を指示する分散ボリュームレンダリング処理要求を行う。

【0062】

ストレージサーバ 2 は、指定されたボクセルデータを必要に応じて指示に従い分割して割り当て先の各ノード 3 に転送する。ノード 3 に転送されたボクセルデータはボクセルデータ格納部 14 に格納される。分散ボリュームレンダリング処理部 12 は、ボクセルデータ格納部 14 に格納された割り当てデータに対し割り当てられた計算処理を行う分散ボリュームレンダリング処理を実行する。

【0063】

各ノード 3 における分散ボリュームレンダリング処理結果は通信によりメインのノード 3 に転送される。データ連結処理部 13 は、自身のノード 3 の分散ボリュームレンダリング処理部 12 からの処理結果と各ノード 3 から転送された分散ボリュームレンダリング処理結果を一つに連結する。連結されることで生成された画像データは画像メモリ 15 に送られる。表示制御部 16 は画像メモリ 15 に格納された画像データを基に表示装置 4 の画

面に画像を表示する。

【0064】

図3は、計算負荷割り当て手法を説明するものである。ここでは512×512ピクセルのボリュームレンダリング画像を作成する場合、512×512=262,144本のレイをキャストするため、処理を各レイごとの区画に分割している。

【0065】

計算処理を複数の処理区画に分割し、各ノードに割り当てる処理区画を決定する。各ノード3はそれぞれに割り当てられた処理区画に従い計算処理を行う。一定数(または一定割合)の処理区画の計算処理が終了したら、処理結果をメインとなる画像処理システムAへ返す。処理結果を返すにあたり、必ずしも割り当てられた処理区画の処理が全て終了するのを待たなくてもよい。そのため、全ての処理が終わり、次の割り当てがくるまで待機するようなことがない。メインとなる画像処理システムAは、処理結果を返してきたノード3に、次の処理区画を割り当てる。例えばメインの画像処理システムAが画像処理システムBに対し区画1-2000の処理要求を行う。これに対して画像処理システムBは区画1-1000の処理を終了するとそこまでの処理結果を画像処理システムAに返す。画像処理システムAは区画1-1000の処理結果を受け取ると、画像処理システムBに対して次の区画2000-4000の処理要求をする。結果として処理結果が速いノード3ほど多くの区画が割り当てられることになり、それぞれのノード3の能力に見合った計算負荷を動的に割り当てることができる。

10

【0066】

図4,図5は本実施形態の協調分散処理システムの特徴的な作動状況を説明するものである。図4に示すように例えば4台の画像処理システムA~Dのうち、メインの画像処理システムCの1台のみ医師が使用していると、使用されていない他の画像処理システムA, B, Dの各計算機3を有効活用して協調分散処理で計算処理(ボリュームレンダリング処理)を行う。この結果、メインの画像処理システムCではデータを間引くことなくリアルタイムの速度で大容量医療画像データのボリュームレンダリング画像を見ることができる。

20

【0067】

また図5に示すように、複数台の画像処理システムA, C, Dを医師が使用していると、このとき例えばピクセルデータ容量の多い医療画像を扱うメインの画像処理システムCは、使用されていない画像処理システムBや使用されていても計算負荷に余裕のある画像処理システムDに対し計算処理(ボリュームレンダリング処理)を分担させる。この結果、メインの画像処理システムA, C, Dが複数台であっても、負荷の大きい計算処理を他の画像処理システムB, Dの計算資源を有効活用して協調分散することによりデータを間引くことなくリアルタイムの速度でボリュームレンダリング画像を見ることができる。なお、各ノード3はローカルストレージを持っており、過去に使用した割り当てデータを既に持っているときはサーバー2からノード3にデータを転送することはせず、既に持っているデータを用いて計算処理を行うようになっている。

30

【0068】

図6,図7は、ノード割り当ておよび負荷割り当ての変更処理を説明するものである。ノード割り当ておよび負荷割り当ては、処理開始時の他、処理中にも行われる場合がある。再割り当てのトリガーとなる状況が図6,図7にそれぞれ示す2つある。

40

【0069】

(A)ユーザーが直接使用していないが、分散処理を割り当てられているサブノードを、新たなユーザーが使用しようとした場合(図6)と、(B)ユーザーが処理を終了し、1つの分散処理に携わっていたノードが開放された場合(図7)である。

【0070】

まずノード割り当てについて説明する。割り当てノードには、CPU使用率とメモリ空き容量の利用状況から、CPUとメモリにある程度の空きがある必要がある。CPUの空きは計算処理(ボリュームレンダリング処理)を行うために必要で、これに基づき計算負

50

荷割り当て量（分散ボリュームレンダリング処理割当量）が決まる。一方、メモリの空きはボクセルデータを格納するために必要で、これに基づきボクセルデータ割当量が決まる。ここで、ユーザーが使うノードを「メインノード」、ユーザーに使われていないが処理が割り当てられるノードを「サブノード」と呼ぶことにすると、上記（A）、（B）の状況は、メインノードの数が変わった状況と考えることができる。以下、計算負荷割り当ておよびボクセルデータ割り当ての変更処理について説明する。

【0071】

(1) 計算負荷割り当て

計算負荷割り当てはノードの利用状況に対応できる。図6に示すように、ユーザーがないサブノード（例えば画像処理システムBの計算機）3を、ユーザーが使い始めた場合は、そのサブノード3に割り当てられる計算負荷が軽減される。一方、図7に示すように、ユーザーがいたメインノード（例えば画像処理システムBの計算機）3から、ユーザーがいなくなる場合は、そのサブノードとなったノード3に割り当てられる計算負荷が大きくなる。

10

【0072】

(2) ボクセルデータ割り当て

ボクセルデータの割り当てはメモリ空き容量に対応するように行われる。図6に示すように、ユーザーが直接使用していないが、分散処理を割り当てられているサブノード（例えば画像処理システムBの計算機）3を、新たなユーザーが使用しようとした場合を考える。このとき、その空きメモリ容量が十分でない場合は新しい処理にメモリを確保するために、既に割り当てられている処理で使用されているメモリを減らさなければならない。新しい処理開始時に画像処理システムBは他の画像処理システムA、Cにメッセージを通知する。メッセージを受信した既存処理のメインノード（メイン1）は、メッセージを送信したノード（画像処理システムBの計算機）3に対するボクセルデータの割り当てを減らし、減らした分を他のノード（画像処理システムCの計算機）3に振り分ける。この結果、画像処理システムBの計算機3のメモリのうちメイン1の画像処理システムAの既存処理に使用されていたメモリ容量「M1」が減り、メイン2の画像処理システムBで新しく開始される処理に使用するメモリ容量「M2」が確保される。

20

【0073】

一方、図7に示すように、メイン1の画像処理システムAを使用していたユーザーが処理を終了し、1つの分散処理に携わっていたメインノード（例えば画像処理システムAの計算機）3が開放された場合、終了した処理が使用していたメモリが開放されるため、既存処理はそのメモリを活用することができる。すなわち、画像処理システムAはノード開放時点に他の画像処理システムB、Cにメッセージを通知する。メッセージを受信した既存処理のメインノード（画像処理システムBの計算機）3は、必要であればメッセージを送信したノード（画像処理システムAの計算機）3の空きメモリ容量を利用してボクセルデータを割り当てる。この結果、メイン2の画像処理システムBにおけるノード割り当て数が増え、大容量・高速処理に対処できる。

30

【0074】

次にボリュームレンダリング処理の詳細について説明する。

図8(a)、(b)に示すように、ボリュームレンダリング処理には、「平行投影法」と「透視投影法」とがある。平行投影法の方が計算負荷が軽いため、通常この投影法がよく用いられる。透視投影法では内視鏡から得られるような画像を生成するため、血管内部など組織内の様子を観察したい場合に用いられる。

40

【0075】

図8に示すように、ボクセルデータVDは3次元（但し同図では2次元的に描かれている）の格子点にボクセル値（例えばCT値）を持つデータである。ボリュームレンダリングには一般的にレイキャスティング法が用いられる。レイキャスティング法とは、観察する側（フレーム側）から光の経路を考えるもので、フレーム側のピクセルから光のレイを飛ばし、一定距離を進むごとにその位置での反射光を計算する（図8では「... , V_{n-1} , V_n , V_{n+1} , V_{n+2} , ...」の符号が各到達位置のボクセルに対応している）。レイ到達

50

位置が格子にない場合はその周りのボクセルのボクセル値から補間処理を行ってその位置でのボクセル値を計算する。

【0076】

図9は、レイキャスティング法の計算方法を説明するもので、図8における1本のレイに対応した処理である。同図におけるブロックはボクセルに相当するもので、これら各ボクセルは光に対する特性パラメータとして不透明度(opacity) α_n およびシェーディング係数 β_n を有する。ここで、不透明度 α_n は、 $0 \leq \alpha_n \leq 1$ を満たす数値で表され、値 $(1 - \alpha_n)$ は透明度(transparency)を示す。不透明度 $\alpha_n = 1$ は不透明、 $\alpha_n = 0$ は透明、 $0 < \alpha_n < 1$ は半透明にそれぞれ対応する。シェーディング係数は色、グラディエントなどシェーディングに関する情報を持つ。

10

【0077】

初期入射光(光線) I_1 は、各ボクセルを順次透過してゆくとともに各ボクセルで一部反射および吸収されることによりその残存光(透過光)は徐々に減衰する。各ボクセルにおける部分反射光 R_n ($n = 1, 2, \dots$) の積算値(積算反射光)が、フレーム側におけるピクセルの輝度に相当する。ここで、減衰光 D_n ($n = 1, 2, \dots$) は、 n 番目のボクセルの入射光 I_n を用いて、式 $D_n = \alpha_n I_n$ で表されるため、部分反射光 R_n は、式 $R_n = \beta_n D_n = \beta_n \alpha_n I_n$ で表される。また各ボクセルにおける入射光と残存光(透過光)との関係式から、式 $I_{n+1} = (1 - \alpha_n) I_n$ が成り立つ。よって積算反射光 E は、次式により表される。

【0078】

【数1】

$$E = \sum_{i=1}^n R_i = \beta_1 \alpha_1 I_1 + \beta_2 \alpha_2 I_2 + \dots + \beta_n \alpha_n I_n$$

$$= \sum_{i=1}^n \beta_i \alpha_i I_i = I_1 \sum_{i=1}^n \left\{ \beta_i \alpha_i \left[\prod_{j=1}^{i-1} (1 - \alpha_j) \right] \right\}$$

$$* I_i = I_1 \prod_{j=1}^{i-1} (1 - \alpha_j), \quad \text{但し } i=1 \text{ のとき } \prod_{j=1}^0 (1 - \alpha_j) = 1 \text{ とする}$$

20

30

なお、各ボクセル値に対して不透明度 α_n との関係付けが予めなされており、その関係付け情報に基づきボクセル値から不透明度 α_n を得ている。例えば、血管のボリュームレンダリング画像を得たい場合、血管のCT値が0から200に分布していることから、ボクセル値0から200には不透明度1を対応させ、他のボクセル値には不透明度0を対応させることで、血管を表示することができる。

【0079】

次にボクセルデータ分割割り当ておよび計算負荷割り当てを実施したときの計算方法について説明する。

図10(a)はデータ分割割り当てを示すもので、同図(b)は計算負荷割り当てを示すものである。同図(a)のデータ分割はボクセルデータを2つに分割した例を示す。ボクセルデータVDは分割ボクセルデータVD1とVD2とに分割され、それぞれ別々のノード3に割り当てられる。分割ボクセルデータVD1が割り当てられたノード3では、 $V_1 \sim V_k$ までのボクセルについて分散ボリュームレンダリング処理を行い、分割ボクセルデータVD2が割り当てられたノード3では、 $V_{k+1} \sim V_n$ までのボクセルについて分散ボリュームレンダリング処理を行う。

40

【0080】

また図10(b)の計算負荷分割は、割り当てられたボクセルデータに対して行うべき計算処理を複数に分割し、それぞれ別々のノード3に割り当てる。計算負荷分割の仕方としては光線(レイ)の本数で分割する方法が挙げられるが、光線の光軸方向(奥行き方向

50

)に計算を分割しても構わない。例えば図3に挙げた処理区画で分割する方法は、光線の本数で分割した計算負荷分割である。フレーム側のピクセルがボリュームレンダリングを行う際の光線に相当するため、光線の本数で分割するこの計算負荷分割は、フレームをエリア分割した計算処理に相当する。なお、データ分割したときは、その分割に伴って計算処理も必然的に光軸方向に分割される。

【0081】

図11は、光線が光軸方向で分割された計算分割の場合における計算方法を説明するものである。使用されている記号の意味は図9と同じである。同図のように、符号「V1 ~ Vn」のボクセル群が、符号「V1 ~ Vk」のボクセル群と「Vk+1 ~ Vn」のボクセル群とに分割されたとすると、符号「V1 ~ Vk」のボクセル群に対する計算と、符号「V

10

【0082】

【数2】

$$E = \sum_{i=1}^n \beta_i \alpha_i I_i = \sum_{i=1}^k \beta_i \alpha_i I_i + \sum_{i=k+1}^n \beta_i \alpha_i I_i$$

$$= I_1 \sum_{i=1}^k \{ \beta_i \alpha_i [\prod_{j=0}^{i-1} (1 - \alpha_j)] \} + I_{k+1} \sum_{i=k+1}^n \{ \beta_i \alpha_i [\prod_{j=k+1}^{i-1} (1 - \alpha_j)] \}$$

20

$$* I_i = (1 - \alpha_{i-1}) I_{i-1} = (1 - \alpha_{i-1}) \cdots (1 - \alpha_{k+1}) I_{k+1}$$

$$= I_{k+1} \prod_{j=k+1}^{i-1} (1 - \alpha_j) \quad (i \geq k+1)$$

$$\text{但し } i=k+1 \text{ のとき } \prod_{j=k+1}^k (1 - \alpha_j) = 1 \text{ とする}$$

30

ここで、分割された計算処理の計算結果を連結するときには、符号「Vk+1 ~ Vn」のボクセル群に対する計算結果に初期入射光として入射光 I_{k+1} の条件を採用し、1つのデータとして連結したときに整合がとれるように、各分割計算処理結果の連結を行う。また図12に示すように、データ分割は、ボクセルデータVDを一部データが重複するように分割し、冗長性を持たせている。これは、光線があるボクセルに到達したときに格子上にないときのボクセル値を求める際の補間計算をするときなどに周囲のボクセルが必要になるからである。このように周囲のボクセルも必要になるその他の処理としては、ボリュームレンダリングにおけるグラディエントの計算方法(グレイレベルグラディエント)が挙げられる。このようにグラディエントやボクセル補間値などボリュームレンダリングに必要なパラメータを計算するために、注目座標の周囲のボクセルも必要になる。

40

【0083】

次に、各モードにおける計算負荷割り当ておよびデータ分割割り当ての詳細について説明する。図13はデータコピーモード、図14はデータ分割モード、図15は混合モードをそれぞれ示す。

【0084】

図13に示すようにデータコピーモードでは、ボクセルデータは各ノード3にまるごと割り当てられる。計算負荷割り当ての仕方によって同図(a)~(c)が挙げられる。同図(a),(b)は平行投影法、同図(c)が透視投影法である。同図(a)は光線(レイ)の本数で分割することにより計算分割したもので、光線の本数を4分割したそれぞれ

50

が4つのノード3に割り当てられる。同図(b)は多数本の光線(レイ)をデータの領域別に4分割することにより計算分割したもので、4分割したそれぞれの計算負荷が4つのノード3に割り当てられる。同図(c)は透視投影による多数本の光線(レイ)をデータの領域別に4分割することにより計算分割したもので、4分割したそれぞれの計算負荷が4つのノード3に割り当てられる。

【0085】

図14に示すようにデータ分割モードでは、ボクセルデータは複数分割され、各分割データVD1~VD4がそれぞれ異なる各ノード3に割り当てられる。つまりノード毎にデータ負荷が割り当てられる。データコピーモードの図13(b)における計算負荷割当では、ボクセルデータの回転時に計算負荷分割されるボクセルデータのエリアの変更が有り得るが、データ分割モードでは、ボクセルデータの回転時でも計算分割されるデータの分割面は変わらない。

10

【0086】

図15に示すように混合モードでは、ボクセルデータは複数分割され、各分割データVD1, VD2のうち同一データがそれぞれ異なるノード3に割り当てられる。さらに同一データが割り当てられた異なるノード3間で計算負荷割当が実施される。

【0087】

従って、この実施形態によれば以下の効果を得ることができる。

(1) 大容量のメモリと高速なCPUなどの膨大な計算資源を持つ計算機ではなく、現実的なコストで得られるコンピュータ(ワークステーション)を使用しても、ボリュームレンダリング処理を行うことができるボクセルデータの容量を著しく増やすことができる。すなわち、コストパフォーマンスの高い計算機1台あたりが装備可能なメモリ容量と計算能力に限界があっても、ボリュームレンダリング処理を行うことができるボクセルデータの容量にほぼ限界がない。従来はバッチ的な長時間処理、あるいは極めて高価な専用システムによる処理でしか扱うことができなかった大容量データを低コスト、リアルタイムで扱うことができる。

20

【0088】

(2) データ転送時に画像処理システムのローカル・ストレージに保存せずに、受信と同時に直接メモリに読み込むため、ローカル・ストレージの浪費がなく、事前のデータ転送も不要となる。

30

【0089】

(3) 計算資源(メモリ・CPU)に余力がある利用可能なノードをその時々に見つけて動的にノード割り当てを行うので、他ノードの計算資源を有効活用でき、高可用性を実現できる。

【0090】

(4) 画像表示処理受け付け時にノード数および負荷(データ負荷・計算負荷)を動的に割り当てることで動的にノードを活用できる。このため、時間帯により刻々と変化する運用形態・状況にも無駄なく、最適なオペレーション環境を全てのユーザーに対して常に提供することができる。

【0091】

(5) 動的負荷割り当ての種類として3モード用意し、CPU使用率やメモリ空き容量に応じて大容量データの高速度処理により適したノード割り当ておよび負荷割り当てができるようにしたので、大容量データの高速度処理をより適切に実現できる。すなわち、院内ネットワーク上の複数の計算機3の計算資源を有効活用できる。

40

【0092】

(6) 負荷割当部18は、負荷割り当てした分散ボリュームレンダリング処理のうち予め設定された所定量又は所定割合終了した処理結果を返してきたノードに対して、次の分散ボリュームレンダリング処理の負荷割り当てを行う(図3)。従って、処理速度の速いノード3に優先的に負荷が割り当てられるため、各ノード3間の協調分散処理を高速度に実現可能である。

50

【0093】

(7) メインノード3のメモリ容量やCPUの空きが必要なだけ確保される場合は、ボクセルデータを分割せずメインノード3に全て送り、メインノード3単独で(単独計算機で)ボリュームレンダリング処理が行われる。よって、不要な分散処理の実施によるボリュームレンダリング演算の処理速度低下を防ぐことができる。

【0094】

(8) サーバ2に対し要求されたボクセルデータのうち割り当てるべきデータを既にローカル・ストレージに持つノードに対しては、サーバ2からその割り当てデータを送信することなく、その割り当てノード3はローカル・ストレージに持つデータを用いてボリュームレンダリング処理を実行する。従って、割り当てノード3が既にローカルストレージに必要なボクセルデータを持つ場合は、サーバ2から割り当てノード3へのデータ送信時間を省き一層の高速処理を実現できる。

10

【0095】

(9) メインノード3の数に変更があると、ノード割当部17はノードの割り当てをし直すとともに、負荷割当部18は割り当てノード3に対する動的負荷割り当てをし直す。従って、メインノード3の数の増減により計算資源の空き度合いに変更が生じた時にノード割り当てと動的負荷割り当ての見直しが図られるので、他ノード3の処理を妨げずその計算資源を有効活用でき、高速なボリュームレンダリング処理を実行できる。

【0096】

(10) 複数のノード3はサブノード、メインノードに必要な機能を全て持っており、役割に応じてサブノード、メインノードと切り替わる。よって、どのノード3もメインノードになることができ、どのノード3の画面からもボリュームレンダリング画像を見ることができる。

20

【0097】

なお、実施形態は前記に限定されず、例えば次の態様に変更してもよい。

・ CPU使用率とメモリ空き容量のうち少なくとも一方を含む計算資源利用状況の情報から割り当てノードを決定すれば足りる。例えばCPU使用率のみ、あるいはメモリ空き容量だけをノード割り当てを決めるときに使う情報としてもよい。

【0098】

・ 前記実施形態では、3つのモードを設定し、動的負荷割り当ての種類を幾つか用意したが、いずれか1つのモードだけを実施するものであっても構わない。

30

・ メインノード(またはサブノード)の数に変更があったときには、負荷割り当てをし直したが、対象とする画像処理が開放されるまで処理開始時の割り当てを維持する構成であっても構わない。

【0099】

・ ストレージサーバとノードの機能を両方持つ計算機であってもよい。前記実施形態ではストレージサーバ2と画像処理システム(例えばA)5は別々の計算機に分かれていたが、これが同一の計算機であっても構わない。つまり、ストレージサーバ2となる計算機は固定されているが、ユーザーがストレージサーバ2上でボリュームレンダリング処理を行いその画像をストレージサーバ2び画面で見ることができる。

40

【0100】

・ サーバにCPUやメモリの空きがあるときにサーバにボリュームレンダリングの計算処理を負担させても構わない。この場合、ストレージサーバ2はボリュームレンダリング画像を表示する表示装置4を備えず、単に分散処理目的で分散ボリュームレンダリング処理のみを実行する構成でも構わない。

【0101】

・ ネットワークを構成する複数のノードのうち少なくとも一つに入力手段及び表示手段が備えられるだけの構成でも構わない。例えば複数のノードのうち幾つかが画像処理計算用ではなく他の用途で使われるもので、他の用途のノード(計算機)のCPU使用率やメモリ空き容量から空きがあれば分散ボリュームレンダリング処理に利用する構成でも構

50

わない。

【0102】

・ 1つのノードが、要求手段、ノード割当手段、負荷割当手段、演算手段、連結手段及び表示制御手段のすべての機能を実現するプログラムを持つ必要はない。つまりノードのすべてがメインノードとなる機能を備えるのではなく、サブノードとなる機能のプログラムだけを持つノードが含まれていてもよい。サブノードは例えば演算手段の機能（ボリュームレンダリング演算機能）のプログラムを持つ。またサーバ機能を持つノードについては、要求手段の機能をプログラムから削除することができる。

【0103】

また、ボリュームレンダリング処理システムを構成する複数台のノードの中に、比較的負荷のかかるボリュームレンダリング演算はせず、画像表示などの比較的演算負荷の小さい処理だけを主に担当するノードが存在してもよい。すなわち、このノードは、入力手段、要求手段、ノード割当手段、負荷割当手段、連結手段および表示制御手段は備えるが、比較的負荷のかかる演算手段の機能は他のノード（割り当てノード）に任せる。具体的には、演算能力の低い計算機に端末機能を持たせ、ボリュームレンダリング画像の演算はネットワークを通じて他のノードに任せる形とする。例えば4次元以上の高次元ボリュームレンダリング処理をするにはノードに高い演算能力が要求されるが、ボリュームレンダリング演算を行わないノードであれば演算能力の低いパーソナルコンピュータ（ノートパソコン等）で済ませられ、比較的高価なワークステーションばかりを取り揃える必要がなくなる。なお、この例のノードは、図2における要求処理部10、割当処理部11、データ連結処理部13、画像メモリ15および表示制御部16を備え、分散ボリュームレンダリング処理部12およびボクセルデータ格納部14は備えていない。また、要求処理部10、割当処理部11、データ連結処理部13および表示制御部16は、コンピュータにより実行されるプログラムにより実現される。

【0104】

さらにノード割当手段、負荷割当手段、連結手段も他のノードに任せ、他のノードで連結処理まで終わったボリュームレンダリング画像のデータを受信し、ボリュームレンダリング画像を表示手段に表示させる表示制御手段の機能だけを持つノード（端末ノード）が存在してもよい。つまり、この場合のノードは、入力手段、要求手段、表示制御手段を備えるが、ノード割当手段、負荷割当手段、演算手段、連結手段は他のノードが受け持つ。例えば連結手段はメインノードが受け持ち、このノードは、メインノードから受信したボリュームレンダリング画像データに基づき画像表示処理を行う。見かけ上は、この端末ノードとは別の、演算処理を受け持つ複数台のノードのうち一台がメインノードとなる。

【0105】

具体的には、この端末ノードは回転、縮小、オパシティ値の変更、色の変更などの表示指示をメインノードに送信する。このメインノードがノード割当ておよび負荷分割からデータ連結処理までを行い、得られた画像を端末ノードに送信する。端末ノードは受信した画像を表示する。この例の端末ノードは、図2における要求処理部10と、画像メモリ15と、表示制御部16とを備え、割当処理部11、分散ボリュームレンダリング処理部12およびデータ連結処理部13、ボクセルデータ格納部14は備えていない。端末ノードでは、メインノードから受信したボリュームレンダリング画像データが画像メモリ15に格納され、表示制御部16はこの画像データを基に表示装置4の画面にボリュームレンダリング画像を表示させる。なお、要求処理部10および表示制御部16は、コンピュータにより実行されるプログラムにより実現される。

【0106】

・ サーバに対するボクセルデータの要求元はメインノードのみに限定されない。例えばサブノードが割り当てられたボクセルデータ（または分割ボクセルデータ）をサーバに対して個々に要求する方法を採用しても構わない。この場合、メインノードの要求処理部10とサブノードの要求処理部10とにより要求手段が構成される。

【0107】

・ ノード割当手段がノード割り当て時に各ノードの計算資源使用状況の情報を基に判断する「分散処理可能な余力」、「ノードの処理能力」とは、ノードのCPU余力およびメモリ空き容量のみを指標とする方法に限定されない（計算余力もCPU余力に限定されない）。要するに、計算能力の指標とデータ容量の指標の2種類の指標からノード割り当てを判断することを基本とし、各指標としてCPU余力とメモリ空き容量のみを用いることに限定されない。計算能力の指標はCPU以外にも計算処理する部分がある場合、例えばボリュームレンダリング専用拡張ボード上でも計算を行う場合はこれも考慮する。またデータ容量の指標は、通常のメモリのほかに専用拡張ボード上のメモリがある場合は、専用拡張ボード上のメモリのメモリ空き容量も考慮する。また同一ノード上で異なるメモリ構成を取る場合、メモリの種類の違いを考慮して指標とする。また、同一のボリュームレンダリング専用拡張ボードに専用演算チップと、専用メモリの両方がある場合、それらも指標として考慮する。さらに計算能力の指標として、例えばシステムバスの性能（帯域幅、クロック）を考慮する。システムバスは、CPUとその他のコンポーネント（メモリやビデオカードなど）を結ぶ重要なパイプで、大量の命令やデータをCPUに送ったり、CPUが処理した結果をメモリやビデオカードなどに送り返す役割を果たし、計算能力に影響を与えるからである。

10

【0108】

計算能力の指標の算出方法としては以下の例が挙げられる。各ノードは、能力値ベクトルを持っており、能力値ベクトルの要素として考えられるものを以下に列挙する。ベクトルの個々の要素はそれぞれ相対的な値を持つ。なお、以下の(3)、(7)、(12)、(15)はノード割り当て時にその都度計測の必要がある。

20

- (1) CPUクロック
- (2) CPUキャッシュ容量
- (3) CPU未使用率（= 1 - CPU使用率）
- (4) システムバス帯域幅
- (5) システムバスクロック
- (6) 専用拡張ボード演算処理部クロック
- (7) 専用拡張ボード演算処理部未使用率（= 1 - ボード使用率）
- (8) 拡張ボードとシステム間のバス帯域幅
- (9) 拡張ボードとシステム間のバスクロック
- (10) メモリ総容量
- (11) メモリの動作クロック
- (12) メモリ空き容量
- (13) 専用拡張ボードの総メモリ容量
- (14) 専用拡張ボードのメモリの動作クロック
- (15) 専用拡張ボードのメモリ空き容量

30

ここで、能力値ベクトルを入力とし、スカラー値を出力する関数を定義すれば、計算能力の指標を得ることができる。例えばスカラー値を出力する関数の例を以下に示す。なお、以下の関数式において（ ）内に示す各要素の数値は、機種間の相違を考慮した相対値で表している。また（ ）の前の乗算値は各要素の重み付け値である。

40

【0109】

(A) 計算能力の指標（計算余力）= $[35 \cdot (40: \text{CPUクロック}) + 10 \cdot (10: \text{CPUキャッシュ容量})] \cdot (0.70: \text{CPU未使用率}) + 15 \cdot (20: \text{システムバス帯域幅}) + 10 \cdot (30: \text{システムバスクロック}) + [17.5 \cdot (50: \text{専用拡張ボード演算処理部クロック})] \cdot (0.65: \text{専用拡張ボード演算処理部未使用率}) + 5 \cdot (70: \text{拡張ボードとシステム間のバス帯域幅}) + 7.5 \cdot (60: \text{拡張ボードとシステム間のバスクロック})$

(B) データ容量の指標（保持可能データ余力）= $5 \cdot (70: \text{システムバス帯域幅}) + 5 \cdot (40: \text{システムバスクロック}) + [70 \cdot (10: \text{メモリ総容量}) + 5 \cdot (3: \text{メモリの動作クロック})] \cdot (0.55: \text{メモリ空き容量}) + [5 \cdot (30: \text{専用拡張ボードのメモリの動作クロック}) + 10 \cdot (20: \text{専用拡張ボードの総メモリ容量})] \cdot (0.35: \text{専用拡張ボードのメモリ空き容量})$

50

なお、上記(1)～(15)の要素は全て採用する必要はもちろんなく、必要に応じて適宜選択することができるが、CPU余力(例えばCPU未使用率)とメモリ空き容量は必須とすることが好ましい。

【0110】

・ ノード割り当てをする際に、通信オーバーヘッドを考慮してもよい。例えば通信オーバーヘッドは、データ容量が小さい場合は複数ノードで分散処理するよりも、一台のノードで行う方が効率的であることを判断する条件の1つになる。データ分割モードではノード割り当てをせざるを得ないが、データコピーモードおよび混合モードでは、通信オーバーヘッドが大きいと判断されれば同一データ(同一分割データを含む)を複数ノードに割り当てることはせず一台のノードのみ割り当てるか、ノード割り当て数を通常より減らすようにする。このように構成すれば、通信オーバーヘッドに起因する画像処理速度の低下を防ぐことができる。

10

【0111】

・ メインノード数が変化したときに動的負荷割り当て内容を見直す制御方法を採用したが、動的負荷割り当て内容の見直し時期は特にメインノード数変化時に限らない。例えば所定時間(例えば数100ミリ秒～数秒)ごとに動的負荷割り当て内容の見直しを行ってもよい。

【0112】

・ 各割り当てノードへのボクセルデータ(分割ボクセルデータを含む)のサーバへの転送要求は、メインノードがサーバに対して一括的に行うことに限定されない。例えばメインノードがサブノードに対して割当負荷の情報を伝え、各ノードがそれぞれサーバに対し自分が使うボクセルデータ(分割ボクセルデータを含む)を個々に要求するシステム構成を採用することもできる。さらにメインノードから1つのサブノードに対し各割り当てノード分すべての割当負荷の情報を伝え、そのサブノードがサーバに対し各割り当てノード分すべてのボクセルデータ(分割ボクセルデータを含む)を一括要求するシステム構成を採用することもできる。つまりノードは、ボクセルデータをサーバに対して一括要求する要求手段を必ずしも持つ必要はなく、他ノードが一括要求したボクセルデータ(分割ボクセルデータを含む)を受け付けるだけの構成であっても構わない。

20

【0113】

・ 4次元(4D)以上のボリュームレンダリング処理においては、複数のノードに動的に負荷を割り当てる際の負荷分割方法は前記実施形態の方法に限定されない。次元に応じた負荷分割方法を採用できる。4D処理の具体例を以下に示す。

30

4D処理は3D処理を時系列方向に繰り返す。データコピーモード、分割モード、混合モードのいずれも採用できる。但し、データの分割は、3D処理の場合は空間的な分割であったが、4D処理の場合は空間的な分割とともに時系列方向の分割も含まれる。例えば、「300スライス×10フェーズ(時刻)=計3000スライス」のような4Dデータ、計算ノード2台(同等の能力)があったとする。計算ノード1がメインノードとする。

(1)時間コピーモード

計算ノード1,2に全データをコピーする。メインノード(ノード1)は時刻1を計算しながら、ノード2に時刻2のデータの分散処理要求を出す。ノード1の演算結果に基づき時刻1の画像を表示手段に表示した後、ノード2の演算結果に基づき時刻2の画像を表示する。

40

(2)時間分割モード

計算ノード1に「時刻1,3,5,7,9」、計算ノード2に「時刻2,4,6,8,10」を分担させる。各計算ノードに全てのデータをコピーすることがベストだが、ストレージ容量の都合により、時刻別に半分の容量ずつ重複しないように割り当てるとする。各ノードに割り当てる時刻は割り当て手段が決定する(この例では交互の時刻)。

(3)時間混合モード

(1),(2)の組み合わせ

(4)時間&空間分割ノード

時間分割と空間分割(空間分割モード、空間混合モード)を組み合わせる負荷分担方法も

50

採用できる。現実的には4Dデータの分割は、同時刻では空間的に分割せず、異なる時刻をそれぞれ別のノードに振り分ける時間的な分割のみを行う場合が多いと考えられるが、時間的な分割に加え同時刻のデータを重複しないように別々のノードに空間的にも分割する方法も採用できる。

【0114】

従って、これらの方法の採用により、3D処理よりも扱うデータ容量が大きい4D処理において、時系列方向に分割して協調分散処理を行うことで、空間的な分割のみ採用する場合に比べ、システムの処理能力をより効果的に向上させることができる。よって、4Dデータのボリュームレンダリング処理の需要が増えつつある医療業界で、従来技術よりも著しい性能差が出る。

10

【0115】

前記実施形態及び別例から把握できる技術的思想を、以下に記載する。

(1) 請求項1~4のいずれか一項の発明において、前記サーバは前記複数のノードとは別に設けられ、三次元以上のボクセルデータを記憶するサーバと、前記サーバからボクセルデータを受け付けてボリュームレンダリング処理を行う複数のノードとによってコンピュータネットワークが構築されたボリュームレンダリング処理システムであることを特徴とする。この構成であれば、サーバがデータを保管することで、ノードの計算資源をデータの計算処理のためにより多く活用でき、一層の高速処理に対応できる。

【0116】

(2) 請求項1~4のいずれか一項の発明において、前記サーバは前記複数のノードのうち少なくとも一つが兼ねていることを特徴とする。この構成であれば、サーバ機能を持つノードからも表示対象指定による表示要求を出してその表示手段に協調分散処理結果の画像(ボリュームレンダリング画像)を表示させることができる。

20

【0117】

(3) 請求項1~4のいずれか一項に記載の発明において、前記複数のノードのうち少なくとも二つは、前記入力手段、前記要求手段、前記ノード割当手段、前記負荷割当手段、前記演算手段、前記連結手段及び表示制御手段をそれぞれ備え、表示手段に画像を表示可能な表示対象指定元のメインノードとなり得るノードであることを特徴とする。この構成によれば、複数のノードのうち各手段のすべてを備える少なくとも二つのノードは、ボリュームレンダリング画像を見ることが可能なメインノードにも、分散処理(分散ボリュームレンダリング処理)でメインノードを支援するサブノードにもなり得る。よって、ユーザーは少なくとも二つ(二つ以上)のどのノードからも、入力手段で指定したボクセルデータのボリュームレンダリング画像を表示手段を通して見ることができる。

30

【0118】

(4) 前記技術的思想(3)において、前記複数のノードのすべてが前記メインノードとなり得るノードである。この構成によれば、各ノードはサブノード、メインノードに必要な機能を全て持っており、役割に応じてサブノード、メインノードと切り替わるので、どのノードもメインノードに成り得る。よって、どのノードからもボリュームレンダリング画像を見ることができる。

【0119】

(5) 請求項1~4及び技術的思想(1)~(4)のいずれかにおいて、前記ノード割当手段は、ノード割り当て条件として前記計算資源使用状況以外に通信オーバーヘッドを考慮し、計算資源使用状況から決まるノード数では通信オーバーヘッドが大きいと判断したときには、前記表示対象指定元となるメインノードを含む割り当てノード数を、前記計算資源使用状況から決まるノード数より少なく設定し、前記負荷割当手段は、前記割り当てノードに対し当該割り当てノード数に応じた負荷割り当てを行うことを特徴とする。

40

【0120】

(6) 請求項1~4及び技術的思想(1)~(5)のいずれかにおいて、前記ノード割当手段は、前記表示対象指定元となるメインノードの処理能力で足りると判断すれば他のノードに割り当てをせず当該メインノード単独で割り当て、前記負荷割当手段は、当該メ

50

インノードにのみ前記表示対象のボクセルデータとボリュームレンダリング処理を割り当て、当該メインノードの前記演算手段が前記表示対象のボクセルデータに対するボリュームレンダリング処理を単独で実行することを特徴とする。この構成によれば、メインノードの処理能力で足りると判断すればメインノード単独でボクセルデータに対するボリュームレンダリング処理が行われる。そして、ノード割当手段は、メインノードの処理能力では足りないと判断したときに他のノードをも割り当て、各割り当てノードにより協調分散処理が行われる。

【0121】

(7) 請求項1～4及び技術的思想(1)～(6)のいずれかにおいて、前記表示対象指定元となるメインノードから入力手段により指定されて前記サーバに要求されたボクセルデータのうち少なくとも割り当てべきデータを既にローカルストレージに持つ割り当てノードに対しては、前記サーバはその割り当てデータを送信することはせず、当該割り当てノードにおける前記演算手段は前記ローカルストレージに持つデータを用いて当該データに対するボリュームレンダリング処理を実行することを特徴とする。この構成によれば、サーバから割り当てノードへのデータ送信時間を省け一層の高速処理に対応できる。

10

【0122】

(8) 請求項1～4及び技術的思想(1)～(7)のいずれかにおいて、前記負荷割当手段は、前記ノード割当手段により割り当てられた各ノードに対し各々の処理能力に応じて前記表示対象のボクセルデータと該ボクセルデータに施すボリュームレンダリング処理とのうち少なくとも一方の分割割合を決定して動的負荷割り当てを行うことを特徴とする。

20

【0123】

(9) 請求項1～4及び技術的思想(1)～(8)のいずれかにおいて、前記負荷割当手段は、前記ノード割当手段により割り当てられた各ノードの処理能力に応じて、前記表示対象のボクセルデータに施すべき計算処理のみの分割割合を決定するデータコピーモードと、前記表示対象のボクセルデータのみの分割割合を決定するデータ分割モードと、前記表示対象のボクセルデータと該ボクセルデータに施すべき計算処理との両方の分割割合を決定する混合モードとの中から一つを選択することを特徴とする。この構成によれば、ノードの計算資源を一層効率よく活用できボリュームレンダリング処理を高速に実行できる。

30

【0124】

(10) 請求項1～4及び技術的思想(1)～(9)のいずれかにおいて、前記表示対象指定元のメインノードは、他のノードとの通信によって各ノードの計算資源利用状況の情報を取得しており、前記ノード割当手段は前記各ノードの計算資源利用状況の情報に基づいて当該各ノードの処理能力を判断して処理能力上必要な空きのあるノードを選択してノード割り当てを行い、前記負荷割当手段は、前記各割り当てノードの計算資源利用状況の情報に基づいて当該各割り当てノードの処理能力を判断して処理能力上の空きに応じてボクセルデータとボリュームレンダリング処理との少なくとも一方の分割割合を決定して動的負荷割り当てを行うことを特徴とする。

【0125】

(11) 請求項1～4及び技術的思想(8)～(10)のいずれかにおいて、前記要求手段は、前記ノード割当手段によるノード割当情報と、前記負荷割当手段による負荷割当情報とを前記サーバに要求として送り、前記サーバは前記ノード割当情報で指定された割り当てノードに対し、前記負荷割当情報で指定された割合でデータ分割された分割ボクセルデータ又はボクセルデータをそのまま送ることを特徴とする。

40

【0126】

(12) 前記技術的思想(11)において、前記各割り当てノードの演算手段は、前記サーバから取得した分割ボクセルデータ又はボクセルデータに対し前記表示対象指定元となるメインノードから得た負荷割当情報で指定された計算負荷の計算処理を実行することを特徴とする。

50

【 0 1 2 7 】

(1 3) 請求項 1 ~ 4 及び技術的思想 (1) ~ (1 2) のいずれかにおいて、前記負荷割当手段は、予め通信で情報として得た各ノードの処理能力に応じて、前記各割り当てノードのメモリ空き容量に応じてデータ容量負荷を決定するとともに CPU 空き容量に応じて計算負荷を決定して動的負荷割り当てを行うことを特徴とする。この場合、複数のノード (計算機) が動的にデータ容量負荷と計算負荷を分担しながら協調分散処理を行うことにより、請求項 1 の発明と同様の効果が得られる。

【 0 1 2 8 】

(1 4) 請求項 1 ~ 4 及び技術的思想 (1) ~ (1 3) のいずれかにおいて、前記表示対象指定元となるメインノードの数に変更があると、前記ノード割当手段はノードの割り当てをし直すとともに、前記負荷割当手段は割り当てノードに対する動的負荷割り当てをし直すことを特徴とする。この構成によればメインノードの数に変更があったときにノード割り当てと動的負荷割り当ての見直しが図られるので、他ノードの処理を妨げず計算資源を有効に活用でき、ボクセルデータに施すボリュームレンダリング処理を高速に実行できる。

10

【 0 1 2 9 】

(1 5) 請求項 1 ~ 3 のいずれか一項の発明において、前記ノード割当手段は、計算能力の指標とデータ容量の指標に基づいてノード割当を行う。

(1 6) 前記技術的思想 (5) ~ (1 5) のいずれかにおける前記ノード割当手段及び前記負荷割当手段を備える請求項 6 に記載の計算機。

20

【 0 1 3 0 】

(1 7) 請求項 1 ~ 7 及び技術的思想 (1 1) , (1 2) のいずれかに記載の分散処理システムにおいて使用されるサーバであって、前記表示対象指定元となるメインノードからノード割当情報及びデータ分割割当情報と共にボクセルデータの要求を受け付けると、前記割り当てノードに対しデータ分割割当情報から決まる分割ボクセルデータを送信することを特徴とするサーバ。

【 0 1 3 1 】

(1 8) 請求項 1 0 に記載のボリュームレンダリング処理方法において、コンピュータの前記ノード割当手段は、前記ノードを割り当てる段階において、前記表示対象指定元となるメインノードの処理能力で足りると判断すれば他のノードに割り当てをせず当該メインノード単独で割り当て、コンピュータの負荷割当手段は前記動的負荷割り当てを行う段階において当該メインノードにのみボクセルデータとボリュームレンダリング処理を割り当て、当該メインノードを構成するコンピュータの前記演算手段は単独で前記表示対象のボクセルデータに対するボリュームレンダリング処理を実行することを特徴とする。

30

【 0 1 3 2 】

(1 9) 請求項 1 0 に記載のボリュームレンダリング処理方法において、前記表示対象指定元となるメインノードから入力手段により指定されて前記サーバに要求されたボクセルデータのうち少なくとも割り当てるべきデータを既にローカルストレージに持つ割り当てノードに対しては、前記サーバはその割り当てデータを送信することはせず、当該割り当てノードにおけるコンピュータの前記演算手段は前記ローカルストレージに持つデータを用いて当該データに対するボリュームレンダリング処理を実行することを特徴とする。この方法によれば、割り当てノードが既にローカルストレージに必要なボクセルデータを持つ場合は、サーバから割り当てノードへのデータ送信時間を省け一層の高速処理に対応できる。

40

【 図面の簡単な説明 】

【 0 1 3 3 】

【 図 1 】 一実施形態におけるボリュームレンダリング処理システムのブロック図。

【 図 2 】 ボリュームレンダリング処理システムの機能ブロック図。

【 図 3 】 計算負荷割り当て手法の説明図。

【 図 4 】 協調分散処理システムの特徴的な作動状況を説明する説明図。

50

- 【図5】 協調分散処理システムの特徴的な作動状況を説明する説明図。
- 【図6】 ノード割り当ておよび負荷割り当ての変更処理の説明図。
- 【図7】 ノード割り当ておよび負荷割り当ての変更処理の説明図。
- 【図8】 (a)「平行投影法」、(b)「透視投影法」を示す説明図。
- 【図9】 ボリュームレンダリング処理を説明するブロック図。
- 【図10】 (a)データ分割、(b)は計算負荷分割の説明図。
- 【図11】 計算分割時のボリュームレンダリング処理を説明するブロック図。
- 【図12】 ボクセルデータ分割の説明図。
- 【図13】 (a)～(c)データコピーモードにおける計算負荷割当の説明図。
- 【図14】 データ分割モードにおけるデータ分割の説明図。
- 【図15】 混合モードにおけるデータ分割および計算分割の説明図。
- 【図16】 従来技術における画像処理システムのブロック図。
- 【図17】 従来技術における画像処理用ネットワークシステムのブロック図。
- 【図18】 同じくブロック図。

10

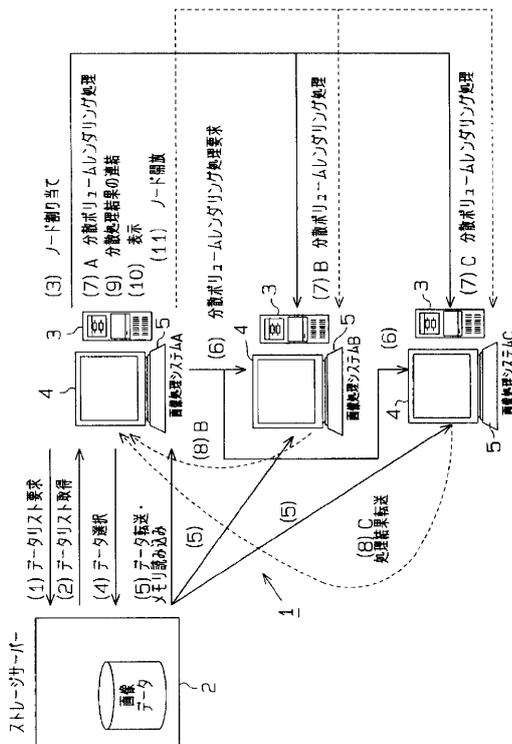
20

【符号の説明】

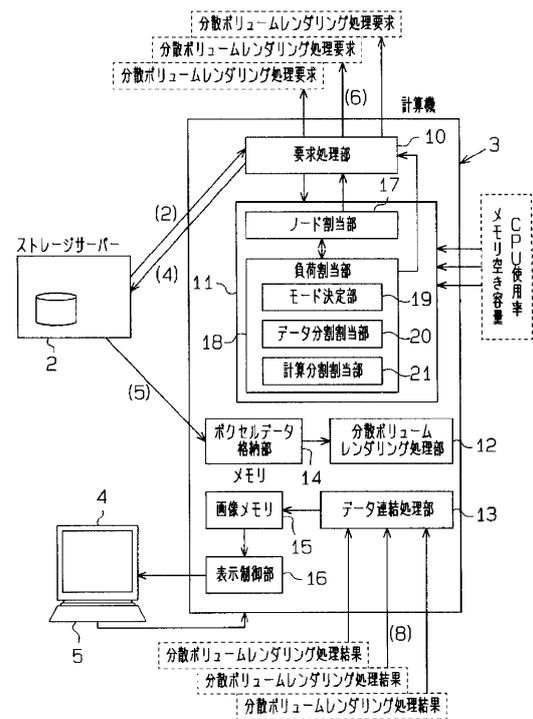
【0134】

1 ... ボリュームレンダリング処理システムとしての協調分散処理システム、2 ... サーバとしてのストレージサーバ、3 ... ノード（計算機）、4 ... 表示手段としての表示装置、5 ... 入力手段としての入力装置、10 ... 要求手段としての要求処理部、12 ... 演算手段としての分散ボリュームレンダリング処理部、13 ... 連結手段としてのデータ連結処理部、16 ... 表示制御手段としての表示制御部、17 ... ノード割当手段としてのノード割当部、18 ... 負荷割当手段としての負荷割当部、20 ... データ分割割当部、21 ... 計算分割割当部、VD ... ボクセルデータ、VD1, VD2 ... 分割ボクセルデータ。

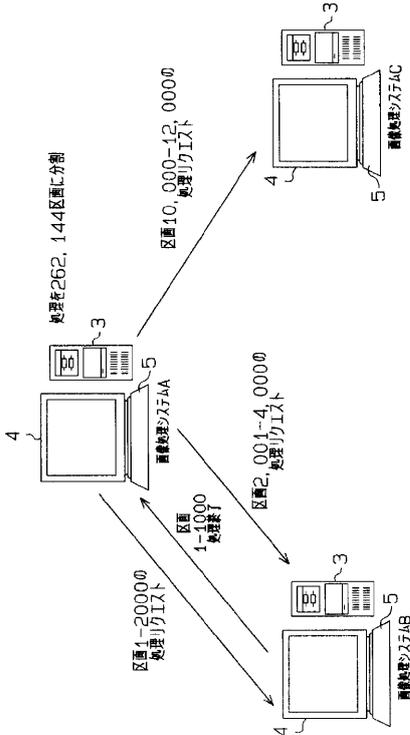
【図1】



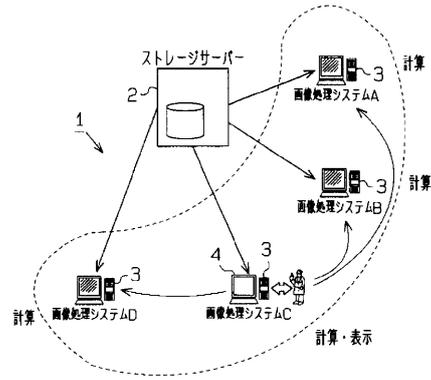
【図2】



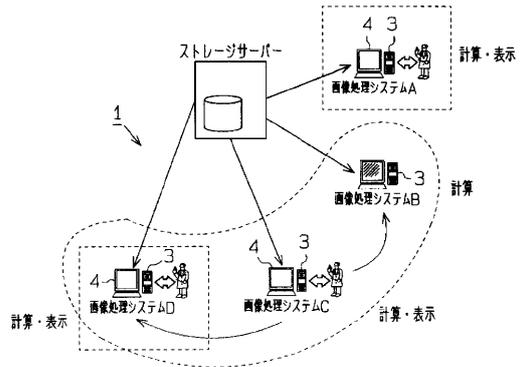
【 図 3 】



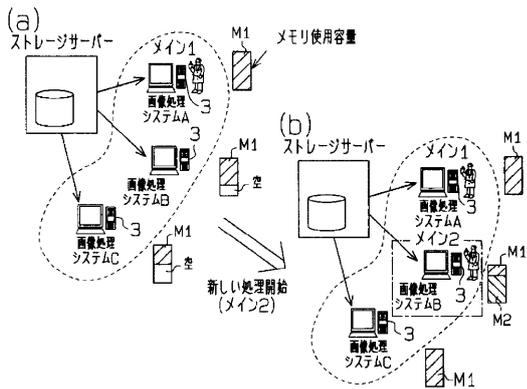
【 図 4 】



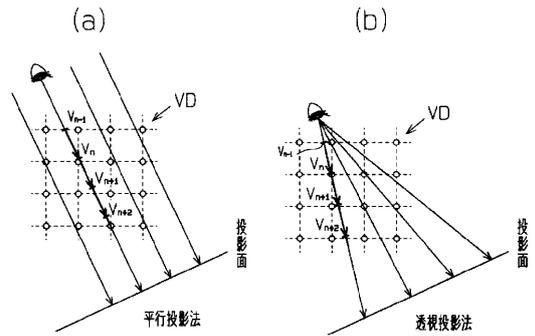
【 図 5 】



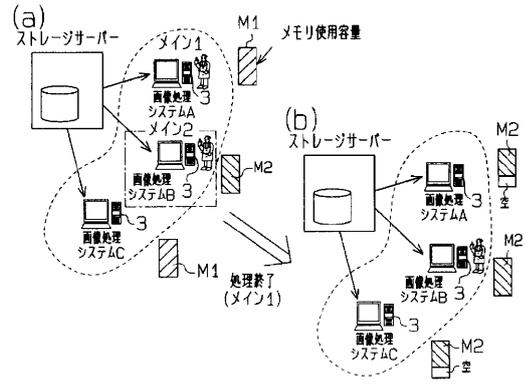
【 図 6 】



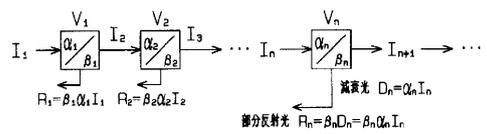
【 図 8 】



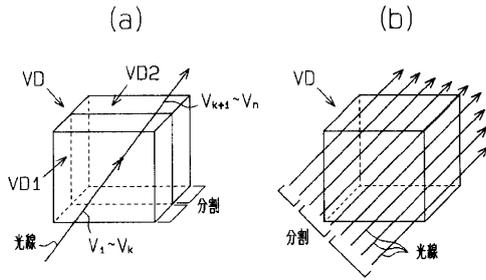
【 図 7 】



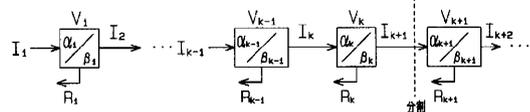
【 図 9 】



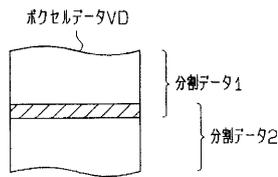
【図10】



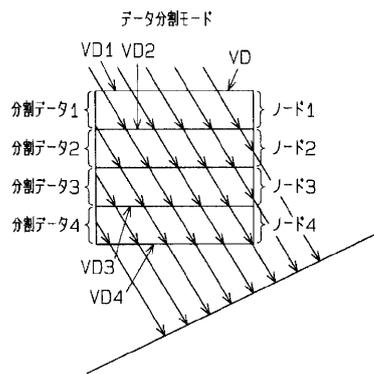
【図11】



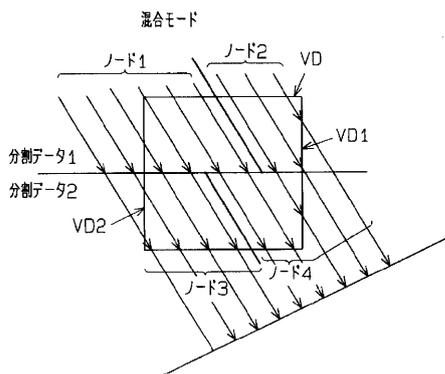
【図12】



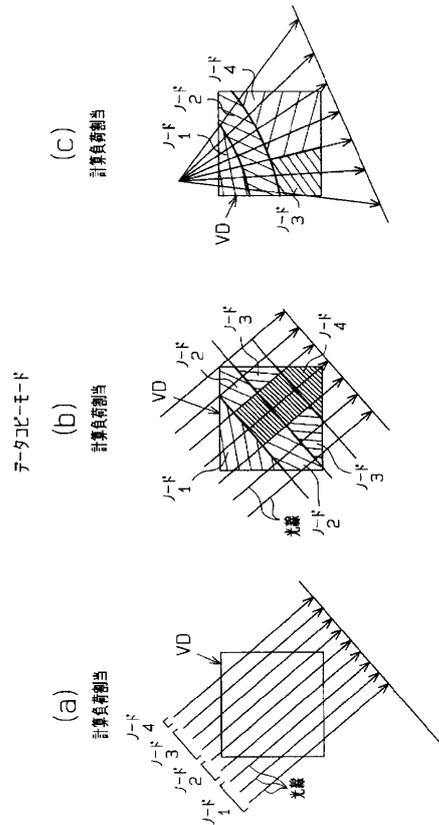
【図14】



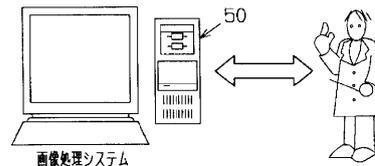
【図15】



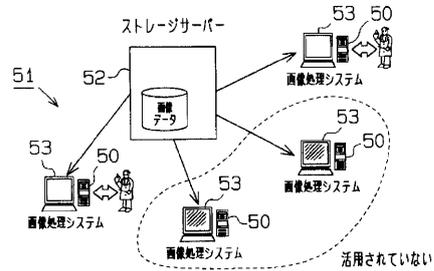
【図13】



【図16】



【図17】



【図18】

