

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5368254号
(P5368254)

(45) 発行日 平成25年12月18日 (2013.12.18)

(24) 登録日 平成25年9月20日 (2013.9.20)

(51) Int.Cl.	F I	
HO4N 5/91 (2006.01)	HO4N 5/91	Z
GO6T 3/40 (2006.01)	GO6T 3/40	A
GO6T 11/80 (2006.01)	GO6T 11/80	A
HO4N 1/387 (2006.01)	HO4N 1/387	1 O 1
GO6F 12/00 (2006.01)	GO6F 12/00	5 2 O A
請求項の数 27 (全 41 頁) 最終頁に続く		

(21) 出願番号 特願2009-245797 (P2009-245797)
 (22) 出願日 平成21年10月26日 (2009.10.26)
 (65) 公開番号 特開2011-91769 (P2011-91769A)
 (43) 公開日 平成23年5月6日 (2011.5.6)
 審査請求日 平成24年7月4日 (2012.7.4)

(73) 特許権者 310021766
 株式会社ソニー・コンピュータエンタテインメント
 東京都港区港南1丁目7番1号
 (74) 代理人 100105924
 弁理士 森下 賢樹
 (74) 代理人 100109047
 弁理士 村田 雄祐
 (74) 代理人 100109081
 弁理士 三木 友由
 (74) 代理人 100134256
 弁理士 青木 武司

最終頁に続く

(54) 【発明の名称】 画像ファイル生成装置、画像処理装置、画像ファイル生成方法、画像処理方法、および画像ファイルのデータ構造

(57) 【特許請求の範囲】

【請求項1】

入力された画像データから、異なる解像度の複数の画像データを生成し、解像度順に階層化して構成される階層データを生成する画像階層生成部と、

前記階層データを構成する各階層の画像データを所定サイズのタイル画像に分割する画像分割部と、

前記階層データを構成する各階層の画像をタイル画像に分割した際の各領域であるタイル領域と、当該領域を描画する際に用いる前記タイル画像のデータとを対応づけたインデックスデータを作成するインデックス作成部と、

前記タイル画像のデータと前記インデックスデータとを含む画像ファイルを生成する画像ファイル生成部と、を備え、

前記インデックス作成部は、前記階層データの画像平面とそれに垂直な解像度の軸によって定義される仮想空間において前記階層データで構成される階層構造体を分割してなる領域ごとに、前記タイル領域ごとのインデックスデータをまとめたインデックスブロックの形式で、前記インデックスデータを生成することを特徴とする画像ファイル生成装置。

【請求項2】

前記インデックス作成部は、前記階層構造体を分割してなる領域と、それに対応する前記インデックスブロックとを対応づけたヘッダデータをさらに生成し、

前記画像ファイル生成部は、前記ヘッダデータおよび前記インデックスブロックを前記インデックスデータに含めることを特徴とする請求項1に記載の画像ファイル生成装置

10

20

【請求項 3】

ある解像度の階層の画像を解析することによって、当該解像度の画像に含まれるタイル画像のデータを流用して描画できる、当該解像度より高解像度の階層の画像の領域を特定する冗長性検出部をさらに備え、

前記インデックス作成部は、前記インデックスデータにおいて、前記冗長性検出部が特定した領域以外の領域にのみタイル画像のデータを対応づけることを特徴とする請求項 1 または 2 に記載の画像ファイル生成装置。

【請求項 4】

ある解像度の階層の画像を解析することによって、当該解像度の画像に含まれるタイル画像のデータを流用して描画できる、当該解像度より高解像度の階層の画像の領域を特定する冗長性検出部をさらに備え、

前記冗長性検出部が特定した領域が、前記階層構造体を分割してなる一の領域の全体に渡るとき、前記画像ファイル生成部は、当該領域に対応する前記インデックスブロックを前記画像ファイルから除外することを特徴とする請求項 1 または 2 に記載の画像ファイル生成装置。

【請求項 5】

前記インデックスデータは、一のタイル画像のデータを複数のタイル領域に対応づけ、前記画像ファイルは、前記インデックスデータにおいて対応づけられていないタイル画像のデータを含まないことを特徴とする請求項 1 から 4 のいずれかに記載の画像ファイル生成装置。

【請求項 6】

前記階層構造体を分割してなる領域はそれぞれ、前記階層データに含まれる複数の階層の画像の対応する領域を含むことを特徴とする請求項 1 から 5 のいずれかに記載の画像ファイル生成装置。

【請求項 7】

一つのインデックスブロックにおいてタイル画像のデータに対応づけることのできる前記タイル領域の数は、全てのインデックスブロックで同一であることを特徴とする請求項 1 から 6 のいずれかに記載の画像ファイル生成装置。

【請求項 8】

一の画像を異なる解像度で表す画像データによって構成される階層データのそれぞれの画像を所定サイズに分割してなるタイル画像のデータと、前記階層データを構成する各階層の画像をタイル画像に分割した際の各領域であるタイル領域と当該領域を描画する際に用いる前記タイル画像のデータとを対応づけたインデックスデータと、を含む画像ファイルを保持する記憶装置と、

画像平面上の移動および解像度の変更を含む表示領域変更要求をユーザから受け付ける入力情報取得部と、

前記表示領域変更要求に応じて新たに表示すべき領域を描画するのに必要なタイル画像のデータを、前記インデックスデータを参照することによって特定するタイル画像特定部と、

特定したタイル画像のデータを前記記憶装置より読み出し、新たに表示すべき領域を描画する表示画像処理部と、

を備え、

前記インデックスデータは、前記階層データの画像平面とそれに垂直な解像度の軸によって定義される仮想空間において前記階層データで構成される階層構造体を分割してなる領域ごとに、前記タイル領域ごとのインデックスデータをまとめたインデックスブロックの形式を有することを特徴とする画像処理装置。

【請求項 9】

前記インデックスデータは、前記階層構造体を分割してなる領域と、それに対応する前記インデックスブロックとを対応づけたヘッダデータを含み、

前記タイル画像特定部は、新たに表示すべき領域の情報に基づき前記ヘッダデータを検

10

20

30

40

50

索し、当該表示すべき領域を含む、前記階層構造体を分割してなる領域を特定したうえで、それに対応づけられた前記インデックスブロックを参照することにより、前記必要なタイル画像のデータを特定することを特徴とする請求項 8 に記載の画像処理装置。

【請求項 10】

前記インデックスデータは、ある解像度の画像に含まれるタイル画像のデータを流用して描画できる、当該解像度より高解像度の階層の画像のタイル領域に対しては無効データを対応づけ、

前記タイル画像特定部は、前記インデックスデータにおいて、新たに表示すべき領域に含まれる前記タイル領域に対して無効データが対応づけられていたら、前記階層データを縮小方向に辿り、前記領域の縮小領域に含まれ特定のタイル画像のデータに対応づけられているタイル領域を探索することにより、当該タイル画像のデータを特定し、

前記表示画像処理部は、前記タイル画像特定部が特定したタイル画像を拡大することにより新たに表示すべき領域を描画することを特徴とする請求項 8 または 9 に記載の画像処理装置。

【請求項 11】

前記ヘッダデータは、ある解像度の画像に含まれるタイル画像のデータを流用して描画できる、当該解像度より高解像度の階層の画像のタイル領域が前記階層構造体を分割してなる一の領域の全体に渡るとき、当該領域に対しては無効データを対応づけ、

前記タイル画像特定部は、前記ヘッダデータにおいて、新たに表示すべき領域を含む、前記階層構造体を分割してなる領域に対して無効データが対応づけられていたら、前記構造体を縮小方向に辿り、新たに表示すべき領域の縮小領域を含み特定のインデックスブロックに対応づけられている領域を探索することにより、当該インデックスブロックを特定したうえで、前記縮小領域のタイル画像を特定し、

前記表示画像処理部は、前記タイル画像特定部が特定したタイル画像を拡大することにより新たに表示すべき領域を描画することを特徴とする請求項 9 に記載の画像処理装置。

【請求項 12】

前記インデックスデータは、前記タイル領域と、前記画像ファイルと異なるファイルに属するタイル画像のデータとを、当該ファイルの識別情報を指定したうえで対応づけ、

前記表示画像処理部は、新たに表示すべき領域に用いるタイル画像のデータを、前記インデックスデータにおいて指定された別のファイルから読み出し描画することを特徴とする請求項 8 から 11 のいずれかに記載の画像処理装置。

【請求項 13】

一の画像を異なる解像度で表す画像データによって構成される階層データのそれぞれの画像を所定サイズに分割してなるタイル画像のデータと、前記階層データを構成する各階層の画像をタイル画像に分割した際の各領域であるタイル領域と当該領域を描画する際に用いる前記タイル画像のデータとを対応づけたインデックスデータと、を含む画像ファイルを読み出す画像ファイル取得部と、

前記画像ファイルを用いて表示される画像の更新情報を取得する更新情報取得部と、

前記更新情報に基づき、更新対象の領域を描画する際に用いる新たなタイル画像のデータを生成するタイル画像生成部と、

前記インデックスデータにおいて更新対象の領域に含まれるタイル領域に対応づけられるタイル画像のデータを、新たなタイル画像のデータに更新するインデックスデータ更新部と、

前記タイル画像生成部が生成したタイル画像のデータと、前記インデックスデータ更新部が更新した前記インデックスデータを含む新たな画像ファイルを生成する画像ファイル生成部と、を備え、

前記インデックスデータは、前記階層データの画像平面とそれに垂直な解像度の軸によって定義される仮想空間において前記階層データで構成される階層構造体を分割してなる領域ごとに、前記タイル領域ごとのインデックスデータをまとめたインデックスブロックの形式を有することを特徴とする画像処理装置。

10

20

30

40

50

【請求項 14】

前記タイル画像生成部は、前記階層データを構成する全ての階層に対し、更新対象の領域を描画する際に用いる新たなタイル画像のデータを生成し、

前記インデックスデータ更新部は、前記全ての階層の更新対象の領域に含まれるタイル領域に対応づけられるタイル画像のデータを更新することを特徴とする請求項 13 に記載の画像処理装置。

【請求項 15】

前記タイル画像生成部は、前記階層データのうち、前記更新情報において指定される特定の階層のみに対し、更新対象の領域を描画する際に用いる新たなタイル画像のデータを生成し、

前記インデックスデータ更新部は、当該特定の階層の更新対象の領域に含まれるタイル領域に対応づけられるタイル画像のデータを更新することを特徴とする請求項 13 に記載の画像処理装置。

【請求項 16】

前記更新情報において、更新前の前記階層データにおける最高解像度の階層を超える解像度の画像の書き込みが指定されている場合、

前記インデックスデータ更新部は、前記インデックスデータにおいて、前記階層データに追加すべき、書き込まれる画像の解像度を有する新たな階層のタイル領域を追加で定義し、そのうち更新対象のタイル領域を、前記タイル画像生成部が新たに生成した、書き込まれる画像を含むタイル画像に対応づけることを特徴とする請求項 13 から 15 のいずれかに記載の画像処理装置。

【請求項 17】

前記更新情報において、更新前の画像に新たな画像領域を追加する指定がされている場合、

前記インデックスデータ更新部は、画像領域の追加によって前記階層構造体の変化した部分に対応する領域を新たに生成し、当該領域に対応する前記インデックスブロックをさらに生成することを特徴とする請求項 13 から 15 のいずれかに記載の画像処理装置。

【請求項 18】

前記タイル画像生成部は、追加される画像領域と、更新前の画像が混在するタイル画像の領域を特定し、前記追加される画像領域のタイル画像のデータとともに当該混在するタイル画像のデータを生成することを特徴とする請求項 17 に記載の画像処理装置。

【請求項 19】

画像ファイル生成装置が行う画像ファイル生成方法であって、
画像階層生成部が、入力された画像データから、異なる解像度の複数の画像データを生成し、解像度順に階層化して構成される階層データを生成しメモリに格納するステップと

画像分割部が、前記階層データを構成する各階層の画像データをメモリより読み出し所定サイズのタイル画像に分割しメモリに格納するステップと、

インデックス作成部が、前記階層データを構成する各階層の画像をタイル画像に分割した際の各領域であるタイル領域と当該領域を描画する際に用いる前記タイル画像のデータとを対応づけたインデックスデータを作成しメモリに格納するステップと、

画像ファイル生成部が、前記タイル画像のデータと前記インデックスデータとを前記メモリより読み出し、それらを含む画像ファイルを生成しメモリに格納するステップと、を
含み、

前記インデックスデータを作成しメモリに格納するステップは、前記階層データの画像平面とそれに垂直な解像度の軸によって定義される仮想空間において前記階層データで構成される階層構造体を分割してなる領域ごとに、前記タイル領域ごとのインデックスデータをまとめたインデックスブロックの形式で、前記インデックスデータを生成することを特徴とする画像ファイル生成方法。

【請求項 20】

10

20

30

40

50

画像処理装置が行う画像処理方法であって、

表示画像処理部が、一の画像を異なる解像度で表す画像データによって構成される階層データのそれぞれの画像を所定サイズに分割してなるタイル画像のデータと、前記階層データを構成する各階層の画像をタイル画像に分割した際の各領域であるタイル領域と当該領域を描画する際に用いる前記タイル画像のデータとを対応づけたインデックスデータと、を含む画像ファイルを記憶装置よりメモリに読み出すステップと、

入力情報取得部が、画像平面上の移動および解像度の変更を含む表示領域変更要求をユーザから受け付けるステップと、

タイル画像特定部が、前記表示領域変更要求に応じて新たに表示すべき領域を描画するのに必要なタイル画像のデータを、前記メモリに読み出された前記インデックスデータを参照することによって特定するステップと、

前記表示画像処理部が、特定したタイル画像のデータを前記メモリより読み出し、新たに表示すべき領域を描画し表示装置に表示するステップと、

を含み、

前記インデックスデータは、前記階層データの画像平面とそれに垂直な解像度の軸によって定義される仮想空間において前記階層データで構成される階層構造体を分割してなる領域ごとに、前記タイル領域ごとのインデックスデータをまとめたインデックスブロックの形式を有することを特徴とする画像処理方法。

【請求項 2 1】

画像処理装置が行う画像処理方法であって、

画像ファイル取得部が、一の画像を異なる解像度で表す画像データによって構成される階層データのそれぞれの画像を所定サイズに分割してなるタイル画像のデータと、前記階層データを構成する各階層の画像をタイル画像に分割した際の各領域であるタイル領域と当該領域を描画する際に用いる前記タイル画像のデータとを対応づけたインデックスデータと、を含む画像ファイルを記憶装置よりメモリに読み出すステップと、

更新情報取得部が、前記画像ファイルを用いて表示される画像の更新情報を入力装置から取得するステップと、

タイル画像生成部が、前記更新情報に基づき、更新対象の領域を描画する際に用いる新たなタイル画像のデータを生成しメモリに格納するステップと、

インデックスデータ更新部が、メモリに読み出された前記インデックスデータにおいて更新対象の領域に含まれるタイル領域に対応づけられるタイル画像のデータを、新たなタイル画像のデータに更新するステップと、

画像ファイル生成部が、新たに生成したタイル画像のデータと、更新した前記インデックスデータを前記メモリから読み出し、それらを含む新たな画像ファイルを生成してメモリに格納するステップと、

を含み、

前記インデックスデータは、前記階層データの画像平面とそれに垂直な解像度の軸によって定義される仮想空間において前記階層データで構成される階層構造体を分割してなる領域ごとに、前記タイル領域ごとのインデックスデータをまとめたインデックスブロックの形式を有することを特徴とする画像処理方法。

【請求項 2 2】

入力された画像データから、異なる解像度の複数の画像データを生成し、解像度順に階層化して構成される階層データを生成しメモリに格納する機能と、

前記階層データを構成する各階層の画像データをメモリより読み出し所定サイズのタイル画像に分割する機能と、

前記階層データを構成する各階層の画像をタイル画像に分割した際の各領域であるタイル領域と、当該領域を描画する際に用いる前記タイル画像のデータとを対応づけたインデックスデータを作成しメモリに格納する機能と、

前記タイル画像のデータと前記インデックスデータとを含む画像ファイルを生成しメモリに格納する機能と、

10

20

30

40

50

をコンピュータに実現させるコンピュータプログラムであって、

前記インデックスデータを作成する機能は、前記階層データの画像平面とそれに垂直な解像度の軸によって定義される仮想空間において前記階層データで構成される階層構造体を分割してなる領域ごとに、前記タイル領域ごとのインデックスデータをまとめたインデックスブロックの形式で、前記インデックスデータを生成することを特徴とするコンピュータプログラム。

【請求項 2 3】

一の画像を異なる解像度で表す画像データによって構成される階層データのそれぞれの画像を所定サイズに分割してなるタイル画像のデータと、前記階層データを構成する各階層の画像をタイル画像に分割した際の各領域であるタイル領域と当該領域を描画する際に用いる前記タイル画像のデータとを対応づけたインデックスデータと、を含む画像ファイルをメモリより読み出す機能と、

10

画像平面上の移動および解像度の変更を含む表示領域変更要求をユーザから受け付ける機能と、

前記表示領域変更要求に応じて新たに表示すべき領域を描画するのに必要なタイル画像のデータを、前記インデックスデータを参照することによって特定する機能と、

特定したタイル画像のデータをメモリより読み出し、新たに表示すべき領域を描画し表示装置に表示する機能と、

をコンピュータに実現させるコンピュータプログラムであって、

前記インデックスデータは、前記階層データの画像平面とそれに垂直な解像度の軸によって定義される仮想空間において前記階層データで構成される階層構造体を分割してなる領域ごとに、前記タイル領域ごとのインデックスデータをまとめたインデックスブロックの形式を有することを特徴とするコンピュータプログラム。

20

【請求項 2 4】

一の画像を異なる解像度で表す画像データによって構成される階層データのそれぞれの画像を所定サイズに分割してなるタイル画像のデータと、前記階層データを構成する各階層の画像をタイル画像に分割した際の各領域であるタイル領域と当該領域を描画する際に用いる前記タイル画像のデータとを対応づけたインデックスデータと、を含む画像ファイルをメモリより読み出す機能と、

前記画像ファイルを用いて表示される画像の更新情報を取得する機能と、

30

前記更新情報に基づき、更新対象の領域を描画する際に用いる新たなタイル画像のデータを生成しメモリに格納する機能と、

前記インデックスデータにおいて更新対象の領域に含まれるタイル領域に対応づけられるタイル画像のデータを、新たなタイル画像のデータに更新する機能と、

新たに生成したタイル画像のデータと、更新した前記インデックスデータを含む新たな画像ファイルを生成してメモリに格納する機能と、

をコンピュータに実現させるコンピュータプログラムであって、

前記インデックスデータは、前記階層データの画像平面とそれに垂直な解像度の軸によって定義される仮想空間において前記階層データで構成される階層構造体を分割してなる領域ごとに、前記タイル領域ごとのインデックスデータをまとめたインデックスブロックの形式を有することを特徴とするコンピュータプログラム。

40

【請求項 2 5】

画像の少なくとも一部をディスプレイに表示するために記憶装置から読み出される画像ファイルのデータ構造であって、

一の画像を異なる解像度で表す画像データを所定サイズに分割してなるタイル画像のデータと、

前記画像データを解像度順に階層化した階層データにおける各階層の画像をタイル画像に分割した際の各領域であるタイル領域と当該領域を描画する際に用いる前記タイル画像のデータとを対応づけたインデックスデータと、を対応づけ、

前記インデックスデータは、前記階層データの画像平面とそれに垂直な解像度の軸によ

50

って定義される仮想空間において前記階層データで構成される階層構造体を分割してなる領域ごとに、前記タイル領域ごとのインデックスデータをまとめたインデックスブロックの形式を有することを特徴とする画像ファイルのデータ構造。

【請求項 26】

一つの前記インデックスブロックで定義可能なタイル領域の数は、全てのインデックスブロックで同一であることを特徴とする請求項 25 に記載の画像ファイルのデータ構造。

【請求項 27】

前記インデックスデータは、ある解像度の画像に含まれるタイル画像のデータを流用して描画できる、当該解像度より高解像度の階層の画像のタイル領域に対しては無効データを対応づけられていることにより、前記タイル画像の流用によって画像を描画する旨を表したことを特徴とする請求項 25 および 26 のいずれかに記載の画像ファイルのデータ構造。

10

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、ディスプレイに表示する画像を拡大/縮小、または上下左右に移動させる画像処理技術に関する。

【背景技術】

【0002】

ゲームプログラムを実行するだけでなく、動画を再生できる家庭用エンタテインメントシステムが提案されている。この家庭用エンタテインメントシステムでは、GPUがポリゴンを用いた三次元画像を生成する(例えば特許文献1参照)。

20

【0003】

画像表示の目的に関わらず、画像をいかに効率よく表示するかは常に重要な問題となる。特に高精細な画像を高速に描画するためには様々な工夫が必要となり、例えばテクスチャデータを別に保持しマッピングを効率的に行う手法について提案がなされている(例えば非特許文献1および2参照)。

【先行技術文献】

【特許文献】

【0004】

【特許文献1】米国特許第6563999号公報

30

【非特許文献】

【0005】

【非特許文献1】Sylvain Fefevre, et. al., Unified Texture Management for Arbitrary Meshes, Repport de recherche, N5210, May 2004, Institut National De Recherche En Informatique Et En Automatique

【非特許文献2】Martin Kraus, et. al., Adaptive Texture Maps, Graphics Hardware (2002), pp1-10, The Eurographics Association

【発明の概要】

【発明が解決しようとする課題】

40

【0006】

画像が高精細化しても、データサイズを小さく抑えること、高速に描画することは、画像を応答性良く表示するためには常に重要な課題である。また、画像の一部などを修正する必要が生じた場合、上述のテクスチャマッピングなどデータ構造を工夫する技術を適用したとしても、一度構築したデータ構造を全て更新し直す必要があるなど、更新処理が煩雑になりやすいという問題があった。

【0007】

本発明はこのような課題に鑑みてなされたものであり、その目的は、サイズの大きな画像であってもデータサイズを抑えるとともに、効率的に生成、表示、修正を行える画像処理技術を提供することにある。

50

【課題を解決するための手段】

【0008】

本発明のある態様は画像ファイル生成装置に関する。この画像ファイル生成装置は、入力された画像データから、異なる解像度の複数の画像データを生成し、解像度順に階層化して構成される階層データを生成する画像階層生成部と、階層データを構成する各階層の画像データを所定サイズのタイル画像に分割する画像分割部と、階層データを構成する各階層の画像データの分割後の各タイル領域と、当該領域を描画する際に用いるタイル画像のデータとを対応づけたインデックスデータを作成するインデックス作成部と、タイル画像のデータとインデックスデータとを含む画像ファイルを生成する画像ファイル生成部と、を備え、インデックス作成部は、階層データの画像平面とそれに垂直な解像度の軸によって定義される仮想空間において階層データで構成される階層構造体を分割してなる領域ごとに、前記タイル領域ごとのインデックスデータをまとめたインデックスブロックの形式で、前記インデックスデータを生成することを特徴とする。

10

【0009】

本発明の別の態様は画像処理装置に関する。この画像処理装置は、一の画像で異なる解像度の画像データで構成される階層データのそれぞれの画像を所定サイズに分割してなるタイル画像データと、階層データを構成する各階層の画像データの分割後の各タイル領域と、当該領域を描画する際に用いるタイル画像のデータとを対応づけたインデックスデータを含む画像ファイルを保持する記憶装置と、画像平面上の移動および解像度の変更を含む表示領域変更要求をユーザから受け付ける入力情報取得部と、表示領域変更要求に応じて新たに表示すべき領域を描画するのに必要なタイル画像のデータを、インデックスデータを参照することによって特定するタイル画像特定部と、特定したタイル画像のデータを前記記憶装置より読み出し、新たに表示すべき領域を描画する表示画像処理部と、を備え、インデックスデータは、階層データの画像平面とそれに垂直な解像度の軸によって定義される仮想空間において階層データで構成される階層構造体を分割してなる領域ごとに、タイル領域ごとのインデックスデータをまとめたインデックスブロックの形式を有することを特徴とする。

20

【0010】

本発明の別の態様は画像処理装置に関する。この画像処理装置は、一の画像で異なる解像度の画像データで構成される階層データのそれぞれの画像を所定サイズに分割してなるタイル画像データと、階層データを構成する各階層の画像データの分割後の各タイル領域と、当該領域を描画する際に用いるタイル画像のデータとを対応づけたインデックスデータを含む画像ファイルを読み出す画像ファイル取得部と、画像ファイルを用いて表示される画像の更新情報を取得する更新情報取得部と、更新情報に基づき、更新対象の領域を描画する際に用いる新たなタイル画像のデータを生成するタイル画像生成部と、インデックスデータにおいて更新対象の領域に含まれるタイル領域に対応づけられるタイル画像のデータを、新たなタイル画像のデータに更新するインデックスデータ更新部と、タイル画像生成部が生成したタイル画像のデータと、前記インデックスデータ更新部が更新したインデックスデータを含む新たな画像ファイルを生成する画像ファイル生成部と、を備え、インデックスデータは、前記階層データの画像平面とそれに垂直な解像度の軸によって定義される仮想空間において前記階層データで構成される階層構造体を分割してなる領域ごとに、タイル領域ごとのインデックスデータをまとめたインデックスブロックの形式を有することを特徴とする。

30

40

【0011】

本発明のさらに別の態様は画像ファイル生成方法に関する。この画像ファイル生成方法は、入力された画像データから、異なる解像度の複数の画像データを生成し、解像度順に階層化して構成される階層データを生成しメモリに格納するステップと、階層データを構成する各階層の画像データをメモリより読み出し所定サイズのタイル画像に分割するステップと、階層データを構成する各階層の画像データの分割後の各タイル領域と、当該領域を描画する際に用いるタイル画像のデータとを対応づけたインデックスデータを作成しメ

50

メモリに格納するステップと、タイル画像のデータとインデックスデータとを含む画像ファイルを生成しメモリに格納するステップと、を含み、インデックスデータを作成するステップは、階層データの画像平面とそれに垂直な解像度の軸によって定義される仮想空間において階層データで構成される階層構造体を分割してなる領域ごとに、タイル領域ごとのインデックスデータをまとめたインデックスブロックの形式で、前記インデックスデータを生成することを特徴とする。

【0012】

本発明のさらに別の態様は画像処理方法に関する。この画像処理方法は、一の画像で異なる解像度の画像データで構成される階層データのそれぞれの画像を所定サイズに分割してなるタイル画像データと、階層データを構成する各階層の画像データの分割後の各タイル領域と、当該領域を描画する際に用いるタイル画像のデータとを対応づけたインデックスデータを含む画像ファイルをメモリより読み出すステップと、画像平面上の移動および解像度の変更を含む表示領域変更要求をユーザから受け付けるステップと、表示領域変更要求に応じて新たに表示すべき領域を描画するのに必要なタイル画像のデータを、インデックスデータを参照することによって特定するステップと、特定したタイル画像のデータをメモリより読み出し、新たに表示すべき領域を描画し表示装置に表示するステップと、を含み、インデックスデータは、階層データの画像平面とそれに垂直な解像度の軸によって定義される仮想空間において階層データで構成される階層構造体を分割してなる領域ごとに、タイル領域ごとのインデックスデータをまとめたインデックスブロックの形式を有することを特徴とする。

【0013】

本発明のさらに別の態様は画像処理方法に関する。この画像処理方法は、一の画像で異なる解像度の画像データで構成される階層データのそれぞれの画像を所定サイズに分割してなるタイル画像データと、階層データを構成する各階層の画像データの分割後の各タイル領域と、当該領域を描画する際に用いる前記タイル画像のデータとを対応づけたインデックスデータを含む画像ファイルをメモリより読み出すステップと、画像ファイルを用いて表示される画像の更新情報を取得するステップと、更新情報に基づき、更新対象の領域を描画する際に用いる新たなタイル画像のデータを生成しメモリに格納するステップと、インデックスデータにおいて更新対象の領域に含まれるタイル領域に対応づけられるタイル画像のデータを、新たなタイル画像のデータに更新するステップと、新たに生成したタイル画像のデータと、更新した前記インデックスデータを含む新たな画像ファイルを生成してメモリに格納するステップと、を含み、インデックスデータは、階層データの画像平面とそれに垂直な解像度の軸によって定義される仮想空間において階層データで構成される階層構造体を分割してなる領域ごとに、タイル領域ごとのインデックスデータをまとめたインデックスブロックの形式を有することを特徴とする。

【0014】

本発明のさらに別の態様は画像ファイルのデータ構造に関する。このデータ構造は、画像の少なくとも一部をディスプレイに表示するために記憶装置から読み出される画像ファイルのデータ構造であって、一の画像で異なる解像度の画像データを所定サイズに分割してなるタイル画像のデータと、画像データを解像度順に階層化した階層データにおける各画像データの分割後の各領域と当該領域を描画する際に用いるタイル画像のデータとを対応づけたインデックスデータと、を対応づけ、インデックスデータは、階層データの画像平面とそれに垂直な解像度の軸によって定義される仮想空間において階層データで構成される階層構造体を分割してなる領域ごとに、タイル領域ごとのインデックスデータをまとめたインデックスブロックの形式を有することを特徴とする。

【0015】

なお、以上の構成要素の任意の組合せ、本発明の表現を方法、装置、システム、コンピュータプログラムなどの間で変換したものもまた、本発明の態様として有効である。

【発明の効果】

【0016】

本発明によると、幅広い解像度で画像を表示するための画像データを効率的に構築できる画像処理装置を提供することができる。

【図面の簡単な説明】

【0017】

【図1】実施の形態1の形態にかかる情報処理システムの使用環境を示す図である。

【図2】図1の画像処理システムに適用できる入力装置の外観構成を示す図である。

【図3】実施の形態1において使用する画像データの階層構造の概念図である。

【図4】実施の形態1における情報処理装置の構成を示す図である。

【図5】実施の形態1における画像データの流れを模式的に示す図である。

【図6】実施の形態1における画像データの先読み処理を説明するための図である。

10

【図7】実施の形態1において階層データを表示する機能を有する制御部の構成を詳細に示す図である。

【図8】実施の形態1で用いる階層データのデータ構造を模式的に示す図である。

【図9】実施の形態1において表示対象の画像ファイルを生成する機能を有する制御部の構成を示す図である。

【図10】実施の形態1において画像に冗長性がある場合の元の画像とタイル画像との関係を説明する図である。

【図11】実施の形態1において画像に冗長性がある場合の元の画像とタイル画像との関係の別の例を説明する図である。

【図12】実施の形態1において冗長性によるタイル画像の共有をヘッダによって定義する手法を説明する図である。

20

【図13】実施の形態1において冗長性によるタイル画像の共有をインデックスブロックによって定義する手法を説明する図である。

【図14】図9で示した制御部が画像ファイルを生成する処理手順を示すフローチャートである。

【図15】実施の形態1において制御部が画像ファイルを表示するための処理手順を示すフローチャートである。

【図16】実施の形態1において画像を修正する機能を有する制御部の構成を示す図である。

【図17】実施の形態1における画像の修正例を示す図である。

30

【図18】図17のような修正におけるインデックスブロックのポインタの変化を模式的に示す図である。

【図19】実施の形態1において高解像度側に階層を追加する場合の処理を説明するための図である。

【図20】実施の形態1において追加すべき階層が属するインデックスブロックが存在しなかった場合のヘッダおよびインデックスブロックの変化を説明する図である。

【図21】実施の形態1において既存の画像に新たな領域を追加する処理を説明するための図である。

【図22】実施の形態1において新たな領域を追加する場合のヘッダおよびインデックスブロックの変化を説明する図である。

40

【図23】実施の形態1において階層構造体の頂点側に階層を追加する際にヘッダにおいて定義される領域の分割手法を説明する図である。

【図24】実施の形態1の画像処理装置においてユーザが画像を修正、改変する際の処理手順を示すフローチャートである。

【図25】実施の形態2における情報処理装置のうち制御部およびメインメモリの構成をより詳細に示す図である。

【図26】実施の形態2において階層データへのレンダリングの処理手順を示すフローチャートである。

【図27】実施の形態2においてスケーリング処理を行うスケーラをメインメモリに内蔵した場合の装置構成を示す図である。

50

【図 28】実施の形態 2 における GPU バッファのアドレッシングの様子を示す図である。

【図 29】実施の形態 2 における GPU バッファのアドレッシングの様子を示す図である。

【発明を実施するための形態】

【0018】

実施の形態 1

本実施の形態において処理対象とする画像のデータは、原画像を複数段階に縮小して生成した異なる解像度の画像からなる階層構造を有する。各階層の画像は一又は複数のタイル画像に分割する。たとえば最も解像度の低い画像は 1 つのタイル画像で構成し、最も解像度の高い原画像は、最も多い数のタイル画像で構成する。画像表示時は、描画に使用しているタイル画像を、表示画像が所定の解像度になったときに異なる階層のタイル画像に切り替えることで、拡大表示または縮小表示を迅速に行う。

10

【0019】

まず、このような階層構造を有する画像の基本的な表示態様について説明する。図 1 は、本発明の実施の形態を適用できる情報処理システム 1 の使用環境を示す。情報処理システム 1 は、画像処理を含むアプリケーションプログラムを実行する情報処理装置 10 と、情報処理装置 10 による処理結果を出力する表示装置 12 とを備える。表示装置 12 は、画像を出力するディスプレイおよび音声を出力するスピーカを有するテレビであってよい。

20

【0020】

表示装置 12 は、情報処理装置 10 に有線ケーブルで接続されてよく、また無線 LAN (Local Area Network) などにより無線接続されてもよい。情報処理システム 1 において、情報処理装置 10 は、ケーブル 14 を介してインターネットなどの外部ネットワークに接続し、階層化された圧縮画像データを含むコンテンツなどをダウンロードして取得してもよい。なお情報処理装置 10 は、無線通信により外部ネットワークに接続してもよい。

【0021】

情報処理装置 10 は、ユーザからの要求に応じて、表示装置 12 のディスプレイに表示する画像の拡大 / 縮小処理や、上下左右方向への移動処理など、表示領域を変更する処理を行う。ユーザが、ディスプレイに表示された画像を見ながら入力装置を操作すると、入力装置が、表示領域の変更要求信号を情報処理装置 10 に送信する。

30

【0022】

図 2 は、入力装置 20 の外観構成を示す。入力装置 20 は、ユーザが操作可能な操作手段として、十字キー 21、アナログスティック 27a、27b と、4 種の操作ボタン 26 を備える。4 種の操作ボタン 26 は、ボタン 22、x ボタン 23、ボタン 24 およびボタン 25 から構成される。

【0023】

情報処理システム 1 において、入力装置 20 の操作手段には、表示画像の拡大 / 縮小要求、および上下左右方向へのスクロール要求を入力するための機能が割り当てられる。たとえば、表示画像の拡大 / 縮小要求の入力機能は、右側のアナログスティック 27b に割り当てられる。ユーザはアナログスティック 27b を手前に引くことで、表示画像の縮小要求を入力でき、また手前から押すことで、表示画像の拡大要求を入力できる。また、表示領域の移動要求の入力機能は、十字キー 21 に割り当てられる。ユーザは十字キー 21 を押下することで、十字キー 21 を押下した方向への移動要求を入力できる。なお、画像変更要求の入力機能は別の操作手段に割り当てられてもよく、たとえばアナログスティック 27a に、スクロール要求の入力機能が割り当てられてもよい。

40

【0024】

また後に述べる各種機能を実現するため、入力装置 20 にはさらに、画像に表示されたカーソルを移動させたり、ファイルやコマンドを選択したりする機能も割り当てる。あるいは入力装置 20 は、ポインティングデバイス、マウス、キーボード、タッチパネルなど

50

一般的な入力装置で実現してもよい。上述したような機能の割り当ては、入力装置 20 の種類によって適宜決定してよい。

【0025】

入力装置 20 は、入力された表示領域変更要求信号などを情報処理装置 10 に伝送する機能をもち、本実施の形態では情報処理装置 10 との間で無線通信可能に構成される。入力装置 20 と情報処理装置 10 は、Bluetooth (ブルートゥース) (登録商標) プロトコルや IEEE802.11 プロトコルなどを用いて無線接続を確立してもよい。なお入力装置 20 は、情報処理装置 10 とケーブルを介して接続して、表示領域変更要求信号などを情報処理装置 10 に伝送してもよい。

【0026】

図 3 は、本実施の形態において使用する画像データの階層構造の概念図を示す。画像データは、深さ (Z 軸) 方向に、第 0 階層 30、第 1 階層 32、第 2 階層 34 および第 3 階層 36 からなる階層構造を有する。なお同図においては 4 階層のみ示しているが、階層数はこれに限定されない。以下、このような階層構造をもつ画像データを「階層データ」とよぶ。ただし同図の階層データは概念的なものであり、実際には後に述べるように複数のデータセットで階層データを表現する。

【0027】

図 3 に示す階層データは 4 分木の階層構造を有し、各階層は 1 以上のタイル画像 38 で構成される。すべてのタイル画像 38 は同じ画素数をもつ同一サイズに形成され、たとえば 256×256 画素を有する。各階層の画像データは、一つの画像を異なる解像度で表現しており、最高解像度をもつ第 3 階層 36 の原画像を複数段階に縮小して、第 2 階層 34、第 1 階層 32、第 0 階層 30 の画像データが生成される。たとえば第 N 階層の解像度 (N は 0 以上の整数) は、左右 (X 軸) 方向、上下 (Y 軸) 方向ともに、第 (N + 1) 階層の解像度の $1/2$ であってよい。

【0028】

情報処理装置 10 において、階層データは、所定の圧縮形式で圧縮された状態で記憶装置に保持されており、ディスプレイに表示される前に記憶装置から読み出されてデコードされる。本実施の形態の情報処理装置 10 は、複数種類の圧縮形式に対応したデコード機能を有し、たとえば S3TC 形式、JPEG 形式、JPEG 2000 形式の圧縮データをデコード可能とする。

【0029】

階層データの階層構造は、図 3 に示すように、左右方向を X 軸、上下方向を Y 軸、深さ方向を Z 軸として設定され、仮想的な 3 次元空間を構築する。情報処理装置 10 は、入力装置 20 から供給される表示領域変更要求信号から表示画像の変更量を導出すると、その変更量を用いて仮想空間におけるフレームの 4 隅の座標 (フレーム座標) を導出する。仮想空間におけるフレーム座標は、後述するメインメモリへの圧縮データのロードおよび表示画像の生成処理に利用される。なお、仮想空間におけるフレーム座標の代わりに、情報処理装置 10 は、階層を特定する情報と、その階層におけるテクスチャ座標 (UV 座標) を導出してもよい。以下、階層特定情報およびテクスチャ座標の組み合わせも、フレーム座標と呼ぶ。

【0030】

図 4 は情報処理装置 10 の構成を示している。情報処理装置 10 は、無線インタフェース 40、スイッチ 42、表示処理部 44、ハードディスクドライブ 50、記録媒体装着部 52、ディスクドライブ 54、メインメモリ 60、バッファメモリ 70 および制御部 100 を有して構成される。表示処理部 44 は、表示装置 12 のディスプレイに表示するデータをバッファするフレームメモリを有する。

【0031】

スイッチ 42 は、イーサネットスイッチ (イーサネットは登録商標) であって、外部の機器と有線または無線で接続して、データの送受信を行うデバイスである。スイッチ 42 は、ケーブル 14 を介して外部ネットワークに接続し、サーバからコンテンツのデータな

10

20

30

40

50

どを受信できるように構成される。またスイッチ 42 は無線インタフェース 40 に接続し、無線インタフェース 40 は、所定の無線通信プロトコルで入力装置 20 と接続する。入力装置 20 においてユーザから入力された信号は、無線インタフェース 40、スイッチ 42 を経由して、制御部 100 に供給される。

【0032】

ハードディスクドライブ 50 は、データを記憶する記憶装置として機能する。スイッチ 42 を介して受信された階層データは、ハードディスクドライブ 50 に格納される。記録媒体装着部 52 は、メモリカードなどのリムーバブル記録媒体が装着されると、リムーバブル記録媒体からデータを読み出す。ディスクドライブ 54 は、読出専用の ROM ディスクが装着されると、ROM ディスクを駆動して認識し、データを読み出す。ROM ディスクは、光ディスクや光磁気ディスクなどであってよい。階層データはこれらの記録媒体に格納されていてもよい。

10

【0033】

制御部 100 は、マルチコア CPU を備え、1つの CPU の中に1つの汎用的なプロセッサコアと、複数のシンプルなプロセッサコアを有する。汎用プロセッサコアは PPU (PowerPC Processor Unit) と呼ばれ、残りのプロセッサコアは SPU (Synergistic Processor Unit) と呼ばれる。

【0034】

制御部 100 は、メインメモリ 60 およびバッファメモリ 70 に接続するメモリコントローラを備える。PPU はレジスタを有し、演算実行主体としてメインプロセッサを備えて、実行するアプリケーションにおける基本処理単位としてのタスクを各 SPU に効率的に割り当てる。なお、PPU 自身がタスクを実行してもよい。SPU はレジスタを有し、演算実行主体としてのサブプロセッサとローカルな記憶領域としてのローカルメモリを備える。ローカルメモリは、バッファメモリ 70 として使用されてもよい。

20

【0035】

メインメモリ 60 およびバッファメモリ 70 は記憶装置であり、RAM (ランダムアクセスメモリ) として構成される。SPU は制御ユニットとして専用の DMA (Direct Memory Access) コントローラをもち、メインメモリ 60 とバッファメモリ 70 の間のデータ転送を高速に行うことができ、また表示処理部 44 におけるフレームメモリとバッファメモリ 70 の間で高速なデータ転送を実現できる。本実施の形態の制御部 100 は、複数の SPU を並列動作させることで、高速な画像処理機能を実現する。表示処理部 44 は、表示装置 12 に接続されて、ユーザからの要求に応じた画像処理結果を出力する。

30

【0036】

本実施の形態の情報処理装置 10 は、表示画像の拡大/縮小処理や表示領域の移動処理を行う際に表示画像をスムーズに変更させるために、圧縮画像データの一部をハードディスクドライブ 50 からメインメモリ 60 にロードしておく。また、メインメモリ 60 にロードした圧縮画像データのさらに一部をデコードしてバッファメモリ 70 に格納しておく。これにより、後の必要なタイミングで、表示画像の生成に使用する画像を瞬時に切り替えることが可能となる。

【0037】

40

図 5 は本実施の形態における画像データの流れを模式的に示している。まず階層データはハードディスクドライブ 50 に格納されている。ハードディスクドライブ 50 に代わり、記録媒体装着部 52 やディスクドライブ 54 に装着された記録媒体が保持していてもよい。あるいは、情報処理装置 10 がネットワークを介して接続した画像サーバから階層データをダウンロードするようにしてもよい。ここでの階層データは上述のとおり、S3TC 形式などによる固定長圧縮、あるいは JPEG 形式などによる可変長圧縮がなされている。

【0038】

この階層データのうち、一部の画像データを圧縮した状態のままメインメモリ 60 にロードする (S10)。ここでロードする領域は、現在の表示画像の仮想空間における近傍

50

や、画像の内容、ユーザの閲覧履歴等に基づいて、高頻度で表示要求がなされると予測される領域など、あらかじめ定めた規則によって決定する。ロードは、画像変更要求がなされたときのみならず、例えば所定の時間間隔で随時行う。これによりロード処理が一時期に集中しないようにする。

【 0 0 3 9 】

次に、メインメモリ 6 0 に格納されている圧縮画像データのうち、表示に必要な領域のタイル画像、または必要と予測される領域のタイル画像のデータをデコードし、バッファメモリ 7 0 に格納する (S 1 2)。バッファメモリ 7 0 は、少なくとも 2 つのバッファ領域 7 2、7 4 を含む。各バッファ領域 7 2、7 4 のサイズは、フレームメモリ 9 0 のサイズよりも大きく設定され、入力装置 2 0 から表示領域変更要求信号が入力された場合に、ある程度の量の変更要求に対しては、バッファ領域 7 2、7 4 に展開した画像データで表示画像を生成できるようにする。

10

【 0 0 4 0 】

バッファ領域 7 2、7 4 の一方は、表示画像の生成に用いる画像を保持するための表示用バッファであり、他方は、以後、必要と予測される画像を準備するためのデコード用バッファである。図 5 の例では、バッファ領域 7 2 が表示用バッファ、バッファ領域 7 4 がデコード用バッファで、表示領域 6 8 が表示されているものとする。後述する先読み処理によってデコード用バッファに格納する画像は、表示用バッファに格納されている画像と同一階層の画像でもよいし、縮尺の異なる別階層の画像でもよい。

【 0 0 4 1 】

20

次に、表示用バッファであるバッファ領域 7 2 に格納された画像のうち表示領域 6 8 の画像を、フレームメモリ 9 0 に描画する (S 1 4)。この間に、新たな領域の画像が必要に応じてデコードされ、バッファ領域 7 4 に格納される。格納が完了したタイミングや表示領域 6 8 の変更量などに応じて、表示用バッファとデコード用バッファを切り替える (S 1 6)。これにより、表示領域の移動や縮尺率の変更などに対し表示画像をスムーズに切り替えることができる。

【 0 0 4 2 】

図 6 は、先読み処理を説明するための図である。図 6 は、階層データの構造を示しており、各階層は L 0 (第 0 階層)、L 1 (第 1 階層)、L 2 (第 2 階層)、L 3 (第 3 階層) と表現されている。図 6 に示す階層データ構造において、深さ (Z 軸) 方向における位置は解像度を示し、L 0 に近い位置ほど解像度が低く、L 3 に近い位置ほど解像度は高い。なおディスプレイに表示される画像の大きさに注目すると、深さ方向における位置は、縮尺率に対応し、L 3 の表示画像の縮尺率を 1 とすると、L 2 における縮尺率は 1 / 4、L 1 における縮尺率は 1 / 1 6 となり、L 0 における縮尺率は 1 / 6 4 となる。

30

【 0 0 4 3 】

したがって深さ方向において、表示画像が L 0 側から L 3 側へ向かう方向に変化する場合は、表示画像は拡大していき、L 3 側から L 0 側へ向かう方向に変化する場合は、表示画像は縮小していく。矢印 8 0 は、ユーザからの表示領域変更要求信号が、表示画像の縮小を要求しており、縮尺率 1 / 4 (L 2) をまたいだ様子を示している。情報処理装置 1 0 では、タイル画像 3 8 として用意している L 1、L 2 の深さ方向の位置を、深さ方向の先読み境界として設定し、画像変更要求信号が先読み境界をまたぐと、先読み処理を開始する。

40

【 0 0 4 4 】

表示画像の縮尺率が L 2 の近傍にある場合、表示画像は、L 2 (第 2 階層) のタイル画像を用いて作成される。具体的には、表示する画像の縮尺率が、L 1 タイル画像と L 2 タイル画像の切替境界 8 2 と、L 2 タイル画像と L 3 タイル画像の切替境界 8 4 の間にある場合に、L 2 タイル画像が利用される。したがって、矢印 8 0 に示すように画像の縮小処理が要求されると、L 2 のタイル画像が拡大された画像から、縮小された画像に変換されて表示される。一方、画像変更要求信号から予測される将来必要なタイル画像 3 8 を特定して、デコードしておく。図 6 の例では、表示領域変更要求信号による要求縮尺率が L 2

50

をまたいだときに、情報処理装置 10 は、縮小方向にある L1 の対応するタイル画像 38 をハードディスクドライブ 50 またはメインメモリ 60 から先読みしてデコードし、バッファメモリ 70 に書き込む。

【0045】

なお以上は深さ方向の先読み処理について説明したが、上下左右方向の先読み処理についても同様に処理される。具体的には、バッファメモリ 70 に展開されている画像データに先読み境界を設定しておき、画像変更要求信号による表示位置が先読み境界をまたいだときに、先読み処理が開始されるようにする。

【0046】

図 7 は本実施の形態において、上述の階層データを表示する機能を有する制御部 100 a の構成を詳細に示している。制御部 100 a は、入力装置 20 からユーザが入力した情報を取得する入力情報取得部 102、新たに表示すべき領域を含むタイル画像を特定するタイル画像特定部 110、新たにロードすべき画像データを決定するロードブロック決定部 106、必要な画像ブロックをハードディスクドライブ 50 からロードするロード部 108 を含む。制御部 100 a はさらに、圧縮画像データをデコードするデコード部 112、および表示画像を描画する表示画像処理部 114 を含む。

【0047】

図 7 において、さまざまな処理を行う機能ブロックとして記載される各要素は、ハードウェア的には、CPU (Central Processing Unit)、メモリ、その他の LSI で構成することができ、ソフトウェア的には、メモリにロードされたプログラムなどによって実現される。既述したように、制御部 100 は 1 つの PPU と複数の SPU とを有し、PPU および SPU がそれぞれ単独または協同して、各機能ブロックを構成できる。したがって、これらの機能ブロックがハードウェアのみ、ソフトウェアのみ、またはそれらの組合せによっていろいろな形で実現できることは当業者には理解されるところであり、いずれかに限定されるものではない。

【0048】

入力情報取得部 102 は、ユーザが入力装置 20 に対して入力した、画像表示の開始 / 終了、表示領域の移動、表示画像の拡大、縮小などの指示内容を取得する。タイル画像特定部 110 は、現在の表示領域のフレーム座標とユーザが入力した表示領域変更要求の情報に従い、新たに表示すべき領域を含むタイル画像を特定する。そして当該タイル画像が既にメインメモリ 60 にロードされていれば、デコード部 112 に、ロードされていなければロードブロック決定部 106 に、当該情報を供給する。タイル画像特定部 110 はその時点の表示画像の描画に必要な画像の他、以後必要と予測されるタイル画像も特定してよい。

【0049】

ロードブロック決定部 106 は、タイル画像特定部 110 からの情報に基づき、ハードディスクドライブ 50 からメインメモリ 60 へ新たにロードすべき、複数のタイル画像からなる画像ブロックを特定し、ロード部 108 にロード要求を発行する。ロードブロック決定部 106 は、ロード部 108 がロード処理中でない状態において、例えば所定の時間間隔、あるいは、ユーザが表示領域変更要求を行った際など、所定のタイミングでロード要求を行ってもよい。ロード部 108 は、ロードブロック決定部 106 からの要求に従い、実際のロード処理を行う。

【0050】

デコード部 112 は、タイル画像特定部 110 から取得したタイル画像の情報に基づき、メインメモリ 60 からタイル画像のデータを読み出しデコードし、デコード用バッファまたは表示用バッファにデコード後のデータを格納する。表示画像処理部 114 は、新たな表示画像のフレーム座標に基づき、バッファメモリ 70 の表示用バッファから対応する画像データを読み出し、表示処理部 44 のフレームメモリに描画する。

【0051】

次に、これまで述べた階層構造を有する画像データの表示技術において、画像データを

10

20

30

40

50

より効率的に構築し出力するための実施の形態について説明する。図8は本実施の形態における階層データのデータ構造を模式的に示している。本実施の形態では階層データを、ヘッダ150、インデックスブロック160、タイル画像170、の3つのデータからなる構造とする。ヘッダ150およびインデックスブロック160は、階層データが形成する仮想空間内での位置に基づき、処理対象のタイル画像を特定するためのインデックスである。本実施の形態では、仮想空間における階層構造を領域分割し、領域ごとにタイル画像データへのインデックスをまとめることにより、タイル画像へのアクセスを効率化する。

【0052】

ヘッダ150は、図3に示したような、仮想空間におけるピラミッド状の階層構造体を分割してなる複数の領域を定義し、領域ごとにインデックスブロック160のいずれかへのポインタが設定されたデータである。図8に示すヘッダ150において三角形151は階層構造体を横から見た形状を示しており、水平な破線は各階層の画像154a、154b、154c、154d、154e、・・・を示している。このような階層構造体を、例えば実線で示すように領域分割し、領域152a、領域152b、領域152cなどの領域を定義する。ヘッダ150では原則として、各領域に対してインデックスブロック160のいずれかへのポインタが設定される。

【0053】

図8の例は、領域152aに対しインデックスブロック160aへのポインタ(矢印A)、領域152bに対しインデックスブロック160bへのポインタ(矢印B)、領域152cに対しインデックスブロック160cへのポインタ(矢印C)が設定されている様子を示している。ただし後に述べるように、ヘッダで定義される領域の一部に対して特定のインデックスブロックを指さないヌルポインタを設定してもよい。

【0054】

インデックスブロック160は、ヘッダ150で定義された構造体の領域ごとに生成され、各領域に属する複数階層の画像上の位置に対しタイル画像170のいずれかへのポインタが設定されるデータである。図8に示すように、ヘッダ150における一つの領域に対応する一つのインデックスブロックは、当該領域に属する、複数階層の画像の少なくとも一部の領域のデータを含む。同図の例では、領域152aに対応するインデックスブロック160aは、領域152aに属する2階層分の画像154a、154b全体のデータを保持する。領域152bに対応するインデックスブロック160bは、領域152bに属する3階層分の画像154c、154d、154e全体のデータを保持する。

【0055】

領域152cに対応するインデックスブロック160cは、領域152cに属する3階層分の画像154f、154g、154hの一部の領域のデータを保持する。そして各階層の画像平面をタイル画像のサイズに分割して生成したタイル領域ごとに、当該タイル領域の実際の画像データであるタイル画像170のいずれかを指すポインタを保持する。ただし後述するように、一つのタイル領域の描画を複数の情報を用いて行うようにした場合は、一つのタイル領域が複数のタイル画像を指すポインタを保持してもよい。同図において各インデックスブロック160a、160b、160cにそれぞれ示される3平面は、対応する領域に属する階層に対応し、各階層上の小さい矩形のそれぞれがタイル領域を示している。

【0056】

図8の例は、インデックスブロック160bにおいて、タイル領域164aに対しタイル画像170aへのポインタ(矢印D)が設定され、タイル領域164bに対しタイル画像170bへのポインタ(矢印E)が設定されている様子を示している。ただし後に述べるように、インデックスブロックで定義されるタイル領域の一部は特定のタイル画像へのポインタを設定せずヌルポインタとしてもよい。タイル画像170は、圧縮されたタイル画像のデータである。このように階層データを構成することにより、タイル画像特定部110は、表示すべき画像フレームの仮想空間におけるフレーム座標から、ヘッダ150、

10

20

30

40

50

インデックスブロック160、と辿り、当該画像フレームに含まれるタイル画像170を特定することができる。

【0057】

図8では、インデックスブロック160a、160b、160cはそれぞれ、4×4、8×8、16×16のタイル領域で構成される3階層の画像領域分のインデックスデータ、すなわちタイル画像170へのポインタを設定できるように示されている。このように、一つのインデックスブロック160で定義できるタイル領域の数を統一すると、例えばハードディスクドライブ50からメインメモリ60へインデックスブロック160のいずれかを読み出す際、どのインデックスブロックであっても必要となる記憶領域のサイズが変化せず、記憶領域の管理が容易になる。

10

【0058】

したがって、ヘッダ150において定義される領域は、インデックスブロック160が同じサイズとなるように生成されることが望ましい。階層データが図3で示すように、第0階層が1つのタイル画像である4分木の階層構造を有するとき、図8の領域152bに属する3階層の画像、すなわち第2階層の画像154c、第3階層の画像154d、第4階層の画像154eはそれぞれ、4×4、8×8、16×16のタイル領域からなる。

【0059】

したがって一つのインデックスブロック160で定義できる領域を上述のとおり4×4、8×8、16×16のタイル領域からなる画像領域とした場合、当該3階層の画像全体のデータを、そのまま1つのインデックスブロック160bに納めることができる。すなわち、階層データにおける第2階層の画像154c、第3階層の画像154d、第4階層の画像154eの全領域を、インデックスブロック160bの第0階層162d、第1階層162e、第2階層162fの全領域にそれぞれ対応させて、タイル画像へのポインタを設定することができる。

20

【0060】

一方、ヘッダ150における領域152aは第0階層および第1階層の2階層の画像154a、154bを含み、第0階層の画像154aは1つのタイル領域、第1階層の画像154bは2×2のタイル領域からなるため、1つのインデックスブロック160aで定義できる領域と比較し、階層数およびタイル領域の数が少ない。このような場合、階層データにおける第0階層の画像154aをインデックスブロック160aの第0階層162aの一部、例えば左上の1つのタイル領域に対応させ、タイル画像へのポインタを設定する。第1階層の画像154bも同様に、インデックスブロック160aの第1階層162bの一部、例えば左上の2×2のタイル画像に対応させ、タイル画像へのポインタを設定する。そして、インデックスブロック160aにおけるその他のタイル領域はヌルポインタを設定する。

30

【0061】

ヘッダ150における領域152cは第5階層、第6階層、第7階層の3階層の画像154f、154g、154hを含むが、これらの画像は1つのインデックスブロック160で定義できるタイル領域の数より多いタイル画像からなるため、図8に示すように画像平面を分割することにより、各階層の一部の領域をインデックスブロック160cの第0階層、第1階層、第2階層にそれぞれ対応させる。各インデックスブロックに属する複数階層の画像は解像度の異なる同一の画像領域となるように領域分割する。

40

【0062】

なお図8で示した階層構造体の領域分割の態様はあくまで例示であり、元の階層データの構造や望ましいインデックスブロックのデータサイズなどによって適宜決定してよい。また後述するように、元の画像の更新に応じて、階層データの分割態様を変化させてもよい。

【0063】

次に、これまで述べたデータ構造を有する階層データを含む画像ファイルを生成する態様について説明する。この態様も図1で示した情報処理システム、図4で示した情報処理

50

装置 10 と同様の装置構成で実現できる。図 9 は、本実施の形態において表示対象の画像ファイルを生成する機能を有する制御部 100b の構成を示している。なお制御部 100b は図 7 の制御部 100a に示した、画像表示を行うための機能を備えていてもよいが、ここでは図示を省略している。一方、図 9 に示した機能のみを有する制御部 100b を備えた情報処理装置 10 を、表示機能を発揮する情報処理装置とは別に設けてもよい。

【0064】

制御部 100b は、ハードディスクドライブ 50 に格納された画像データを読み出し階層化する画像階層生成部 120、各階層の画像をタイル画像に分割する画像分割部 122、各階層の画像を解析し冗長性を検出する冗長性検出部 124、冗長性を考慮して、ヘッダ、インデックスブロックのデータを生成するヘッダ・インデックスブロック生成部 126、および最終的に出力する、タイル画像、ヘッダ、インデックスブロックを含む画像ファイルを生成する画像ファイル生成部 128 を含む。

10

【0065】

画像階層生成部 120 は、ハードディスクドライブ 50 から、ファイル作成対象の画像のデータを読み出す。この画像データはある解像度を有する一枚の画像のデータでよい。処理対象の画像データは、図に示すように入力装置 20 を介してユーザが指定してもよいし、原画像を取得した別の機能ブロック（図示せず）からの要求を受け付けてもよい。画像階層生成部 120 はさらに、読み出した画像のデータを所定の解像度へ段階的に縮小した画像のデータを生成することにより、元の画像を含めた階層構造の画像データを生成する。

20

【0066】

画像分割部 122 は、各階層の画像を所定のサイズに分割してタイル画像のデータを生成する。生成したタイル画像はメインメモリ 60 に格納する。この際、各タイル画像が元の画像のどの位置の画像であるかを、識別番号を付与するなどして管理する。

【0067】

冗長性検出部 124 は各階層の画像を解析することにより、同一階層内、および階層間で画像の冗長性を検出する。同一階層内での冗長性としては例えば、複数のタイル領域に渡って同じタイル画像のデータを用いることができる場合が考えられる。また階層間での冗長性としては、低解像度の階層の画像を拡大した画像と高解像度の階層の画像とで見た目の差が大きい場合が考えられる。このように冗長性がある領域は、個々にタイル画像のデータを持たずとも、あるタイル画像のデータを流用することで表示が可能である。このようにすることで、画像データの圧縮を実現する。具体的な手法は後に述べる。

30

【0068】

ヘッダ・インデックスブロック生成部 126 は、上述のヘッダおよびインデックスブロックのデータを作成する。冗長性が検出されない画像では上述のように、ヘッダで定義される全ての領域に対しインデックスブロックのいずれかへのポインタが設定され、さらにインデックスブロックで定義される全てのタイル領域に対し、タイル画像へのポインタが設定される。一方、冗長性を有する画像領域については、ヘッダあるいはインデックスブロックにヌルポインタを設定することでタイル画像のデータ共有を実現する。

【0069】

画像ファイル生成部 128 は、インデックスブロックにおいて設定されたポインタが指すタイル画像のデータを読み出し、順につなげることによって、最終的な画像データを生成する。そして画像データ、ヘッダ、インデックスブロックを含む画像ファイルを生成して、最終的な出力データとする。上述のように、画像の冗長性を検出して、複数の領域の描画に一つのタイル画像のデータを用いるようにすることにより、画像データのサイズを効率的に圧縮することができる。本実施の形態では、解像度の異なる同一の画像を画像データに含むため、異なる階層の画像であっても同じタイル画像のデータを利用することによってデータの高率圧縮が可能である。

40

【0070】

図 10 は画像に冗長性がある場合の元の画像とタイル画像との関係を説明する図である

50

。同図において画像180aおよび画像180bは、同じ領域を表す2つの階層の画像である。すなわち画像180aは画像180bを縮小して得られる。画像180aおよび画像180b上に示した格子はタイル画像に分割する際の境界線である。同図では説明のために画像180a、180bのタイル領域およびタイル画像170のそれぞれに識別番号を付与している。

【0071】

同図において画像180aおよび画像180bには、楕円形と三角形の図形が描かれ、それ以外の領域は背景として単色に塗りつぶされているとする。そのような画像の各領域から切り出されたタイル画像170のデータは、本来の画像上の並びと独立してメインメモリ60などに保持しておく。ここで画像180aにおける単色の塗りつぶし領域であるタイル領域「4」は、そこから切り出されたタイル画像「4」と対応する。このようなとき、インデックスブロック160において、タイル領域「4」に対しタイル画像「4」へのポインタを設定する。

10

【0072】

画像180bは、画像180aの高解像度画像であるから、2つの画像の各タイル領域の対応関係は拡大率によって容易に導出できる。同図では例えば、画像180aのタイル領域「4」と、画像180bのタイル領域「15」、「16」、「19」、「20」が対応する。したがって、画像180aのタイル領域「4」が単色の塗りつぶし領域であれば、画像180bのタイル領域「15」、「16」、「19」、「20」は全て塗りつぶし領域であることが特定できる。すなわちこれらの領域の表示にはタイル画像「4」があれば済むことになる。

20

【0073】

同様に、画像180aを画像180bの解像度まで拡大したときに塗りつぶし領域となる領域を、画像180aを走査することによって判別すると、タイル領域「3」の下半分の領域、すなわち画像180bのタイル領域「17」、「18」が塗りつぶし領域であることがわかる。したがってこれらの領域もタイル画像「4」を用いることにより表示することが可能である。このようにタイル画像を共有することにより、タイル画像のデータ圧縮が可能となる。さらに後に述べるように、低解像度のタイル画像によって表示可能な高解像度の画像の領域は特定のポインタを持たないようにすることで、ヘッダやインデックスブロックのデータ圧縮も可能となる。

30

【0074】

本実施の形態で用いる画像データは、複数の解像度の同一の画像を処理対象とするため、上記の冗長性検出処理を最も小さい解像度の画像から順に繰り返すことにより、高解像度の大きなサイズの画像についても、計算負荷を抑えて冗長性を有する領域を特定することができる。なお上述の説明では単色の塗り潰し領域を検出して、同じ塗りつぶし領域でタイル画像のデータを共有する例を述べたが、同様の処理は、およそ単色の領域、ランダムな模様や繰り返しの模様のみ領域などでも有効である。これらの領域の検出は、領域ごとのカラーヒストグラムや周波数解析などによって低解像度の画像から実施していく。

【0075】

図11は冗長性がある場合の元の画像とタイル画像との関係の別の例を説明する図である。画像180aおよび画像180bは図10で示したのと同様である。同図において画像180aのタイル領域「2」は、画像180bのタイル領域「7」、「8」、「11」、「12」に対応する。画像180bは画像180aより高解像度の画像であるため、これらの領域の4つのタイル画像は領域「2」から切り出したタイル画像「2」を単に拡大した画像より多くの情報を含むのが一般的である。

40

【0076】

しかし元の画像が写真などで、画像180bの領域「7」、「8」、「11」、「12」にピントが合っていない場合など、タイル画像「2」を拡大した画像と大きな差がない場合がある。このような場合、上述の塗りつぶし領域と同様、領域「7」、「8」、「11」、「12」を表示する際、そこから切り出したタイル画像ではなくタイル画像「2

50

」を用い、それを拡大して表示しても、元の画像と大差ない画像を表示させることができる。このような冗長性は、低解像度画像を拡大した画像と高解像度の画像との差がしきい値以下である場合や、高解像度の画像を周波数解析したとき、しきい値以下の周波数帯のみ含む場合などに検出できる。

【 0 0 7 7 】

図 1 2 は、上述のような冗長性によるタイル画像の共有をヘッダによって定義する手法を説明する図である。このときヘッダ 1 5 0 a で定義される階層構造体の領域中、別の領域に含まれる階層の画像を用いて表示が可能な領域は、特定のインデックスブロックへのポインタを設定せずヌルポインタとする。図 1 2 では、ヘッダ 1 5 0 a の構造体において、網掛けで示した領域には特定のインデックスブロックへのポインタ、すなわち有効なポインタが設定され、白抜きの領域にはヌルポインタを設定するとする。

10

【 0 0 7 8 】

このようなヘッダ 1 5 0 a を有する階層データにおいて画像領域 1 5 6 を表示する場合、当該画像を含む領域はヌルポインタとなっている。この場合、階層データの仮想空間を、画像が縮小される方向に辿っていき、同じ画像領域 1 5 6 の縮小画像を含み有効なポインタが設定されている領域を探索する（矢印 F）。同図においては領域 1 5 2 d に有効なポインタが設定されているため、当該ポインタが指すインデックスブロック 1 6 0 d のデータを取得する（矢印 G）。

【 0 0 7 9 】

そしてインデックスブロック 1 6 0 d がデータを保持する階層のうち、最も解像度の大きい第 2 階層において画像領域 1 5 6 を含むタイル領域を特定し、それに対応づけられたポインタを取得する。画像領域 1 5 6 が領域 1 5 2 d の第 2 階層の画像平面上のどこに位置するかは、階層データにおける各階層の縮尺率によって容易に計算できるため、上記のタイル領域の特定はそれに基づき実施する。

20

【 0 0 8 0 】

そして取得したポインタが指すタイル画像 1 7 0 c のデータを取得し（矢印 H）、それを用いて画像領域 1 5 6 の表示画像を生成する。ここで画像領域 1 5 6 がタイル画像のサイズを有するとすると、画像領域 1 5 6 を描画するには同図に示すように、縮小画像であるタイル画像 1 7 0 c の一部の領域 1 7 2 を拡大する。上述のとおり、画像領域 1 5 6 に対応する領域 1 7 2 の位置は、両者の縮尺率によって容易に計算できる。

30

【 0 0 8 1 】

このように、ヘッダ 1 5 0 a においてヌルポインタを定義可能とすることにより、ヘッダ 1 5 0 a が定義する全ての領域に対しインデックスブロックを用意する必要がなくなり、データの圧縮効率が向上する。なおヌルポインタを設定する領域は必ずしも高解像度側になくてもよく、図 1 2 に示すように、低解像度の領域と高解像度の領域に挟まれた領域 1 5 2 e に対し中抜きの状態でヌルポインタを設定することもできる。この場合、領域 1 5 2 e に含まれる画像を表示する際は、解像度の低い側の領域 1 5 2 f に属する階層の画像を拡大して表示し、表示画像の解像度が増加して領域 1 5 2 s に入ったら、当該領域に対して設定されたポインタが指すインデックスブロックを参照してタイル画像を取得する。

40

【 0 0 8 2 】

このようにすることで、表示画像がある解像度を超えるとこれまでと異なる画像が表示されるようにすることができる。例えば解像度を上げると新聞紙面の漢字に対して振り仮名が出現したり、写真の背景に別の物が出現したり、といった表示上の演出が可能となる。なおこのように、ヘッダでは基本的に任意の領域に対してヌルポインタを設定することが可能であるが、階層構造体の頂点、すなわち解像度が最も小さい画像を含む領域 1 5 2 g に対しては必ず有効なポインタを設定する。これにより、その他のいかなる領域に対してヌルポインタを設定しても、階層構造体を辿ることにより、少なくとも領域 1 5 2 g には到達し、表示画像を生成できる。

【 0 0 8 3 】

50

図13は、上述のような冗長性によるタイル画像の共有をインデックスブロックによって定義する手法を説明する図である。ここでインデックスブロック160eに属する画像領域中、同じインデックスブロック160eに含まれる別の階層の画像を用いて表示が可能領域は、特定のタイル画像へのポインタを設定せずにヌルポインタとする。図13では、インデックスブロック160eにおいて網掛けで示したタイル領域には特定のタイル画像へのポインタ、すなわち有効なポインタが設定され、白抜きの領域にはヌルポインタを設定するとする。

【0084】

同図において画像領域158を表示する場合、まずヘッダ150bにおいて当該画像領域158を含む領域152hを特定し、当該領域152hに対して設定されたポインタが指すインデックスブロック160eを取得する(矢印I)。インデックスブロック160eがデータを保持する領域のうち、画像領域158に対応する、第2階層162gの画像中の領域に対しヌルポインタが設定されていた場合、同じインデックスブロック160eに属する階層を縮小方向に辿っていき、同じ画像領域158の縮小画像に対し有効なポインタが設定されている階層を探索する。

【0085】

同図の例では、まず第1階層162hの画像に到達するが(矢印J)、画像領域158に対応する領域に対してはやはりヌルポインタが設定されている。そこでさらに解像度の低い第0階層162iの画像に到達する(矢印K)。第0階層162iの画像では、画像領域158に対応する領域に有効なポインタが設定されているため、当該ポインタが指すタイル画像170dのデータを取得し(矢印L)、それを用いて画像領域158の表示画像を生成することができる。ここで画像領域158がタイル画像のサイズを有するすると、画像領域158を描画するには同図に示すように、縮小画像であるタイル画像170dの一部の領域174を拡大する。上述と同様、画像領域158に対応する領域174の位置は、両者の縮尺率によって容易に計算できる。

【0086】

このように、インデックスブロック160eにおいてヌルポインタを定義可能とすることにより、インデックスブロック160eで定義される全てのタイル領域に対しタイル画像へのポインタを定義せずに済み、タイル画像のデータ圧縮とともにインデックスブロック自体のデータ量を削減することができる。また、タイル画像が更新されても、インデックスブロック内での画像の拡大縮小関係は基本的には変化しないため、全てのタイル画像へのポインタを更新する必要がなくなり、更新処理を最小限に抑えることができる。

【0087】

ヘッダと同様、インデックスブロックにおいても、解像度が最も小さい第0階層162iの画像では全タイル領域に対し有効なポインタを設定するようにする。このようにすることでその他の任意の領域でヌルポインタを設定しても、縮小方向へ階層を辿ることにより少なくとも第0階層162iの画像には到達し、表示画像を生成できる。ただしヘッダで定義される領域のうち、階層構造体の頂点、すなわち解像度が最も小さい階層を含む領域は上述のとおり、タイル画像の数が元々少ない場合があり、このような場合は対応するインデックスブロックの全ての階層にヌルポインタが設定された領域が含まれてよい。

【0088】

またヘッダと同様、インデックスブロックにおいても、ある画像領域について、高解像度の第2階層162gと低解像度の第0階層162iには特定のポインタを設定し、その中間の解像度の第1階層162hのみヌルポインタを設定するようにできる。例えば図13において中間の第1階層162hの左下の領域を表示する際は、解像度の低い第0階層162iの同領域に対して設定されたポインタが指すタイル画像を拡大して表示し、第2階層162gを用いるまで表示画像の解像度が上がったなら、当該第2階層162gの同じ領域に対して設定されたポインタが指すタイル画像を用いて表示画像を生成する。このようにすることで、ヘッダと同様、同じ領域でも拡大すると別の画像が表示されるような態

10

20

30

40

50

様を実現できる。

【 0 0 8 9 】

このようにインデックスブロックでタイル画像の共有を定義する場合、ヘッダでインデックスブロックの共有を定義する場合に比べ、タイル画像単位で詳細な設定を行うことができる。なお図 1 2 および図 1 3 の例では、ヘッダまたはインデックスブロックのいずれかにヌルポインタが設定されている場合を示しているが、双方にヌルポインタが設定されていても同様の処理により、インデックスブロックおよびタイル画像の共有を定義できる。また上述の例では、ヘッダまたはインデックスブロックにヌルポインタを設定することにより、異なる階層同士でタイル画像を共有する態様を実現した。一方、同一階層でタイル画像を共有する場合は、階層構造体の領域のうち、同一階層を含む複数の領域に対し同一のインデックスブロックへのポインタを設定するか、インデックスブロックにおいて複数のタイル領域に対し同一のタイル画像へのポインタを設定すればよい。

10

【 0 0 9 0 】

次に、画像ファイルを生成する機能を有する制御部 1 0 0 b の動作を説明する。図 1 4 は図 9 で示した制御部 1 0 0 b が画像ファイルを生成する処理手順を示すフローチャートである。まずユーザまたは別の処理モジュールによって画像データを指定する入力となされたら、画像階層生成部 1 2 0 は、ハードディスクドライブ 5 0 から指定された画像データを読み出す (S 5 0)。そして一般的な縮小処理を施すことによって所定の解像度を有する画像データを生成して、複数の解像度の画像データからなる階層データを生成する (S 5 2)。次に画像分割部 1 2 2 は、各階層の画像をタイル画像に分割したうえメインメモリ 6 0 に格納する (S 5 4)。このとき、各タイル画像の画像上の位置とメインメモリ 6 0 の記憶領域を示すアドレスとを対応づけて記録しておくことにより、インデックスブロックの生成に利用する。

20

【 0 0 9 1 】

なお S 5 0 において、あらかじめタイル画像を記録したファイルを読み出すようにしてもよい。この場合、S 5 2 の階層データ生成処理、S 5 4 の画像分割処理は省略できる。ファイルの先頭アドレスからの各タイル画像のアドレスのオフセット値と画像上の位置を当該ファイルに含めておけば、S 5 0 で読み出した際の当該ファイルのメインメモリ 6 0 における先頭アドレスを S 5 4 で記録しておくことにより、各タイル画像のアドレスと画像上の位置との関係を取得できる。

30

【 0 0 9 2 】

次に冗長性検出部 1 2 4 は、上述のように冗長性の有無を確認する (S 5 8)。具体的には、N 階層の画像を走査して、N + 1 階層の画像のタイル画像の大きさに対応する大きさの領域に単色で構成される領域があるか否かを確認する。これを解像度が高い階層の方向へ繰り返していく。冗長性検出部 1 2 4 はそのほか、N 階層の画像を N + 1 階層の画像の大きさまで拡大した画像と N + 1 階層の画像との差分画像や、N + 1 階層の画像の周波数解析により、N + 1 階層の画像であっても N 階層の画像と同程度の情報しか含まない領域があるか否かを確認する。確認手法は画像処理の分野で一般的に行われている手法を適宜適用してよい。

【 0 0 9 3 】

次にヘッダ・インデックスブロック生成部 1 2 6 は、冗長性を加味してヘッダおよびインデックスブロックを作成する (S 6 0)。具体的には、ヘッダで定義する領域ごと、低解像度側の領域に属する画像を拡大して表示できる場合は、当該領域に対してヌルポインタを設定し、その他の領域については対応するインデックスブロックを生成してそれぞれを指すポインタをヘッダに設定する。さらに各インデックスブロックにおいて、それに属する複数の階層の画像うち低解像度の画像を拡大して表示できる領域に対してはヌルポインタを設定し、それ以外の領域に対しては、当該領域から切り出したタイル画像へのポインタを設定する。

40

【 0 0 9 4 】

次に画像ファイル生成部 1 2 8 は、最終的に出力するための画像ファイルを作成する (

50

S 6 2)。ここで冗長性検出部 1 2 4 が S 5 8 で冗長性を検出しなかった場合は、S 5 4 で分割して生成されたタイル画像のデータを全てメインメモリ 6 0 より読み出し、S 6 0 で生成したヘッダおよびインデックスブロックとともに画像ファイルとする。冗長性が検出された場合は、S 6 0 で生成したインデックスブロックで設定されたポインタが指すタイル画像のデータのみをメインメモリ 6 0 より読み出し、S 6 0 で生成したヘッダおよびインデックスブロックとともに画像ファイルとする。

【 0 0 9 5 】

次に、このようにして生成した画像ファイルをもちいて画像を表示するときの動作を説明する。図 1 5 は、本実施の形態における画像ファイルを用いて画像を表示するための処理手順を示すフローチャートである。本処理手順は、図 7 に示した制御部 1 0 0 a によって実現できる。同図のフローチャートは、表示領域変更要求によって画像ファイルから必要なデータを読み出しデコードして表示する処理を主に記載しているが、ハードディスクからのロード、必要なデータの先読み処理、バッファメモリへの書き出しなど基本的な流れは図 5 で示したのと同様である。

10

【 0 0 9 6 】

まず表示装置 1 2 に画像の一部が表示された状態で、ユーザが入力装置 2 0 を介して表示領域の変更要求を入力すると、入力情報取得部 1 0 2 はそれを受け付ける (S 7 0)。するとタイル画像特定部 1 1 0 は、要求される表示画像の変更量を導出し、それに基づき新たに表示すべきフレーム座標を決定する (S 7 2)。表示画像の変更量は、仮想空間における上下左右方向の移動量および深さ方向の移動量であり、表示すべきフレーム座標は、それまでに表示されていた表示領域のフレーム座標と導出した変更量により決定できる。

20

【 0 0 9 7 】

次にタイル画像特定部 1 1 0 は、ヘッダを参照してフレーム座標が属する領域を特定し、当該領域に対しインデックスブロックへの有効なポインタ設定されているか否かを確認する (S 7 4)。有効なポインタが設定されていたら (S 7 4 の Y)、当該ポインタが指すインデックスブロックを取得する (S 7 8)。ヌルポインタが設定されていたら (S 7 4 の N)、階層構造体を画像の縮小方向へ辿り、有効なポインタが設定されている領域を探索し、検出した領域に対して設定されているポインタが指すインデックスブロックを取得する (S 7 6、S 7 8)。

30

【 0 0 9 8 】

さらにインデックスブロックにおいて、フレーム座標に対応する領域に対しタイル画像への有効なポインタが設定されているか否かを確認する (S 8 0)。有効なポインタが設定されていたら (S 8 0 の Y)、当該ポインタが指すタイル画像のアドレスや識別番号などタイル画像を一意に定める情報を特定する (S 8 4)。ヌルポインタが設定されていたら (S 8 0 の N)、当該インデックスブロックにおける階層を画像の縮小方向へ辿り、有効なポインタが設定されている階層を探索し、検出した階層の該当領域に対して設定されているポインタが指すタイル画像を特定する (S 8 2、S 8 4)。

【 0 0 9 9 】

バッファメモリ 7 0 に、当該タイル画像のデコードされたデータが格納されていない場合は (S 8 6 の N)、デコード部 1 1 2 がタイル画像のデータをメインメモリ 6 0 から読み出し、デコードする (S 8 8、S 9 0)。

40

【 0 1 0 0 】

S 7 6 でヘッダにおいて有効なポインタが設定されている領域を探索した場合、または、S 8 2 でインデックスブロック内で有効なポインタが設定されている階層を探索した場合は、特定したタイル画像を拡大して描画する必要がある。このため、そのような経緯を辿ってタイル画像を特定した場合、タイル画像特定部 1 1 0 はその旨の情報をデコード部 1 1 2 に与えておく。デコード部 1 1 2 は、その情報に基づきタイル画像の拡大が必要か否かを判定する (S 9 2)。

【 0 1 0 1 】

50

拡大が必要な場合（S92のY）、デコード部112はタイル画像特定部110から取得したフレーム座標に基づきタイル画像を拡大したうえ、必要な領域をバッファメモリ70に格納する（S94）。この処理は、該当タイル画像がデコード済み（S86のY）であっても同様に行う。そして拡大処理の有無に関わらず、表示画像処理部114は、当該タイル画像のうち新たに表示すべき領域をフレームメモリに描画する（S96）。描画処理には、要求される解像度に応じて、バッファメモリ70に格納されたデータを拡大、縮小する処理も含まれる。

【0102】

次にヘッダおよびインデックスブロックにおけるポインタのデータ構造について説明する。例えば11階層で構成される階層データを図8で示したように4×4、8×8、16×16の3階層の領域ずつインデックスブロックにまとめた場合、ヘッダで定義される領域は4162個になる。また一つのインデックスブロックには最大336のタイル領域が定義される。これらの要素のそれぞれにポインタを設定するため、ポインタの検索を効率的に行えるデータ構造とすることが望ましい。ポインタのデータ構造としては、固定長配列、連想配列、ツリー構造が考えられる。

10

【0103】

ヘッダのポインタを固定長配列とした場合は、領域ごとに識別番号を付与し、それをインデックスとして固定長のポインタ配列にアクセスする。この場合、領域を特定できればすぐにポインタが取得できるため、編集や検索処理が速く行える。連想配列とした場合は、領域ごとに識別番号を付与し、それをキーとしてポインタの連想配列にアクセスする。この場合、有効なポインタのみを保持すればよいため、ヘッダのデータサイズを抑えることができる。

20

【0104】

ツリー構造とした場合は、領域間をつなぐポインタをさらに定義し、画像の階層構造体の頂点から領域間ポインタをたどって所望の領域を検索する。この場合、後述するように頂点に階層を追加した際、頂点付近のポインタを更新するのみでよくなる。このようにポインタのデータ構造によって有効性が異なるため、メモリコストや処理コストなど重視すべき点によって適宜選択する。さらにBツリー構造などデータ探索で一般的に用いられる手法を適宜導入してもよい。インデックスブロックのポインタも同様の態様となるため、処理速度などを考慮して最適な手法を選択してよい。

30

【0105】

またインデックスブロックへのポインタ、およびタイル画像へのポインタは、目的とするインデックスブロックやタイル画像が記録されたファイル名やネットワークに接続したサーバのロケーション情報などを含めてもよい。これにより、1つの階層データに複数のファイルやサイトからの画像を含めたり、1つのファイルを複数の階層データで共有したりすることができる。

【0106】

次に、上述のようにヘッダ、インデックスブロック、タイル画像で構成される画像ファイルを修正、改変する場合について説明する。この態様も図1で示した情報処理システム、図4で示した情報処理装置10と同様の装置構成で実現できる。図16は、本実施の形態において画像を修正する機能を有する制御部100cの構成を示している。なお制御部100cは図7の制御部100aに示した、画像表示を行うための機能や、図9の制御部100bに示した、画像ファイルを生成する機能を備えていてもよいが、ここでは図示を省略している。一方、図16に示した機能のみを有する制御部100cを備えた情報処理装置10を、表示機能や画像ファイル生成機能を発揮する画像処理装置とは別に設けてもよい。

40

【0107】

制御部100cは、修正対象の画像ファイルを取得する画像ファイル取得部318、修正によって更新される領域や更新部分の画像データなどからなる更新情報を取得する更新情報取得部320、更新後のタイル画像を生成するタイル画像生成部322、ヘッダおよ

50

びインデックスブロックを更新するヘッダ・インデックスブロック更新部 3 2 4、修正中の画像を表示する表示画像制御部 3 2 6、および修正後の画像ファイルを生成する画像ファイル生成部 3 2 8 を含む。

【 0 1 0 8 】

画像ファイル取得部 3 1 8 は、ユーザによる修正対象の画像の指定入力を受け付け、該当する画像ファイルをハードディスクドライブ 5 0 から読み出してメインメモリ 6 0 に格納する。この画像ファイルはこれまで説明したように、ヘッダ、インデックスブロック、タイル画像のデータを含む。更新情報取得部 3 2 0 は、ユーザが表示装置 1 2 に表示された修正対象の画像を見ながら、入力装置 2 0 を介して入力した更新情報を取得する。更新情報には上述のとおり、更新すべき領域とその領域における修正後の画像データなどが含まれる。具体例は後に述べる。

10

【 0 1 0 9 】

タイル画像生成部 3 2 2 は、更新すべき領域に更新後の画像を当てはめた際に更新する必要が生じたタイル画像を特定し、新たなタイル画像を生成する。ここで、更新すべき領域があるタイル画像の一部にのみかかっている場合は、元のタイル画像をメインメモリ 6 0 から読み出し、更新すべき領域のみ上書きすることにより新たなタイル画像を生成する。あるタイル画像が全て更新すべき領域に含まれている場合は、修正後の画像を切り出すことによって新たなタイル画像を生成する。このとき、階層データを構成する全ての階層における更新すべき領域に対し新たなタイル画像を生成する。ただしユーザの指定によっては、特定の階層のみ更新するようにしてもよい。この場合、特定の解像度の範囲で異なる画像が表示される態様を実現できる。

20

【 0 1 1 0 】

生成したタイル画像はメインメモリ 6 0 に格納する。この際、画像ファイル取得部 3 1 8 がメインメモリ 6 0 に格納した元の画像データはそのままにしておき、新たに生成したタイル画像は別の記憶領域に格納する。次にヘッダ・インデックスブロック更新部 3 2 4 は、メインメモリ 6 0 に格納されたインデックスブロックにおいて、更新すべき領域に対して設定されたポイントを、新たに生成したタイル画像を指すように書き換える。また、当該領域に対応するインデックスブロックが存在しない場合は、新たにインデックスブロックを生成し、ヘッダにおけるヌルポイントの設定を、生成したインデックスブロックを指すポイントに書き換える。

30

【 0 1 1 1 】

表示画像制御部 3 2 6 は、図 7 で示した制御部 1 0 0 a に含まれる機能ブロックで構成してよいが、ここでは図示を省略している。表示画像制御部 3 2 6 は、図 1 5 で説明したのと同様の処理手順により修正対象の画像を表示するが、ユーザによる画像の修正に応じて上述のようにヘッダ、インデックスブロックが更新されていく。それに伴って、画像を描画する際に用いるタイル画像が新たなものに変化する。このようにすることで、表示画像を確認しながら画像を修正、変更することが可能になる。

【 0 1 1 2 】

本実施の形態では、元のタイル画像のデータを更新せず別の記憶領域に新たに生成したタイル画像を追加し、インデックスブロックやヘッダによって参照先を変化させることにより画像を更新する。このようにすると、元のタイル画像のうち修正領域にあたるタイル画像のデータを検索して上書きする場合と比較して処理コストが低く、応答性よく修正途中の画像を表示させることができる。また、修正途中で元の画像に戻すことも容易にできる。

40

【 0 1 1 3 】

画像ファイル生成部 3 2 8 は、ユーザが修正を終えたときなどに、新たに生成したタイル画像のデータで元のタイル画像のデータを上書きする。あるいは新たに生成したタイル画像のデータを、元のタイル画像のデータとは別のファイルとして保存する。そして更新されたインデックスブロックやヘッダを付加して画像ファイルとする。

【 0 1 1 4 】

50

次に、上述の機構によって画像を修正、改変するときの各データに対する処理の具体例を説明する。図17は説明の前提とする画像の修正例を示している。同図において修正前の画像190には、楕円と三角形が描かれている。これを修正し、修正後の画像192に示されるように、画像の右上に星の図形194を追加する場合を考える。図18はこのような修正におけるインデックスブロックのポインタの変化を模式的に示している。同図においてポインタは矢印で示しているが、図が煩雑になるのを避けるため、代表的な矢印のみを表している。

【0115】

まず修正前において、ヘッダ150で定義される各領域は、インデックスブロック160のいずれかへのポインタ、あるいはヌルポインタが設定されている。そしてポインタが指すインデックスブロック160に属する複数の階層の各タイル領域に対し、破線の矢印によって示されるように、修正前に生成されていたタイル画像170eのいずれかへのポインタが設定されている。図17における修正前の画像190の右上の領域は単色の背景であるとする、当該領域は一つのタイル画像を共有する。図18において多くの破線矢印が一つのタイル画像170fを指しているのはこれを表している。当然、その他の領域は個々のタイル画像へのポインタが設定されている。

【0116】

このような背景部分に、図17における修正後の画像192のように星の図形194を追加すると、各階層の該当領域におけるタイル画像として、各解像度の星の図形が追加された新たなタイル画像170gが生成され、元のタイル画像170eとは別の記憶領域に格納される。するとインデックスブロック160において、更新すべき領域にかかるタイル領域に対するポインタは全て、新たなタイル画像170gのいずれかのタイル画像を指すように更新される。図18では、更新後のポインタを一点鎖線の矢印で表している。更新後、更新対象の領域は背景でなくなったため、図に示すように、各タイル領域に対しそれぞれのタイル画像へのポインタが設定される。また、修正対象の領域以外の領域は、元のタイル画像170eを指したままでよい。

【0117】

次に、図17の星の図形194のように更新によって追加された画像が、それまでの階層データの最高解像度の画像よりさらに高い解像度を有する場合に、高解像度側に階層を追加する手順を説明する。図19は、高解像度側に階層を追加する場合の処理を説明するための図である。同図において、修正前の画像は第0階層154i、第1階層154j、第2階層154k、第3階層154lからなる階層データ200であったとする。ここでヘッダにおいては全ての領域に対しインデックスブロックへの有効なポインタが設定されているものとし、同図ではそれを網掛けで示している。

【0118】

このような画像に対し、図17で示すように右上の領域に高解像度の画像を追加する修正を行った場合、上述のように更新対象の領域のタイル画像を各階層について新たに生成する。ここで、追加した画像は元の階層データで最高解像度を有する第3階層154lより高解像度であるため、新たな階層を追加する必要がある。同図では追加する階層を第4階層154mとしている。また新たにタイル画像を生成すべき領域は、同図において黒く示されている。

【0119】

この例では高解像度画像による画像の修正は局所的なものであるため、更新対象の領域以外の領域を第4階層154mの解像度で表示する場合は、元からある階層データのうち最高解像度を有する第3階層154lの画像を拡大して表示することになる。そのため、第4階層154mのうち更新対象の領域以外に対しては、ヌルポインタを設定し、第3階層154lのタイル画像を流用する。ヌルポインタを設定する領域は、同図では白抜きで表されている。

【0120】

タイル画像の共有には前述のとおり、ヘッダにおいて設定する手法とインデックスブ

10

20

30

40

50

ックにおいて設定する手法がある。一つのインデックスブロックでまとめられる階層数を固定とした場合、追加すべき第4階層154mに対応する階層が、既存のインデックスブロックに既に定義づけられている場合がある。第4階層154mは元の階層データでは存在していないため、インデックスブロックの同階層に対してはヌルポイントが設定されている。

【0121】

このような場合は、当該既存のインデックスブロックにおいて、第4階層154mの更新対象の領域に対して設定されているヌルポイントを、新たに生成したタイル画像へのポイントに更新する。更新対象の領域以外の領域は、ヌルポイントのままとする。低解像度の各階層についても同様に、インデックスブロックのポイントを更新する。

10

【0122】

図20は、追加すべき第4階層154mに対応する階層が属するインデックスブロックが存在しなかった場合のヘッダおよびインデックスブロックの変化を説明する図である。同図のヘッダ150において、領域152iは修正前の階層データに対し定義づけられている領域を示している。ヘッダ150で定義される領域は当然複数でよいが、同図ではそれらをまとめて一つの領域152iで示している。領域152iには、修正前に存在した第0階層154i、第1階層154j、第2階層154k、第3階層154lが属している。そして各領域に対し、インデックスブロック160fへのポイントが設定されている。そしてインデックスブロック160fで定義づけられる各タイル領域に対し、既存のタイル画像170hへのポイントが設定されている(破線矢印)。

20

【0123】

ここで第4階層154mを追加する場合、まずヘッダ150において、当該階層を含む新たな領域を追加する。ただし図20に示すように、更新対象の領域を含む領域152jのみ有効なポイントを設定し、その他の領域はヌルポイントを設定する。さらに新たなインデックスブロック160gを生成し、領域152jに対して当該インデックスブロック160gへのポイントを設定する。追加する階層が第4階層154mのみであれば、新たに生成したインデックスブロック160gに属する3階層のうち、タイル画像へのポイントを設定するのは最も低解像度の一階層でよく、その他の階層は全領域に対しヌルポイントを設定する。

【0124】

そして、新たに生成したインデックスブロック160gにおいて、更新対象の領域に含まれるタイル領域に対し、新たに生成したタイル画像170iのいずれかを指すようにポイントを設定する(一点鎖線矢印)。また既存のインデックスブロック160fも同様に、更新対象の領域に含まれるタイル領域に対し設定されたポイントが、新たに生成したタイル画像170iのいずれかを指すように更新する。なお最終的に画像ファイルを生成する際は、新たに生成したインデックスブロック160gは、元の画像ファイルの末尾に追加してもよいし、別のファイルとしてもよい。これにより更新前後の画像を表示することができる。

30

【0125】

次に、既存の画像に背景や描画領域など新たな領域を追加することにより画像のサイズを拡張する手順を説明する。図21は、既存の画像に新たな領域を追加する処理を説明するための図である。同図において、追加前の画像は第0階層154i、第1階層154j、第2階層154k、第3階層154lからなる階層データ204であったとする。各階層が属する領域は、ヘッダにおいてインデックスブロックへの有効なポイントが設定されている。

40

【0126】

このような画像に対し新たな領域を追加する場合、各解像度に対応するサイズで各階層の画像に新たな領域を追加した階層データ206を生成する。図21の例で階層データ206は、第0階層154n、第1階層154o、第2階層154p、第3階層154q、第4階層154rからなる。第4階層154rおよび第3階層154qは、白抜きで示し

50

た元の画像のタイル画像に、網掛けで示した追加領域のタイル画像を加えることによって生成できる。第2階層154pはそのサイズによって、元のタイル画像がそのまま利用できる白抜きで示した領域に、追加領域と元の画像が混在する網掛けで示したタイル画像を加える。第1階層154oおよび第0階層154nは、それを構成する全てのタイル画像において、元の画像と追加領域が混在するため全て網掛けで示している。

【0127】

このように、画像に新たな領域を追加する場合、階層によっては元からあるタイル画像をそのまま利用できる。この性質を利用し、元からあるインデックスブロックはそのまま利用し、追加領域など必要な領域についてのみ新たなインデックスブロックを生成する。図22は、新たな領域を追加する場合のヘッダおよびインデックスブロックの変化を説明する図である。同図のヘッダ150において、白抜きの三角形は元のタイル画像を利用できる領域208を示し、網掛けの部分は追加または追加によって変化する領域210を示している。

10

【0128】

破線は図21の第0階層154n、第1階層154o、第2階層154p、第3階層154q、第4階層154rを示しており、それぞれの階層において元のタイル画像を利用できる領域、および追加または追加によって更新すべき領域が決定され、図22に示すような構造となる。ヘッダで定義される領域は、階層構造体の頂点から順に生成されるため、このようにヘッダ150が変化すると領域の区切りに変化が生じる。

【0129】

20

しかし後に述べるように領域の区切りを工夫することにより、図22のように階層データの頂点で階層が増加するような変化が生じても、変化前に定義されていた領域は可能な限り保存し、ひいては既存のインデックスブロックをそのまま利用できるようにする。図22で領域追加前は、領域208の各領域（図示せず）に対し、インデックスブロック160hへのポインタが設定されている。そしてインデックスブロック160hで定義づけられる各タイル領域は既存のタイル画像170hへのポインタが設定されている（破線矢印）。

【0130】

領域追加後は、追加する領域に対応する新たなインデックスブロック160iを生成し、ヘッダ150における新たな領域210に対し当該インデックスブロック160iへのポインタを設定する。ここで領域210はヘッダ150においては複数の領域が定義されていてよいが図示を省略している。生成したインデックスブロック160iには、タイル領域ごとに、領域追加によって必要となった新たなタイル画像170kのいずれかへのポインタが設定される（一点鎖線矢印）。ここで新たなタイル画像170kは、図21で網掛けで示した、追加する領域自体のタイル画像や、追加する領域と元の画像が混在するタイル画像である。この際、追加する領域が単色の背景である場合などは、一つのタイル画像を共有できる。

30

【0131】

一方、ヘッダにおいて元から定義づけられていた領域は、領域の区切り方に変更がない場合は、設定されたポインタ、ひいてはそれが指し示すインデックスブロックをそのまま利用できる。そのためインデックスブロック内で設定される、タイル画像へのポインタを更新する必要がない。なお上述の例は画像の右および下の領域を追加する場合であるが、いずれの方向に画像を拡張する場合であっても、既存のインデックスブロックを拡張後も利用できるように領域分割を行うことにより、ヘッダ、インデックスブロックの変化を最小限に抑えることができる。

40

【0132】

図23は、図21および図22で説明したように、画像サイズを拡張する場合など階層構造体の頂点側に階層を追加する際にヘッダにおいて定義される領域の分割手法を説明する図である。まず最初の階層データ230では、第0～1階層の全体を第1領域152k、第2～4階層の全体を第2領域152l、第5～7階層を画像上の領域で分割した領域

50

を順に第3領域152m、第4領域152n、・・・、というように領域分割される。この領域の構成は図8で示したのと同様である。

【0133】

前述のとおりひとつのインデックスブロックが定義するタイル領域の数はインデックスブロックによらず等しく準備するが、第1領域152kは、1つのタイル画像で構成される第0階層、2×2のタイル画像で構成される第1階層のみからなるため、当該領域に対応するインデックスブロック160jにおいてはその数のタイル領域のみを使用し、それ以外のタイル領域にはヌルポインタを設定して無効とする。

【0134】

なお図23に示すインデックスブロックでは、有効なタイル領域の横方向の数を階層別に矩形で示している。従って第1領域152kに対応するインデックスブロック160jは1つおよび2つの矩形が示されている。第2領域152l、第3領域152mなどその他の領域に対応するインデックスブロック160k、160lなどは、4×4個、8×8個、16×16個のタイル領域を全て有効にできるため、4個、8個、16個の矩形が示されている。

【0135】

ここで階層データ230の頂点側に1階層、追加した階層データ232に対しては、元から第1領域に属していた階層はそのままに、追加した階層を第1領域152oにさらに含める。元の第1領域152kに対応するインデックスブロック160jは、データ領域として準備された3階層のうち2階層分のみ用いていたため、無効としていた階層を利用することにより階層の追加を第1領域152oのみで吸収することができる。したがって、元の階層データ230において第2領域152lに含まれていた第2～4階層(階層追加後の第3～5階層)は、そのまま同じ領域に属することができる。ただし各階層の画像は拡張されているため、拡張部分に対し新たな領域を定義しインデックスブロックを生成する。さらに下の階層も同様である。

【0136】

ここで階層データ232の頂点側にさらに1階層、追加した階層データ234に対しても、元から第1領域に属していた階層はそのままに、追加した階層を第1領域152pにさらに含める。この場合、第1領域152pのみ4階層となるが、それに対応するインデックスブロック160mで準備されているタイル領域は、第1領域152pのタイル画像の数より大きいため、4階層であってもタイル画像の情報はインデックスブロック160mで充足する。すると、それ以下の階層は最初の階層データ230のときに属していたのと同様の領域に属することができる。

【0137】

階層データ234にさらに1階層、追加した階層データ236では、階層データ234で第1領域152pに属していた4階層に、追加した階層を加えた5階層を2つの領域に分割して第1領域152q、第2領域152rとする。このとき第1領域152qを第0～1階層、第2領域152rを第2～4階層とすることにより、第1領域152qに対応するインデックスブロック160nは、最初の階層データ230の第1領域152kに対応するインデックスブロック160jと同様、1つのタイル画像の階層、2×2個のタイル画像の階層のみからなる。また第2領域152rも階層データ230の第2領域152lと同様となる。

【0138】

そしてそれ以下の階層は、最初の階層データ230のときに属していたのと同様の領域に属することができる。以上の変化を繰り返すことにより、階層データの頂点に階層を追加しても、可能な限りその他の領域に変化なく領域を定義することができ、ひいては既存のインデックスブロックをそのまま利用することができる。

【0139】

図21で示したように画像の領域を拡張した場合、画像処理で一般的に用いられるようなラスタ方向の順番に基づくタイル画像の管理では、追加した領域のタイル画像が既存の

10

20

30

40

50

タイル画像の間に挿入されてしまうため、タイル画像の並び替えや識別番号のふり直しが必要となる。タイル画像の数が増加するほど、この処理にかかる負荷は多大なものになる。本実施の形態では上述のとおり、局所的なデータの変化のみで画像の領域の拡張が可能になるため、処理コストが低く、更新中のデータ表示や最終的な画像ファイル生成を応答性よく実現できる。

【 0 1 4 0 】

次に上記の構成によって実現できる動作を説明する。図 2 4 は画像処理装置においてユーザが画像を修正、改変する際の処理手順を示すフローチャートである。まずユーザが修正対象の画像ファイルを指定すると、画像ファイル取得部 3 1 8 はハードディスクドライブ 5 0 から該当する画像ファイルを読み出しメインメモリ 6 0 に格納する (S 1 1 0)。すると表示画像制御部 3 2 6 は当該画像ファイルの画像を表示装置 1 2 に表示する (S 1 1 2)。

10

【 0 1 4 1 】

この状態でユーザが表示装置 1 2 に表示された画像を見ながら更新情報を入力すると (S 1 1 4 の Y)、更新情報取得部 3 2 0 はそれを受け付け、タイル画像生成部 3 2 2 が更新部分の新たなタイル画像を生成してメインメモリ 6 0 に格納する (S 1 1 6)。ユーザが行う更新情報の入力、例えば表示装置 1 2 に表示した画像の近傍に、新たに貼り付けたい画像をさらに表示し、ユーザがポインティングデバイスで所望の領域までドラッグさせるなどの操作によってなされる。あるいはポインティングデバイスによって文字や絵を直接描き込んだり、領域を追加するコマンドを入力するなどでもよい。

20

【 0 1 4 2 】

次にヘッダ・インデックスブロック更新部 3 2 4 は、上述のように画像の更新内容によって適宜ヘッダ、インデックスブロックを更新する (S 1 1 8)。すると表示画像制御部 3 2 6 は、新たに参照先となったタイル画像のデータを読み出し、デコードすることによって表示装置 1 2 の表示を更新する (S 1 2 0)。S 1 1 4 から S 1 2 0 までの処理を、修正が終了し更新情報が得られなくなるまで繰り返す。

【 0 1 4 3 】

修正が終了したことをユーザからの入力などにより検知したら (S 1 1 4 の N)、画像ファイル生成部 3 2 8 は、メインメモリ 6 0 に格納しておいた、新たに作成したタイル画像を、元の画像ファイルの画像データに組み込んで、画像データを再構築する (S 1 2 2)。あるいは新たに作成したタイル画像のみを別のファイルに保存する。そして S 1 1 8 で更新した最終的なヘッダおよびインデックスブロックのデータとともに画像ファイルとして出力する (S 1 2 4)。

30

【 0 1 4 4 】

修正中、ヘッダ、インデックスブロック、タイル画像は、既存のデータと別の記憶領域に格納しておく。そしてユーザが、直前の画像の更新を取り消す操作 (U n d o) を行った場合はそれらのデータの参照先を元に戻すことにより、画像を前の状態に戻すことが容易にできる。操作の再履行 (R e d o) でも同様に、参照先を変更するのみで遂行される。

【 0 1 4 5 】

以上述べた本実施の形態によれば、階層データをヘッダ、インデックスブロック、タイル画像、の 3 つのデータセットで構成する。これにより、タイル画像を本来の画像上の位置から独立させることができ、画像を修正した際などに管理が容易になるほか、画像の冗長性を考慮してタイル画像を複数の領域で共有することができるため、タイル画像のデータサイズを効率よく削減することができる。

40

【 0 1 4 6 】

また、タイル画像を複数の領域で共有する際、ヘッダもしくはインデックスブロックで設定するポインタを無効とし、無効となっている領域では、階層を低解像度側へ辿って有効なポインタが設定されている領域を探索する。これにより、全ての領域に対していずれかのタイル画像へのポインタを設定する場合と比較し、インデックスに用いるデータ自体の

50

サイズを削減できる。

【0147】

さらに、ヘッダによって階層構造体を領域分割し、領域ごとにインデックスブロックを対応づける。この際、一つの領域には複数の階層を含める。これにより、一時に表示や修正の対象となりやすいデータをまとめて扱うことができ、不必要なデータをメインメモリにロードするなど、記憶領域や処理の面で非効率な状況を回避しやすくなる。また、インデックスブロックで定義できるタイル領域の数を固定とすることで、データキャッシュ時のフラグメンテーションが発生せず、記憶領域の管理が容易である。また画像が巨大化してデータサイズが大きくなっても、階層データの領域分割により表示や修正対象のデータへのアクセスが効率的に行える。

10

【0148】

例えば80000個のタイル画像で構成される階層データの場合、インデックスブロックで定義するタイル領域の数を400とすると、インデックスブロックは200個生成される。ここで1つのタイル画像にアクセスしたいとき、単なる80000個のタイル画像の羅列であれば、それを順に検索していく必要があるが、本実施の形態のように領域分割されていれば、1つの領域を検索することは400個のタイル画像を検索することと同等となり、効率の良いアクセスが可能となる。

【0149】

同様に、一般的な画像表示装置で表示を行う場合、一度に表示する画像に必要なインデックスブロックは、平均的には1~2個であり、4個以上のインデックスブロックが必要となることは稀である。そのため、領域を分割せずにタイル画像へのインデックスを保持する場合と比較し、一度に必要なインデックスデータサイズが少なく済み、メインメモリを節約できる。結果的に、複数の階層データを同時に処理するような態様も、メインメモリを圧迫せずに実現できる可能性が高くなる。

20

【0150】

さらに、ヘッダおよびインデックスブロックで設定するポインタには、インデックスブロックおよびタイル画像を記録したファイルの設定もできるようにする。このようにすることで、一つの階層データであっても複数のインデックスブロックファイルやタイル画像ファイルから読み出したデータを利用することができ、階層データをより柔軟に生成、更新することができる。またファイルが異なれば同一の記憶媒体に格納しておく必要がなくなるため、例えば4GBを超える巨大な画像データを生成することができる。これにより非圧縮画像、HDR (High Dynamic Range) 画像、アトリビュート情報などを処理対象とすることも可能となる。

30

【0151】

同時に、複数の階層データで、用いるデータのファイルを共有することも可能となる。例えば雑誌のヘッダ、フッタなどテンプレートとして再利用が可能な画像は、個々に保持せず共有することで、階層データ個別のデータサイズを小さくできる。タイル画像ばかりでなくインデックスブロックのデータを共有してもよい。

【0152】

これを応用すると、階層データを用いたコンテンツのアップグレードが容易に実現できる。例えば通常版のコンテンツとしてヘッダ、インデックスブロック、タイル画像のデータを含むファイルを配布する。そして追加用として、さらに高解像度な階層のタイル画像のファイルを別に用意する。通常版のヘッダやインデックスブロックには、追加用のタイル画像へのポインタも記述しておくが、当該タイル画像を入手しない限り、そのポインタは無効となるため表示はされない。ユーザがコンテンツのアップグレードをすると、追加用のファイルがダウンロードされ、それに記録されたタイル画像へのポインタが有効となるため、追加用の画像が表示される。このようにすると、通常版のコンテンツファイルを変化させることなくアップグレードが容易に可能となる。アップグレードには課金を義務づけてもよい。

40

【0153】

50

実施の形態 2

実施の形態 1 では、主にユーザが入力した情報に基づき画像ファイルを生成、修正した。本実施の形態では、グラフィックプロセッサによって階層データへレンダリングする態様について述べる。レンダリングに階層データを導入することにより、実施の形態 1 で述べたのと同様、必要とされる解像度に見合った処理のみを行えばよくなり、高精細の巨大な画像であっても効率的な処理が可能となる。本実施の形態は実施の形態 1 の図 4 で示した情報処理装置 10 によって実現できる。図 25 は本実施の形態における情報処理装置 10 のうち制御部 100 およびメインメモリ 60 の構成をより詳細に示している。

【0154】

制御部 100 はメインプロセッサ 400 とグラフィックプロセッサ 402 を含む。メインプロセッサは情報処理装置 10 において起動したアプリケーションプログラムを主に実行するとともに、情報処理装置 10 に含まれる他の機能ブロックの制御を行う。グラフィックプロセッサ 402 はメインプロセッサ 400 からの画像処理要求に従い画像処理を実行する。画像処理に必要なデータはメインメモリ 60 に格納され、グラフィックプロセッサ 402 はそれを読み出し適宜画像処理を行ったうえ、メインメモリ 60 などに結果を書き出すことにより画像処理を遂行する。

10

【0155】

グラフィックプロセッサ 402 は、メインプロセッサ 400 からの画像処理要求に従い、メインメモリ 60 に格納されたモデルデータ 418 を用いてレンダリングを行い、同じくメインメモリ 60 に格納したカラーバッファ 414 および Z バッファ 416 を更新する。本実施の形態では図 25 に示すように、カラーバッファ 414 および Z バッファ 416 はいずれも階層データの構造とする。すなわち、描画対象の画像の画素ごとの色情報を保持するカラー値画素平面と、画素ごとの視点からの奥行き情報を保持する Z 値画素平面とを、描画対象画像の複数の解像度に対応させて階層化した構造を有する。当該階層データは、実施の形態 1 で説明したように、ヘッダ、インデックスブロック、タイルデータによって構成してよい。ここでタイルデータは、各階層の画素平面を所定のサイズに分割したタイル領域ごとのデータである。

20

【0156】

グラフィックプロセッサ 402 は、メインプロセッサ 400 からの要求に従いレンダリングを行うレンダリング部 404、レンダリング部 404 によるレンダリングの結果をメインメモリ 60 に格納されたカラーバッファ 414 および Z バッファ 416 の各階層に反映させる階層データ更新部 406 を含む。グラフィックプロセッサ 402 はさらに、レンダリング時に一時的に用いる記憶領域として、GPU カラーバッファ 408、GPU-Z バッファ 410、およびダーティマスク 412 を含む。ダーティマスク 412 は GPU カラーバッファ 408、GPU-Z バッファ 410 と同じサイズの領域を有し各ピクセルが 1 ビットの情報を保持する。レンダリングによって GPU カラーバッファ 408、GPU-Z バッファ 410 が更新されたら、更新対象のピクセルの値を変化させることにより、当該ピクセルの領域を検出する。

30

【0157】

次に上記の構成を有する画像処理装置の、レンダリング時の動作を説明する。図 26 は階層データへのレンダリングの処理手順を示すフローチャートである。まずメインプロセッサ 400 はグラフィックプロセッサ 402 に対しレンダリング要求を行う (S200)。レンダリングに必要なデータは上述の通りメインメモリ 60 に格納される。モデルデータ 418 の他、上述の通り階層構造を有するカラーバッファ 414 および Z バッファ 416 もこれに含まれる。

40

【0158】

するとグラフィックプロセッサ 402 のレンダリング部 404 は、モデルデータ 418 などに基づき、カラーバッファ 414 および Z バッファ 416 のうちレンダリング対象の階層および領域を特定する (S202)。次にレンダリング部 404 は、当該領域のデータをカラーバッファ 414 および Z バッファ 416 から読み出し、GPU カラーバッファ

50

408およびGPU-Zバッファ410からなるGPUバッファに敷き詰める(S204)。レンダリング対象の階層および領域からタイルデータを特定する手法は実施の形態1で述べたタイル画像の特定手法と同様に、ヘッダ、インデックスブロックの順で検索していく。したがって読み出しはタイルデータの単位で行う。

【0159】

このときレンダリング部404は、ダーティマスク412を初期化する(S206)。例えば各ピクセル値を「0」にセットする。そして、モデルデータ418を用いて一般的な手法でレンダリングを行い、GPUカラーバッファ408、GPU-Zバッファ410を適宜更新するとともに、ダーティマスク412における更新対象のピクセル値を「1」に更新する(S208)。

10

【0160】

そしてGPUカラーバッファ408、GPU-Zバッファ410におけるレンダリング結果に基づき、メインメモリ60のカラーバッファ414およびZバッファ416の階層データを更新する(S210)。具体的にはまず、各階層データのうち、レンダリング対象の階層および領域において、更新対象のピクセルを含むタイル領域をダーティマスク412のピクセル値に基づき特定し、当該タイル領域のデータを、GPUカラーバッファ408、GPU-Zバッファ410のレンダリング結果に基づき更新する。

【0161】

このとき実施の形態1で述べたように、新たなタイルデータを生成し、インデックスブロックに設定されているポインタを新たなタイルデータへのポインタに更新する。なお、頻繁に更新されることが予想される場合は、タイルデータの生成を省略してポインタの更新のみを行い、画像が確定してからタイルデータを生成してもよい。次にレンダリング対象の階層以外の階層に対しても、更新された領域に対応する領域のタイルデータを新たに生成し、インデックスブロックに設定されているポインタを新たなタイルデータへのポインタに更新する。この際、階層によって解像度が異なるため、レンダリング結果は拡大または縮小して新たなタイルデータに反映させる。

20

【0162】

既存の階層データより高解像度の画像をレンダリングするときなど、レンダリング対象の領域のタイルデータがカラーバッファ414およびZバッファ416に元々存在しない場合は、その上の階層の対応する領域を拡大してGPUカラーバッファ408、GPU-Zバッファ410に敷き詰める。そしてレンダリング後、更新されたタイルデータのみをカラーバッファ414およびZバッファ416に反映させ、ヘッダあるいはインデックスブロックのポインタを更新する。その後、レンダリング対象の階層以外の階層に対しても、タイルデータを生成、ポインタを更新する点は上記と同様である。この処理は、実施の形態1において図19を用いて説明したのと同様の処理となる。

30

【0163】

なおS210においてレンダリングの結果を階層データに反映させる際、レンダリング対象の階層より下の階層へ反映させるためにレンダリング結果を拡大した領域が、当該階層のタイル領域のいずれかを占めていたらそのタイルデータは削除し、それへのポインタを無効とする。これによりタイルデータのサイズを削減できる。表示時は実施の形態1で述べたのと同様、ポインタが有効な階層を探索し、上の階層の画像を拡大すればよい。

40

【0164】

なおカラーバッファ414およびZバッファ416は、ヘッダおよびインデックスブロックを個別に保持してもよいし、共有してもよい。共有とする場合、インデックスブロックは一つのタイル領域に対し複数のポインタ、すなわちカラーバッファのタイルデータおよびZバッファのタイルデータへのポインタを設定する。これにより、個別に保持するよりヘッダおよびインデックスブロックのデータを圧縮することができる。レンダリング後、画像を確定する場合は、インデックスブロックのうち不要となったZバッファのタイルデータへのポインタを削除してもよい。インデックスブロックにおいて一つのタイル領域に設定する複数のポインタが指す対象は、画像の内容によっては、カラーバッファやZバ

50

ッファに限られない。

【0165】

また複数のカラーバッファ間でZバッファを共有してもよい。例えばカラーバッファのデータを2つ、Zバッファのデータを1つ用意し、一方のカラーバッファとZバッファでレンダリングを行い、他方のカラーバッファとZバッファでレンダリングを行ってもよい。この場合はそれぞれ個別にヘッダおよびインデックスブロックを保持する方が制御の面で有効である。

【0166】

カラーバッファおよびZバッファを階層データとしてレンダリングする本実施の態様は、アンチエイリアスに応用することができる。すなわち高解像度の階層とそれより解像度が低い表示用の画像の階層とからなる階層データを生成する。一般的なスーパーサンプリングでは、単色の背景などオブジェクトが存在しない領域についても、高解像度画像用の記憶領域を用意する必要がある。一方、これらのバッファを上述のような階層データとすることにより、低解像度の階層を拡大すれば得られる領域は高解像度画像のレンダリング対象から外すことができる。

10

【0167】

例えば領域ごとに必要な解像度をユーザが指定することにより、各階層において必要最低限の領域のみのレンダリングで済みメモリの使用効率が向上する。頻繁なタイルデータ更新と画像表示を応答性良く行うためには、生成したタイルデータは圧縮しなくても良い。なお画像の全体にわたりアンチエイリアスが必要な場合はスーパーサンプリングを行うなど、対象領域の割合によってアンチエイリアスの手法を切り替えてもよい。

20

【0168】

アンチエイリアスを階層データを用いて行う場合、常に最下層の画像のみをレンダリングし、画像確定後にその縮小画像を生成する場合と、ユーザが指定した階層にレンダリングし、そのたびに全ての階層の対応する領域も更新する場合とが考えられる。前者は縮小処理の頻度が低く抑えられ、後者はラスタライズ、シェーダー処理、ROP (Rendering Output Pipeline)、ピクセル転送などのピクセル生成処理のコストを予測しやすい。ただし後者は、グラフィックプロセッサ402とメインメモリ60間で元の画像、拡大画像、縮小画像などのデータが頻繁に転送されるため、転送バンド幅への負荷がかかり易くなる。

30

【0169】

アンチエイリアスの場合のみならず、階層データをレンダリングする場合は、レンダリング結果を各階層へ反映させるため、画像を拡大、縮小するスケーリング処理が頻繁に必要となる。スケーリング処理の都度、画像データをグラフィックプロセッサ、メインメモリ間で送受すると、両者間の転送バンド幅を圧迫する懸念がある。そこで本実施の形態では、図27で示すようにスケーリング処理を行うスケーラ422をメインメモリ60に内蔵するようにしてもよい。

【0170】

これにより、グラフィックプロセッサ520およびメインメモリ60は、レンダリング対象の解像度の画像のみをやりとりすればよく、メインメモリ60内の階層データにおける各階層へのレンダリング結果の反映は、スケーラ422がメインメモリ60内で行う。結果としてグラフィックプロセッサ520とメインメモリ60間の転送バンド幅の圧迫が防止でき、階層データのレンダリングが効率的に実施できる。なおスケーラ422はメインメモリ60に内蔵せず、グラフィックプロセッサ520とメインメモリ60間のバスとは別のバスでメインメモリ60に接続するようにしても同様の効果が得られる。

40

【0171】

以上述べた本実施の形態は、一般的なグラフィックプロセッサを利用して実現することができる。ここで一般的なグラフィックプロセッサでは、内部のフレームバッファ、すなわち上述のGPUバッファのアドレッシング時、メインメモリからの読み出しおよび書き込みの回数を抑えるため、画面上でまとまった領域に対し近いメモリアドレスを割り当て

50

る、という手法をとる。しかしGPUバッファに用いられる領域は連続したアドレス空間でなくてはならない、という制約がある。すなわちアドレスが不連続な領域を敷き詰めることができない。図28、29はその際の処理の様子を示している。

【0172】

まず図28において、階層データの第3階層の画像の一部の領域430をレンダリング対象としたとき、「11」～「18」のアドレスが割り振られるとする。この場合、グラフィックプロセッサ402はGPUバッファに当該領域430を敷き詰めることができる。その後、図29に示すように、同じ第3階層の一部の領域432をレンダリング対象としたとする。領域432は、図28においてアドレス「11」および「12」を割り当て済みの領域を含む。この場合、領域432のその他の領域に、「19」～「24」のアドレスを割り振る。このようにアドレス割り当てを行うと、領域432はアドレスが不連続な領域を含んでしまい、グラフィックプロセッサ402がGPUバッファに当該領域を敷き詰めることができなくなる。

10

【0173】

このため、領域432を別の記憶領域に一旦コピーし、新たな連続したアドレス(図29ではアドレス「50」～「57」)を割り当てることによりGPUバッファへの敷き詰めを行う必要がある。このような処理コストを軽減するため本実施の形態では、メインメモリ60のメモリコントローラ(図示せず)に、領域の位置と割り振るアドレスとを対応づけるテーブルを設けてもよい。図29の領域432の場合、ラスト順に「0」～「7」で領域の位置を表現し、「0」=「19」、「1」=「20」、「2」=「21」、「3」=「22」、「4」=「23」、「5」=「24」、「6」=「11」、「7」=「12」、と対応づける。

20

【0174】

こうすることにより不連続なアドレスを割り振られた領域のレンダリングであっても、メインメモリ60においてアドレス変換がなされるため、図29で示したような敷き詰めのためのコピーが必要なくなる。さらに図27のようにメインメモリ60がスケーラ422を内蔵していたとすると、拡大、縮小した画像の生成も含め、階層データを生成、更新するための多くの処理をメインメモリ60内で閉じることができる。

【0175】

本実施の形態のように階層データのレンダリングを可能とすることにより、ヘッダ、インデックスブロックによるタイルデータへのランダムアクセス、ヘッダを分割して階層データごとのメモリコストを下げたことによる複数の階層データの同時処理、タイルデータの追加、削除、更新がレンダリングによって可能、といった特徴が得られる。これらの特徴は、ランダムアクセス、マルチテクスチャリング、レンダーターゲット、というテクスチャの要件と同等である。したがって、階層データをテクスチャと同等に扱うことが可能になり、階層データをコンピュータグラフィックスを用いた広範囲の用途に応用することができる。

30

【0176】

以上、本発明を実施の形態をもとに説明した。上記実施の形態は例示であり、それらの各構成要素や各処理プロセスの組合せにいろいろな変形例が可能なこと、またそうした変形例も本発明の範囲にあることは当業者に理解されるところである。

40

【符号の説明】

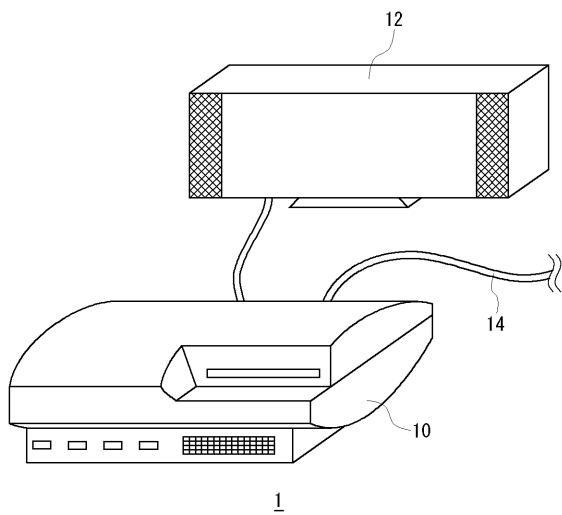
【0177】

1 情報処理システム、 10 情報処理装置、 12 表示装置、 20 入力装置、
38 タイル画像、 50 ハードディスクドライブ、 60 メインメモリ、 100 制御部、
102 入力情報取得部、 106 ロードブロック決定部、 108 ロード部、
110 タイル画像特定部、 112 デコード部、 114 表示画像処理部、
120 画像階層生成部、 122 画像分割部、 124 冗長性検出部、 126 ヘッダ・インデックスブロック生成部、
150 ヘッダ、 160 インデックスブロック、 170 タイル画像、 318 画像ファイル取得部、 320 更新情報取

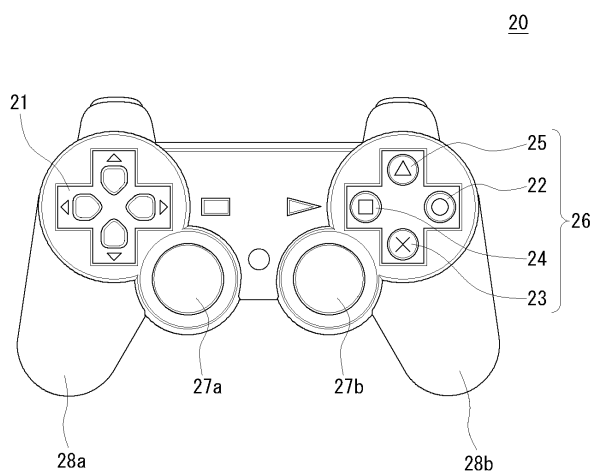
50

得部、 3 2 2 タイル画像生成部、 3 2 4 ヘッダ・インデックスブロック更新部、
3 2 6 表示画像制御部、 3 2 8 画像ファイル生成部、 4 0 0 メインプロセッサ
、 4 0 2 グラフィックプロセッサ、 4 0 4 レンダリング部、 4 0 6 階層デー
タ更新部、 4 0 8 GPUカラーバッファ、 4 1 0 GPU-Zバッファ、 4 1 2
ダーティマスク、 4 1 4 カラーバッファ、 4 1 6 Zバッファ、 4 1 8 モデ
ルデータ、 4 2 2 スケーラ、 5 2 0 グラフィックプロセッサ。

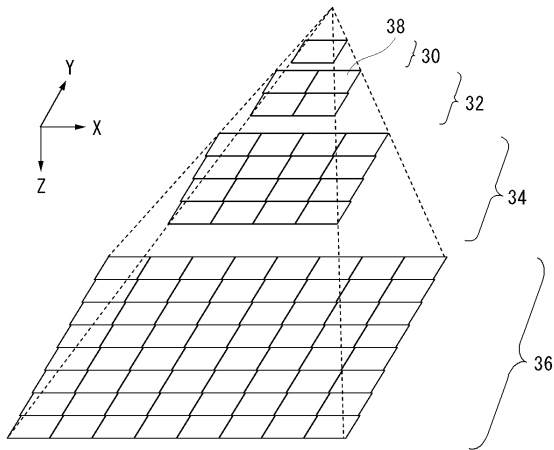
【図1】



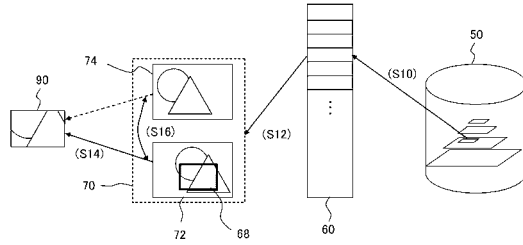
【図2】



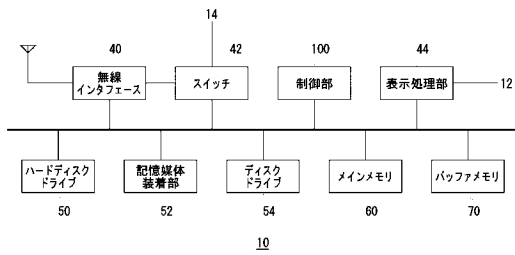
【図3】



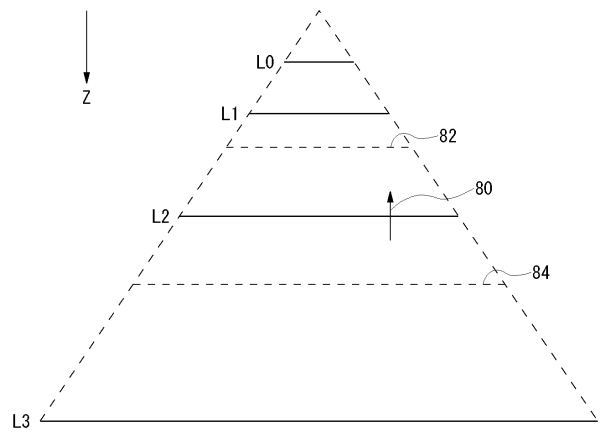
【図5】



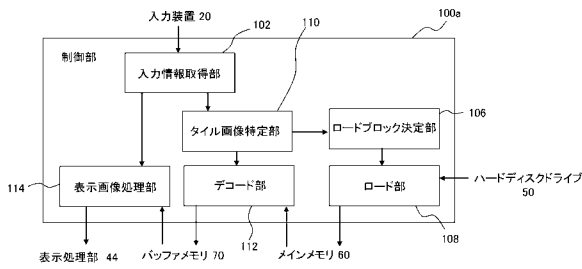
【図4】



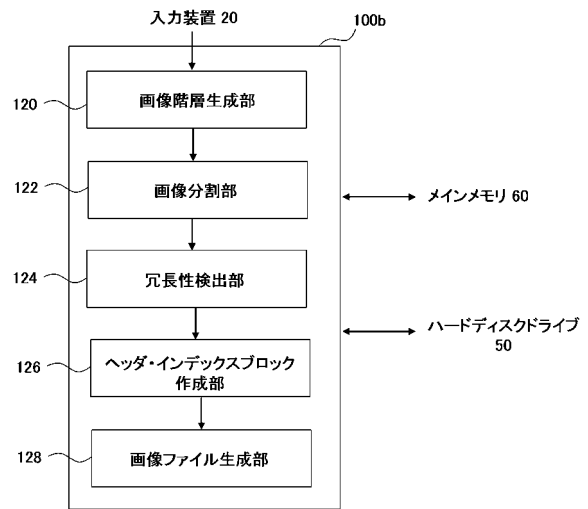
【図6】



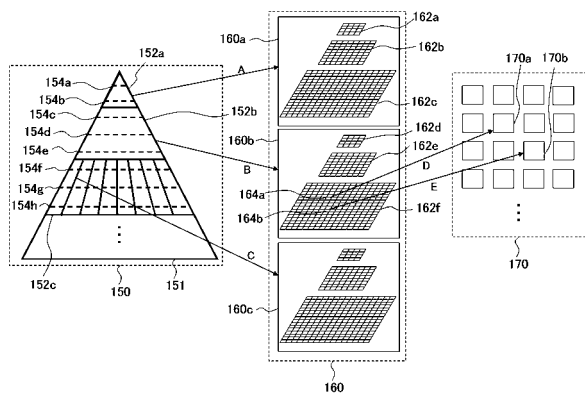
【図7】



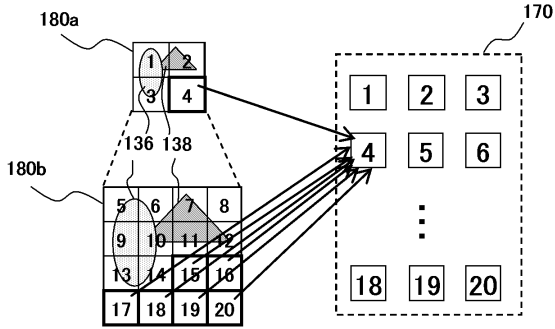
【図9】



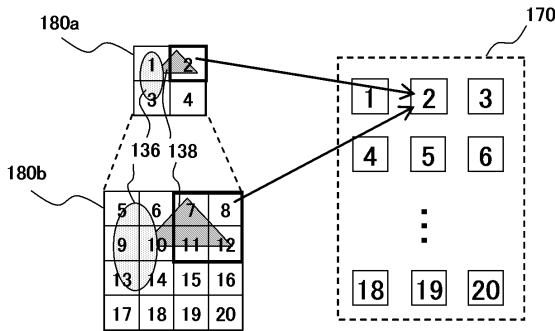
【図8】



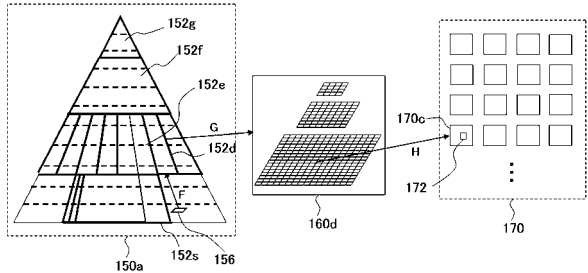
【図10】



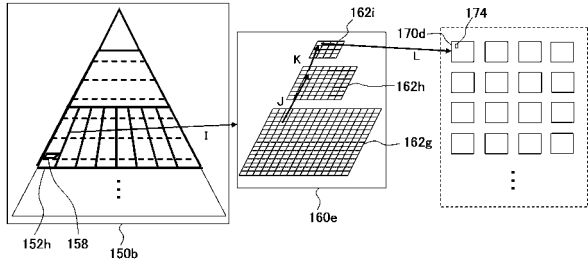
【図11】



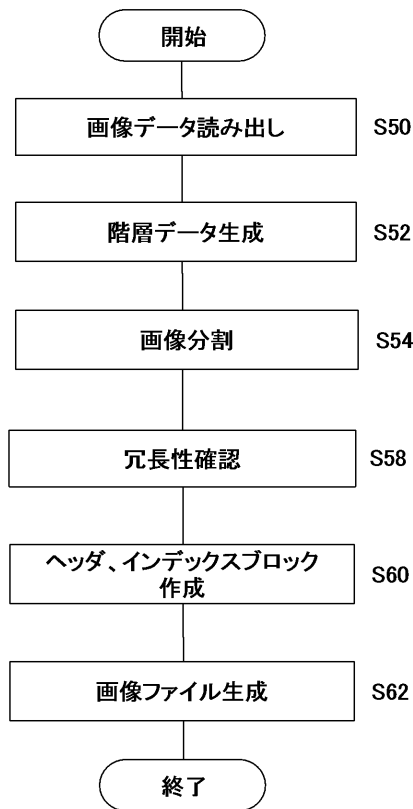
【図12】



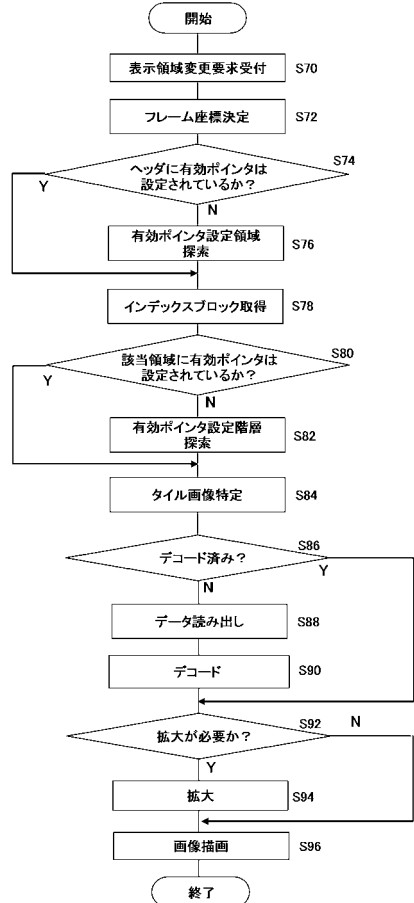
【図13】



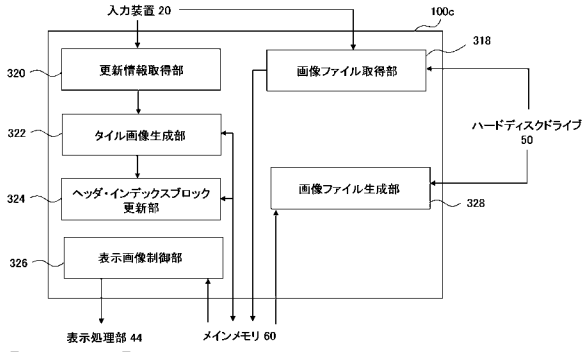
【図14】



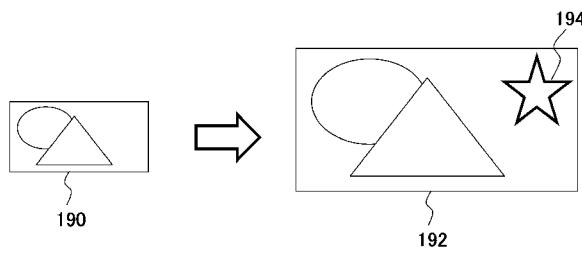
【図15】



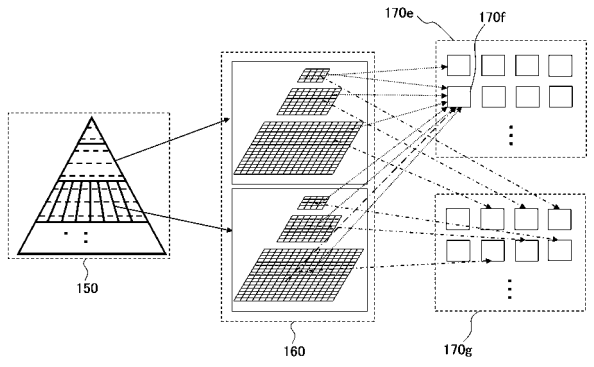
【図16】



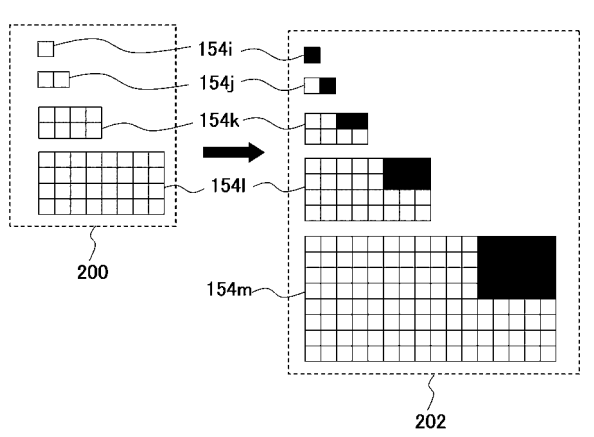
【図17】



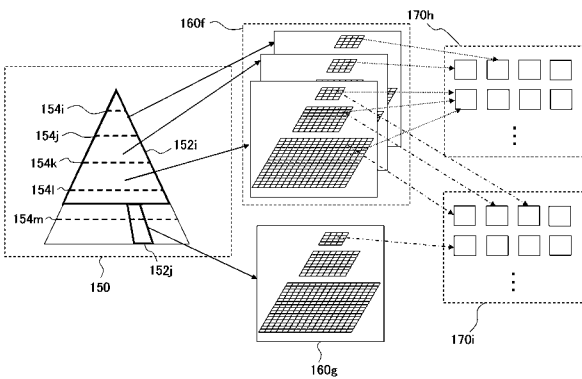
【図18】



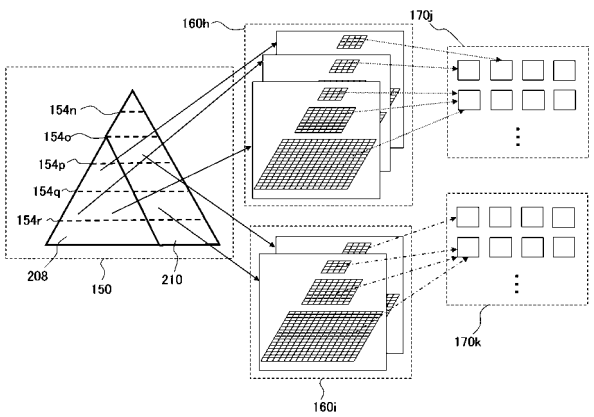
【図19】



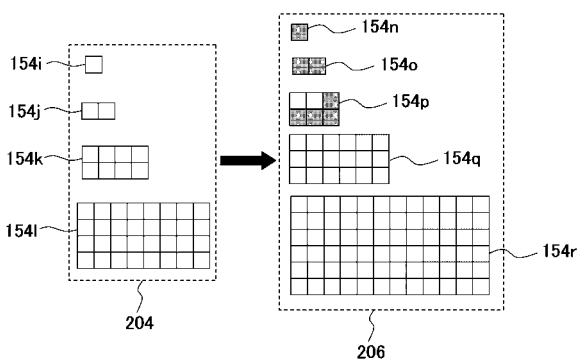
【図20】



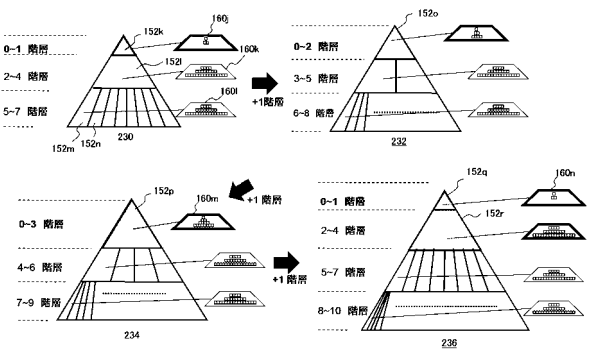
【図22】



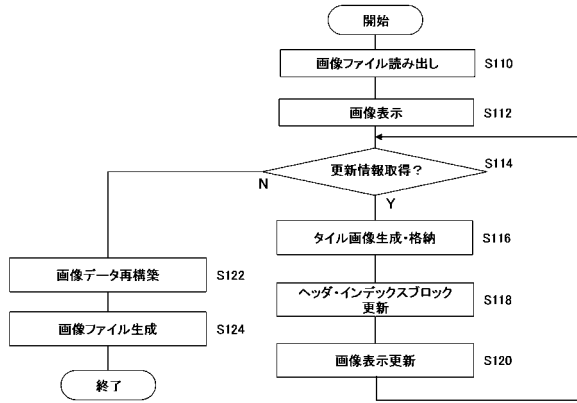
【図21】



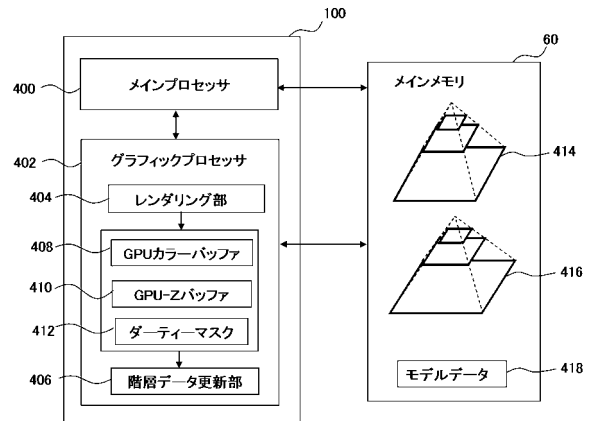
【図23】



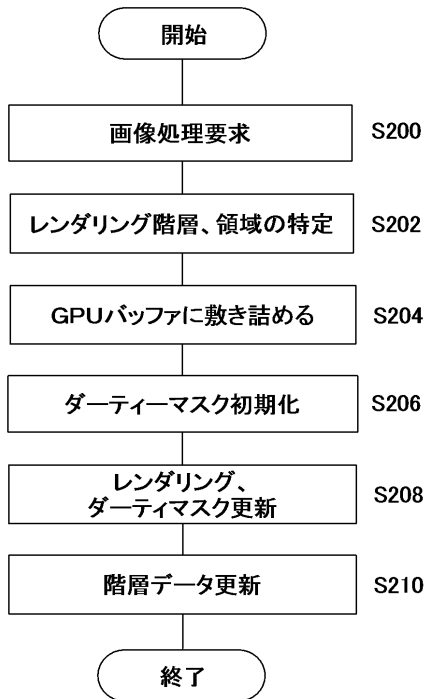
【図24】



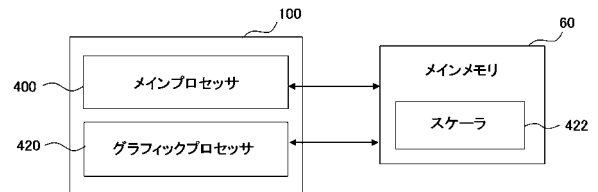
【図25】



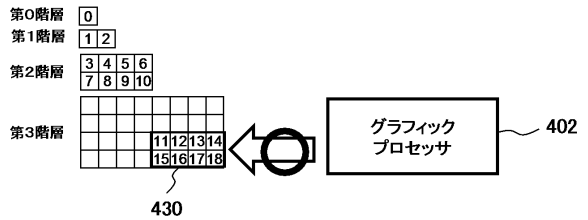
【図26】



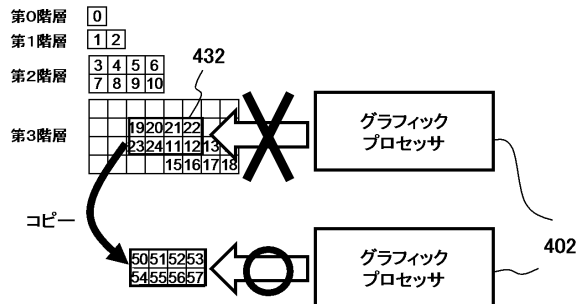
【図27】



【図28】



【図29】



フロントページの続き

(51) Int.Cl. F I
G 0 9 G 5/36 (2006.01) G 0 9 G 5/36 5 2 0 E
G 0 9 G 5/38 (2006.01) G 0 9 G 5/38 A

(72) 発明者 稲田 徹悟
 東京都港区南青山 2 丁目 6 番 2 1 号 株式会社ソニー・コンピュータエンタテインメント内

審査官 伊知地 和之

(56) 参考文献 特開平 0 7 - 1 3 1 6 5 8 (J P , A)
 特開 2 0 0 4 - 3 4 8 7 0 2 (J P , A)
 特開平 0 9 - 3 2 5 7 5 9 (J P , A)
 特開 2 0 0 3 - 5 0 4 9 2 1 (J P , A)
 特開平 0 2 - 2 8 9 0 9 7 (J P , A)
 特開平 0 6 - 2 2 5 1 5 3 (J P , A)

(58) 調査した分野 (Int.Cl. , DB 名)

G 0 6 T 1 / 0 0
 H 0 4 N 5 / 9 1
 G 0 6 F 1 2 / 0 0
 G 0 6 T 3 / 4 0
 G 0 6 T 1 1 / 0 0 - 1 9 / 2 0
 G 0 9 G 5 / 3 6
 G 0 9 G 5 / 3 8
 H 0 4 N 1 / 3 8 7
 H 0 4 N 1 / 4 1