

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第4690723号

(P4690723)

(45) 発行日 平成23年6月1日(2011.6.1)

(24) 登録日 平成23年2月25日(2011.2.25)

(51) Int.Cl. F I
G 0 6 F 12/02 (2006.01) G O 6 F 12/02 5 8 O D
G 0 6 F 12/00 (2006.01) G O 6 F 12/00 5 8 O

請求項の数 15 (全 17 頁)

(21) 出願番号	特願2004-514011 (P2004-514011)	(73) 特許権者	504140417
(86) (22) 出願日	平成15年4月23日(2003.4.23)		アナログ・デバイス・インク
(65) 公表番号	特表2005-530246 (P2005-530246A)		アメリカ合衆国、マサチューセッツ州 O
(43) 公表日	平成17年10月6日(2005.10.6)		2062-9106、ノーウッド、ワン・
(86) 国際出願番号	PCT/US2003/012693		テクノロジー・ウェイ(番地なし)
(87) 国際公開番号	W02003/107272	(74) 代理人	100071010
(87) 国際公開日	平成15年12月24日(2003.12.24)		弁理士 山崎 行造
審査請求日	平成18年4月20日(2006.4.20)	(74) 代理人	100121762
(31) 優先権主張番号	10/170,604		弁理士 杉山 直人
(32) 優先日	平成14年6月13日(2002.6.13)	(74) 代理人	100126767
(33) 優先権主張国	米国(US)		弁理士 白銀 博
		(74) 代理人	100118647
			弁理士 赤松 利昭
		(74) 代理人	100129713
			弁理士 重森 一輝

最終頁に続く

(54) 【発明の名称】 画像片とサーキュラーアドレッシング構成を使用する画像データ処理方法及び装置

(57) 【特許請求の範囲】

【請求項 1】

初期メモリに記憶された画像データを処理する方法であって、

画像を複数の隣接する片に論理的に分割し、

サーキュラーアドレッシング構成を有する動作メモリであって前記初期メモリよりも高速であり複数の連続したアドレス位置を有する動作メモリに第1の複数の前記片をそれらの片が隣接し開始アドレスを有しかつ隣接した状態に記憶し、

前記第1の複数の片を前記動作メモリを介して前記開始アドレスに関して処理し、

前記第1の複数の片が処理されているときに、前記第1の複数の片に隣接する新しい片を前記動作メモリに記憶し、

前記第1の複数の片からの少なくとも1つの片を前記新しい片と共に処理し、

全ての片が処理されるまで連続する片を記憶し処理し、各新しい片は少なくとも1つの先行する片と共に処理されることを含んでなる方法。

【請求項 2】

請求項1の方法であって、前記第1の複数の片を記憶することは前記複数の片のうちの異なる片の隣接部分を前記動作メモリの隣接アドレス位置に記憶する方法。

【請求項 3】

請求項1の方法であって、前記動作メモリは、第1記憶位置が論理的に最後記憶位置に隣接する前記第1記憶位置と前記最後記憶位置とを有する方法。

【請求項 4】

10

20

請求項 1 の方法であって、前記処理は前記開始アドレスからのオフセットを計算する方法。

【請求項 5】

請求項 1 の方法であって、前記動作メモリは与えられた数の片領域に分割され、前記各片の画像データは複数列画像データを含んでなり、前記記憶することは、前記与えられた数の片領域に基づいて前記複数列を前記動作メモリに記憶するために前記複数列をシフトさせることを含んでなる方法。

【請求項 6】

初期メモリに記憶された画像データを処理するためにコンピュータシステム上で使用するコンピュータプログラムであって、前記コンピュータプログラムはコンピュータで読み込み可能なメディアに記憶され、前記コンピュータプログラムは、

前記画像を複数の隣接の片に論理的に分割するプログラムと、

サーキュラーアドレッシング構成を有する動作メモリであって初期メモリよりも高速でありかつ複数の連続したアドレス位置を有する動作メモリに第 1 の複数の前記片を記憶するプログラムであって、前記第 1 の複数の片が隣接し開始アドレスを有しかつ隣接した状態に前記動作メモリに記憶させるプログラムと、

前記第 1 の複数の片を前記動作メモリを介して前記開始アドレスに関して処理する処理プログラムと、

前記第 1 の複数の片が処理されているときに、前記第 1 の複数の片に隣接する新しい片を前記動作メモリに記憶するプログラムと、

前記第 1 の複数の片からの少なくとも 1 つの片を前記新しい片と共に処理するプログラムとを含み、

全ての片が処理されるまで連続する片を記憶し処理し、各新しい片を少なくとも 1 つの先行する片と共に処理するコンピュータプログラム。

【請求項 7】

請求項 6 のコンピュータプログラムであって、前記第 1 の複数の片を隣接した状態に前記動作メモリに記憶する前記プログラムは、前記複数の片のうちの異なる片の隣接部分を前記動作メモリの隣接するアドレス位置に記憶するプログラムを含む、コンピュータプログラム。

【請求項 8】

請求項 6 のコンピュータプログラムであって、動作メモリは、第 1 記憶位置が論理的に最後記憶位置に隣接する前記第 1 記憶位置と前記最後記憶位置とを有するコンピュータプログラム。

【請求項 9】

請求項 6 のコンピュータプログラムであって、前記処理プログラムは前記開始アドレスからのオフセットを計算するプログラムを含むコンピュータプログラム。

【請求項 10】

請求項 6 のコンピュータプログラムであって、前記動作メモリが与えられた数の片領域に分割され、前記各画像データは複数列の画像データを含み、前記記憶させるプログラムは、前記与えられた数の片領域に基づいて前記複数列を前記動作メモリに記憶するために前記複数列をシフトさせるプログラムをさらに含んでなるコンピュータプログラム。

【請求項 11】

論理的に複数の隣接片に分割されて初期メモリに記憶された画像データを処理する装置であって、

前記初期メモリよりも高速であり複数の連続したアドレス位置を有する動作メモリにサーキュラー状態でアクセスするアドレスマネージャと、

第 1 の複数の片を前記動作メモリに記憶するデータ転送装置であって、第 1 の複数の片が隣接し開始アドレスを有しかつ隣接した状態に前記動作メモリに記憶させ、前記第 1 の複数の片が処理されているときに、前記第 1 の複数の片に隣接する新しい片を前記動作メモリに記憶するデータ転送装置と、

前記第 1 の複数の片を前記動作メモリを介して前記開始アドレスに関して処理するプロセッサであって、前記第 1 の複数の片からの少なくとも 1 つの片を前記新しい片と共に処理し、全ての片が処理されるまで連続する片を処理し、各新しい片を少なくとも 1 つの先行する片と共に処理するプロセッサとを、

含んでなる装置。

【請求項 1 2】

請求項 1 1 の装置であって、前記データ転送装置は、前記複数の片の異なる片の隣接部分を前記動作メモリの隣接アドレス位置に記憶する装置。

【請求項 1 3】

請求項 1 1 の装置であって、前記動作メモリは、第 1 記憶位置が最後記憶位置に論理的に隣接する前記第 1 記憶位置と前記最後記憶位置とを有する装置。

10

【請求項 1 4】

請求項 1 1 の装置であって、前記プロセッサは、前記開始アドレスからのオフセットを計算する装置。

【請求項 1 5】

請求項 1 1 の装置であって、前記動作メモリが所定数の片領域に分割され、前記各片内の前記画像データは複数列の画像データを含み、前記データ転送装置は、前記所定数の片領域に基づいて前記複数列を前記動作メモリに記憶するために前記複数列をシフトさせる装置。

【発明の詳細な説明】

20

【技術分野】

【0 0 0 1】

本発明は一般にメモリ管理に関し、特に、図形処理システムのメモリ効率を高めることに関する。

【背景技術】

【0 0 0 2】

近代的なコンピュータシステムは、多くの利点の中でもとりわけ、生き生きとしたグラフィカル表示を作り出す能力を有する。その結果、コンピュータシステムは写真やグラフィックデザインなどの様々なグラフィックアートにおいて広く使用されるツールになった。しかしながら、そのような目的のための増加するコンピュータ用法はデザインの複雑さと、それに対応してグラフィックイメージを作るソフトウェアの複雑さを増加させる結果に至った。

30

【0 0 0 3】

技術は、コンピュータシステムの処理能力を高めることによって、この必要性に応じてきた。例えば、多くのコンピュータシステムは、メインプロセッサの計算負荷を減少させる目的で設計されたグラフィックプロセッサを捧げた。その結果、そのような多くのコンピュータは主記憶装置（すなわち、RAM）又はキャッシュなどの他のコンピュータ構成装置のスピードを超える処理速度を有する。事実、多くの従来のパーソナルコンピュータは、メインメモリとキャッシュのスピードを単独で超えるメインプロセッサを有する。従って、プロセッサの速度は上がったが、メインメモリとキャッシュはコンピュータシステム内で処理ボトルネックを生じて画像ラスタライズ処理を妨害している。

40

【発明の概要】

【0 0 0 4】

本発明の 1 局面によると、比較的高速なサーキュラーメモリ（circular memory）装置は、他のプロセスとの組合せによりイメージプロセッシング効率を高める。そのために、初期メモリに記憶された処理画像データを処理する方法と装置は、論理的に、画像を複数の隣接片に分割する。第 1 の複数の片は、サーキュラーアドレッシング構成を有する動作メモリであって、初期メモリよりも高速かつ複数の連続したアドレス位置を有する動作メモリに記憶される。第 1 の複数の片は隣接し、開始アドレスを有する。さらに、第 1 の複数の片は隣接した状態で動作メモリに記憶され、動作メモリを介して開始アドレスに関し

50

て処理される。

【0005】

いくつかの実施の形態では、(複数の片のうちの)異なる片の隣接する部分が動作メモリの隣接アドレス位置に記憶されるとき、第1の複数の片を隣接した状態に記憶させる。サーキュラー状態(円状性)を維持するために、動作メモリは、第1記憶位置が論理的に最後記憶位置に隣接する第1記憶位置と最後記憶位置を有することとしてもよい。そのうえ、処理の間、開始アドレスからのオフセットを計算することとしてもよい。

【0006】

他の実施の形態では、第1の複数の片が処理されているときに新しい片は動作メモリに記憶される。新しい片は第1の複数の片に隣接する。したがって、第1の複数の片から少なくとも1つの片は新しい片と共に処理される。動作メモリを与えられた数の片領域に分割してもよく、各片の中のイメージデータは複数列の画像データを含んでいる。そのような実施の形態では、与えられた数の片領域に基づいて、複数列をそれらを動作メモリに記憶させるためにシフトさせることによって、片を記憶することとしてもよい。

【0007】

発明の他の局面によると、複数の隣接する片に論理的に分割された初期メモリに記憶された画像データを処理する装置は動作メモリにサーキュラー状態でアクセス可能なアドレスマネージャを含んでいる。動作メモリは初期メモリよりも高速であり複数の連続したアドレス位置を有する。装置はまた、隣接し開始アドレスを有している第1の複数の片を動作メモリに記憶するデータ転送装置を含んでいる。第1の複数の片は隣接状態で動作メモリに記憶される。装置は、動作メモリを介して第1の複数の片を開始アドレスに関して処理するプロセッサをさらに含む。

【0008】

発明の例示的な実施の形態は、コンピュータの読み込み可能なプログラムコードが記憶されたコンピュータで使用可能なメディアを有するコンピュータプログラム製品として実行される。コンピュータで読み込み可能なコードは、従来のプロセスに従うコンピュータシステムによって読まれ利用することができる。

【発明の実施するための最良の形態】

【0009】

発明の上記内容及び長所は添付図面を参照して以下になされる説明から、より完全に理解されるであろう。

【0010】

発明の例示的な実施の形態では、コンピュータシステムの画像データは、低速メモリから、処理用のより高速なサーキュラー(円状)構成のメモリへ移される。例示的な実施の形態の詳細を以下で論じる。

【0011】

図1は例示的な実施の形態を実行することができる例示的なコンピュータシステムを概略的に示す。図1の例示的なコンピュータシステム100を説明目的だけのために論じるものであり、発明をそのように限定するものと考えべきでない。以下の説明は特定のコンピュータシステムを説明する際に一般的に使用される用語に言及するが、説明された概念は図1で示すアーキテクチャと異なるアーキテクチャを有するシステムを含む他のコンピュータシステムに等しく適用される。

【0012】

コンピュータ100は従来のマイクロプロセッサと、情報を一時的に記憶するためのメイン(システム)メモリ110(例えば、当該技術において一般的に「RAM」と呼ばれるランダムアクセスメモリ)と、読出し専用情報を継続的に記憶する固定メモリ(ROM)115とを有する中央演算処理装置(CPU)105を含んでいる。さらに、コンピュータ100はまた、システムRAM110を制御するメモリコントローラ120と、システムRAM110へのダイレクトメモリアccessを制御するダイレクトメモリアccess(DMA)コントローラ160を有する。コンピュータ100はまた、様々な内部構成装置

を結合するメインバス130を含んでいる。

【0013】

マスストレージ（大容量外部記憶装置）はディスク142、デジタル・バーサタイル・ディスク（当該技術分野において一般的に「DVD」と呼ばれるものであり、図示省略する）、CD-ROM147、ハードディスク152などの公知の不揮発性外部記憶装置メディアによって提供することができる。そのうえ、ディスク142、CD-ROM147などのリムーバブルメディアを介し、または、ネットワーク結合を介して、コンピュータシステム100に対してデータとソフトウェアを交換することもできる。

【0014】

コンピュータ100に入力されるユーザ入力はいくつかの装置によって提供することができる。例えば、キーボード156とマウス157をキーボード及びマウスコントローラ155を介してバス130に接続できる。デジタルカメラ、ペン及び／又はタブレット、音声入力のためのマイクロホンなどの他の入力装置をバス130及び適当なコントローラを介してコンピュータ100に接続することができることは当業者にとって明白である。

【0015】

コンピュータシステム100は、望ましくは、WINDOWS NT（登録商標）オペレーティングシステム（アメリカ合衆国ワシントン州レッドモンドのマイクロソフト・コーポレーションから配付される）などのオペレーティングシステムソフトウェアによって制御され調整される。コンピュータシステム制御機能のうち、とりわけ、オペレーティングシステムはシステムリソースの割り当てを制御し、処理スケジューリング、メモリ管理、ネットワーク、および入出力サービスなどのシステムタスクを実行する。特に、例示的な実施の形態（これはメモリ管理に関連する）をオペレーティングシステムの一部として、または、別のユーティリティとして実行してもよい。

【0016】

図2は、発明の例示的な実施の形態に従って処理されるグラフィックイメージ162のフレームを図式的に示す。グラフィックイメージ162は、デジタルカメラまたはビデオ送りなどの公知の如何なるソースから作り出されるデジタル画像であってもよい。一例として、画像162は従来のMP EG標準規格に対応した圧縮作業を行う等の何らかの処理を必要とするかもしれない。

【0017】

そのために、グラフィックイメージ162は論理的に複数の隣接データ片（以下「片」という）に分割され、これらの片は論理的に複数の隣接データセル（以下「セル」という）を形成すると考えられる。各片は画像162のそれぞれの部分を作る2次元配列のピクセルから成る。図2で示される例では、画像162は7片の2つの列に分割される。7片の各列は6つのセルを形成する。片は片A-Nとして識別されるが、セルはセル1-12として識別される。あるセルが隣接の2つの片から成る。例えば、片AとBはセル1を形成する。以下は図2中のセルとそれらの対応する片を示す表である。

【表 1】

セル	片
1	A, B
2	B, C
3	C, D
4	D, E
5	E, F
6	F, G
7	H, I
8	I, J
9	J, K
10	K, L
11	L, M
12	M, N

10

【0018】

セルは、各中央セル（すなわち、列中の最初のセルでもなく、また、最後のセルでもない）が先行セルと後続セルに重なるように各列に形成される。例えば、セル2は片BとCによって形成され、セル3は片CとDによって形成される。従って、セル2と3は片Cを共有する。同じように、セル1と2は片Bを共有する。以下で極めて詳細に説明されるように、このオーバラッピングはコンピュータ100の内部メモリシステムのパイプラインを許可することによって並列処理を容易にする。処理速度改良のような片を論理的に指定することの長所は当業者にとって明らかであろう。

20

【0019】

しかしながら、図2中の論理的な片とセルの議論は例示的なものであり、かつ、その数の片とセルだけが満足 of いくものであることを意図するのではないことに注意されたい。いくつかの実施の形態では、例えば、3片または1片でセルを形成することができる。従って、特定の片とセルサイズ／数の議論は発明の様々な実施の形態の範囲を制限することを意図するものではない。

30

【0020】

図2に説明的に示す画像162の片とセルは、画像162が高速メモリにおいて処理される前に論理的に形成される。図3はそのような論理的に分割された画像162を処理するのに使用できる2個のメモリ装置を示す。高レベルにおいて、画像162は初めに初期メモリ装置（初期メモリ164）に記憶され、次に、動作メモリ装置（動作メモリ166）に片ごとに移される。動作メモリ166は初期メモリ164の速度よりも高速である。中央処理装置105は、画像162をそれが動作メモリ166内にある間に処理する。

40

【0021】

特に、図3に関して、初期メモリ164は全体画像162を記憶する外部SDRAM（同期ダイナミックRAM）であってもよい。DMAコントローラは画像162の片を動作メモリ166に移す。この例では、動作メモリ166は内部キャッシュメモリである。画像162を移送し処理する方法の詳細は図5と6に関して以下でより詳細に論じられる。しかしながら、SDRAMとキャッシュについて論じるが、そのような相対的なスピードを有する他のタイプのメモリを使用することもできることに留意されたい。従って、データ処理のために、より速いメモリに移す如何なる低速メモリ装置も多くの実施の形態において十分である。

【0022】

図4Aは、図3に示す第2メモリ（キャッシュメモリ）の構成の追加詳細を示す。動作

50

メモリ166は、それぞれが画像162の1つのピクセルに関するデータを記憶する所定数のアドレス位置（「記憶位置」とも呼ばれる）を有するように構成される。例えば、図4Aに示す動作メモリ166は96ピクセルのデータに関するアドレス位置を有する。それぞれのアドレス位置は、初期メモリ164から受け取った未加工のイメージデータを記憶するために適宜大きさにされる。従って、それぞれのアドレス位置は与えられたピクセルに必要であるカラー（例えば、赤、緑、および青）、透過性、深み、その他のデータを記憶するに十分なスペースを有するべきである。

【0023】

処理の間、それぞれのアドレス位置を連続して横移動してもよい。例えば、最初に読まれるアドレス位置がアドレス位置0であるならば、2番目に読まれるアドレス位置はアドレス位置1であるだろうし、次に読まれるアドレス位置はアドレス位置2、3…となるであろう。代替の実施の形態では、アドレス位置は連続していて処理されない。しかしながら、どちらの実施の形態においても、（以下で論じられるように、「開始アドレス」と呼ばれる）ベースアドレスが選択され、処理はそのベースアドレスに対応して続く。より明確に言えば、処理はベースアドレスに対するオフセット（位置ずれ）で実行される。例えば、ベースアドレスがアドレス位置4であり、次にアドレス位置6-8を処理するならば、論理は、ベースアドレスから2-4オフセットしたアドレス位置を処理するように指定する。

10

【0024】

例示的な実施の形態において、動作メモリ166はサーキュラーメモリとして構成される。従って、アドレス位置95が読まれ、スキップされ、または横移動された後に、処理はアドレス位置0に戻る。そのうえ、動作メモリ166は、画像162中の片に対応する2つ以上のグループに論理的に仕切られる。より明確に言えば、図4Aに示される動作メモリ166は3つの等しい大きさに分けられたブロックに分割される。これらのブロックはここでは、第1ブロック166A、第2ブロック166B、および第3ブロック166Cと呼ばれる。各メモリブロックは、画像データの1つの片のアドレス位置の数と同じ総数を有するように構成される。このように、例示的な実施の形態では、各メモリブロックは、画像162の1つの片として正確な二次元サイズを有するように構成される。

20

【0025】

図4Bは図4Aに示す動作メモリ166の別の論理的な概要を示す。特に、動作メモリ166は、ポインタその他のメモリ横移動装置にとって、1次元配列のアドレス位置であるように見える。図示の例では、アドレス位置95が横移動された後にアドレス位置0が読まれ、スキップされ、または横移動される。それにもかかわらず、隣接の画像データは図4Aに示されるアドレス位置に記憶される。

30

【0026】

発明の他の要素と同様に、図4Aと4Bに示すアドレス位置の総数は例示的である。従って、異なった数のアドレス位置と区分を使用してもよく、異なった大きさのメモリを使用してもよい。このように、特定のサイズに関する議論は発明の範囲を制限することを意図しない。

【0027】

図5は、図2に示す画像162を処理する例示的な方法を示す。方法はステップ500で始まる。そこでは、画像162は論理的に複数の片に分割される。例えば、画像162を図2に示すように分割することとしてもよい。画像162のサイズに依存して、画像162を1列又は2以上の列の片に分割してもよい。そこでは、上で述べたように、各列は複数の重なるセルを形成する。このように、残りのステップは画像162の各列に関して実行される。

40

【0028】

明確に言えば、第1セットの複数の隣接片は初期メモリ164（すなわち、レベル2メモリなどの低速メモリ）から動作メモリ166（すなわち、レベル1メモリなどの高速メモリ）へ移される。例示的な実施の形態では、片AとB（すなわち、セル1）を最初に、

50

動作メモリ166の第1ブロック166Aと、第2ブロック166Bにロードすることができる。図示のように、片は隣接状態で動作メモリ166にロードされる。より明確に言えば、隣接する2つの片の間の隣接ピクセルは動作メモリ166内の隣接アドレス位置に記憶される。図示の例では、2つの隣接する片は同じ列の隣接ピクセルを有する。

【0029】

そのために、DMAコントローラは2次元DMAとして構成される。特に、DMAコントローラは実例として2つの作業を実行する。すなわち、1つは、セル（すなわち、片のセット）をFIFO（先入れ先出し）パイプにロードする「ロード」作業であり、2番目は、FIFOパイプからの画像データを2次元配列のピクセル（すなわち、片又はセル）として動作メモリ166に記憶する「記憶」作業である。各片が上で論じられるように画像162の部分を形成するピクセルデータの2次元配列であることに注意する。従って、これらの作業は、片データ形式が初期のメモリ164に記憶されるように維持されることを確実にする。

【0030】

例示的な実施の形態では、2つの有名なDMA作業のためのソフトウェアは以下のよう

【表2】

```
// The DMA_LOAD semantics (セマンティクス):
movtofifo (short *in, int xc, int xs, int ye, int ys)
{
    while (yc) {
        for (x=0;x<xc;x++) {
            *fifo_write++ = *in;
            in += xs;
        }
        in += ys;
        yc--;
    }
}

// The DMA_STORE semantics (セマンティクス):
movfromfifo (short *out, int xc, int xs, int yc, int ys)
{
    while (yc) {
        for (x=0;x<xc;x++) {
            *out = *fifo_read++;
            out += xs;
        }
        out += ys;
        yc--;
    }
}

//Parameters (パラメータ):
xs - xスライド
xc - xカウント
ys - yスライド
yc - yカウント
sz - 要素サイズ8ビット、16ビット、32ビット
```

【0031】

第1セットの片がロードされた後に、対のポインタはステップ504で設定される。より詳しくは、システムは、任意セルの第1アドレス位置（動作メモリ166のもの）を示すベースポインタと、該ベースポインタに対してオフセットしたアドレス位置（これも動作メモリ166のもの）を示す読取りポインタを有するように構成されている。初めに、両方のポインタは同じアドレス位置を示す。例えば、ポインタは初めに片Aにおける第1アドレス位置を示すように設定される。図6（以下で論じられる）に示すように、ベースポインタは、新しいセルが処理される都度、そのアドレス位置セット数を増加させ、一方

、読取りポインタは、セル上を横移動するときに各セル内において（各アドレス位置のために）一旦増加する。

【0032】

したがって、方法はステップ506へと続き、そこでは現在のセットの片（すなわち、現在処理されているセル）は以下で図6に関して論じられる方法に従って処理される。処理されている列が処理する追加片を有するか否かが次にステップ508で決定される。その列の追加片を処理する必要があるならば、（その列のための）処理は終わる。

【0033】

それとは逆に、追加片の処理が必要であるならば、処理はステップ510へと続き、ここでは、次のセットの片は動作メモリ166に記憶される。図2に示す例では、単一片のデータ（例えば、片C）は動作メモリ166内の次に続くブロック（この場合、動作メモリ166の3番目のブロック166C）に記憶される。処理は次にステップ504に戻り、そこでは、ポインタがリセットされる。そのために、ベースポインタと読取りポインタは、次のセル（例えば、片Bの第1記憶位置）の第1アドレス位置を示すように設定される。従って、次のセル（片BとCから成る）は処理され、その結果、データ（すなわち、片B）の再利用を可能にする。この方法は列全体が処理されるまで繰り返される。画像162が追加列を有するならば、方法は次の列のために繰り返される。

【0034】

図示の実施の形態では、図5の方法の様々なステップは異なる順番で又は実質的に同時に実行される。例えば、現在のセットの片が処理されている間（ステップ506）、CPU105は、追加片の処理の必要があるかどうかを決定し（ステップ508）、必要がある場合、さらに、動作メモリ166の次のブロックにそのような片を記憶する（ステップ510）。従って、いくつかの実施の形態はこれらの3ステップを実質的に1つの並列操作に結合する。

【0035】

図6はステップ506によって記載される現在のセットの片を処理する方法を示す。方法はステップ600で始まり、そこでは、ベースポインタと読取りポインタの位置が定められる。このステップは本処理に内在するものであり、したがって、様々な実施の形態において必要性が肯定されるものではない。処理は次にステップ602へと続き、そこでは、1セットのアドレス位置が処理のために読まれる。そのために、読取りポインタはベースポインタから1セット数のアドレス位置を読取るために増加される。ここで論じる例では、読取りポインタは、現在のセル中の第1番目の8つのアドレス位置を読み出すために8倍に増加される。従って、このステップは、方法が、セルデータを有する動作メモリ166の2つの隣接ブロックの一部分を読取れるようにする。

【0036】

アドレス位置のセット後に、読取りポインタは次に続く1セットのアドレス位置をスキップする（ステップ604）。図示の例では、読取りポインタは次の4つのアドレス位置をスキップする。このステップは、このように、方法に、セルデータを有していない動作メモリ166の1ブロックをスキップさせる。次にステップ606において、読取りポインタがベースポインタによって示されたもの（すなわち、ベースアドレス）と同じアドレスを示しているか否かが決定される。ベースアドレスを示しているならば、全体のセルが動作メモリ166に記憶されたので、方法は終了する。逆に、読取りポインタが異なったアドレス位置を示しているならば、処理はステップ602に戻り、そこでは、1セット数の次のアドレス位置が読まれる。従って、この方法は、全体セルが動作メモリ166に記憶されるまで繰り返される。上で述べたように、アドレス位置を読取った後に、CPU105はイメージデータを圧縮する等のいくつかの処理機能を実行してもよい。

【0037】

従って、図5と6で説明される方法は、2次元配列のピクセルデータが動作メモリ166に記憶されることを可能にする。ベースポインタから読取り／記憶プロセスを開始し、かつ、動作メモリ166をサーキュラーメモリとして構成することにより、上記記憶を効

10

20

30

40

50

率的に行うことができる。

【0038】

以下は一例であり、図4Aに示す動作メモリ166における、ここで説明する方法を9回反復することを示すものである。より詳しく言えば、この例はイメージプロセスを9回反復することにより動作メモリ166の内容を示す。画像162は列として連続してアドレッシングされるピクセルを有する。言い換えれば、各列は64の連続してアドレッシングされるピクセル位置を有する。

【0039】

この例のパラメータは以下の通りである。

【0040】

画像サイズ：64 x 64

片サイズ：4 x 8

片数：1列あたり16片、8列の片、合計128片

セルサイズ：8 x 8（2片）

サーキュラーアドレッシングサイズ：3 * 4 * 8

【0041】

以下の例において、画像162の各ピクセルは次のフォーマット、a（列、カラム）で識別される。したがって、「a（0, 1）」のような印は、画像162の列0、カラム1に位置するピクセルのピクセルデータを表す。さらに、動作メモリ166の第1カラムの各記憶位置アドレス数は動作メモリ166の第1カラムのすぐ左に示される。もちろん、例示的な動作メモリ166のすべてのアドレスもまた図4Aに関して確かめることができる。最後に、各反復のベースアドレスは大文字で書かれかつアンダーラインが引かれており、一方、画像データを受け取るメモリ部分は太文字で記載されている。

【0042】

シミュレーション出力：

反復（1）、ベースアドレス = 0

【表3】

0000:	<u>a[0,0]</u>	a[0,1]	a[0,2]	a[0,3]	a[0,4]	a[0,5]	a[0,6]	a[0,7]	N/A	N/A	N/A	N/A
0012:	a[1,0]	a[1,1]	a[1,2]	a[1,3]	a[1,4]	a[1,5]	a[1,6]	a[1,7]	N/A	N/A	N/A	N/A
0024:	a[2,0]	a[2,1]	a[2,2]	a[2,3]	a[2,4]	a[2,5]	a[2,6]	a[2,7]	N/A	N/A	N/A	N/A
0036:	a[3,0]	a[3,1]	a[3,2]	a[3,3]	a[3,4]	a[3,5]	a[3,6]	a[3,7]	N/A	N/A	N/A	N/A
0048:	a[4,0]	a[4,1]	a[4,2]	a[4,3]	a[4,4]	a[4,5]	a[4,6]	a[4,7]	N/A	N/A	N/A	N/A
0060:	a[5,0]	a[5,1]	a[5,2]	a[5,3]	a[5,4]	a[5,5]	a[5,6]	a[5,7]	N/A	N/A	N/A	N/A
0072:	a[6,0]	a[6,1]	a[6,2]	a[6,3]	a[6,4]	a[6,5]	a[6,6]	a[6,7]	N/A	N/A	N/A	N/A
0084:	a[7,0]	a[7,1]	a[7,2]	a[7,3]	a[7,4]	a[7,5]	a[7,6]	a[7,7]	N/A	N/A	N/A	N/A

【0043】

反復（2）、ベースアドレス = 4

【表 4】

0000:	a[0, 0]	a[0, 1]	a[0, 2]	a[0, 3]	A[0, 4]	a[0, 5]	a[0, 6]	a[0, 7]	a[0, 8]	a[0, 9]	a[0, 10]	a[0, 11]
0012:	a[1, 0]	a[1, 1]	a[1, 2]	a[1, 3]	a[1, 4]	a[1, 5]	a[1, 6]	a[1, 7]	a[1, 8]	a[1, 9]	a[1, 10]	a[1, 11]
0024:	a[2, 0]	a[2, 1]	a[2, 2]	a[2, 3]	a[2, 4]	a[2, 5]	a[2, 6]	a[2, 7]	a[2, 8]	a[2, 9]	a[2, 10]	a[2, 11]
0036:	a[3, 0]	a[3, 1]	a[3, 2]	a[3, 3]	a[3, 4]	a[3, 5]	a[3, 6]	a[3, 7]	a[3, 8]	a[3, 9]	a[3, 10]	a[3, 11]
0048:	a[4, 0]	a[4, 1]	a[4, 2]	a[4, 3]	a[4, 4]	a[4, 5]	a[4, 6]	a[4, 7]	a[4, 8]	a[4, 9]	a[4, 10]	a[4, 11]
0060:	a[5, 0]	a[5, 1]	a[5, 2]	a[5, 3]	a[5, 4]	a[5, 5]	a[5, 6]	a[5, 7]	a[5, 8]	a[5, 9]	a[5, 10]	a[5, 11]
0072:	a[6, 0]	a[6, 1]	a[6, 2]	a[6, 3]	a[6, 4]	a[6, 5]	a[6, 6]	a[6, 7]	a[6, 8]	a[6, 9]	a[6, 10]	a[6, 11]
0084:	a[7, 0]	a[7, 1]	a[7, 2]	a[7, 3]	a[7, 4]	a[7, 5]	a[7, 6]	a[7, 7]	a[7, 8]	a[7, 9]	a[7, 10]	a[7, 11]

10

【 0 0 4 4 】

反復 (3) 、ベースアドレス = 8

【表 5】

0000:	a[7, 12]	a[7, 13]	a[7, 14]	a[7, 15]	a[0, 4]	a[0, 5]	a[0, 6]	a[0, 7]	A[0, 8]	a[0, 9]	a[0, 10]	a[0, 11]
0012:	a[0, 12]	a[0, 13]	a[0, 14]	a[0, 15]	a[1, 4]	a[1, 5]	a[1, 6]	a[1, 7]	a[1, 8]	a[1, 9]	a[1, 10]	a[1, 11]
0024:	a[1, 12]	a[1, 13]	a[1, 14]	a[1, 15]	a[2, 4]	a[2, 5]	a[2, 6]	a[2, 7]	a[2, 8]	a[2, 9]	a[2, 10]	a[2, 11]
0036:	a[2, 12]	a[2, 13]	a[2, 14]	a[2, 15]	a[3, 4]	a[3, 5]	a[3, 6]	a[3, 7]	a[3, 8]	a[3, 9]	a[3, 10]	a[3, 11]
0048:	a[3, 12]	a[3, 13]	a[3, 14]	a[3, 15]	a[4, 4]	a[4, 5]	a[4, 6]	a[4, 7]	a[4, 8]	a[4, 9]	a[4, 10]	a[4, 11]
0060:	a[4, 12]	a[4, 13]	a[4, 14]	a[4, 15]	a[5, 4]	a[5, 5]	a[5, 6]	a[5, 7]	a[5, 8]	a[5, 9]	a[5, 10]	a[5, 11]
0072:	a[5, 12]	a[5, 13]	a[5, 14]	a[5, 15]	a[6, 4]	a[6, 5]	a[6, 6]	a[6, 7]	a[6, 8]	a[6, 9]	a[6, 10]	a[6, 11]
0084:	a[6, 12]	a[6, 13]	a[6, 14]	a[6, 15]	a[7, 4]	a[7, 5]	a[7, 6]	a[7, 7]	a[7, 8]	a[7, 9]	a[7, 10]	a[7, 11]

20

【 0 0 4 5 】

反復 (4) 、ベースアドレス = 1 2

【表 6】

0000:	a[7, 12]	a[7, 13]	a[7, 14]	a[7, 15]	a[7, 16]	a[7, 17]	a[7, 18]	a[7, 19]	a[0, 8]	a[0, 9]	a[0, 10]	a[0, 11]
0012:	A[0, 12]	a[0, 13]	a[0, 14]	a[0, 15]	a[0, 16]	a[0, 17]	a[0, 18]	a[0, 19]	a[1, 8]	a[1, 9]	a[1, 10]	a[1, 11]
0024:	a[1, 12]	a[1, 13]	a[1, 14]	a[1, 15]	a[1, 16]	a[1, 17]	a[1, 18]	a[1, 19]	a[2, 8]	a[2, 9]	a[2, 10]	a[2, 11]
0036:	a[2, 12]	a[2, 13]	a[2, 14]	a[2, 15]	a[2, 16]	a[2, 17]	a[2, 18]	a[2, 19]	a[3, 8]	a[3, 9]	a[3, 10]	a[3, 11]
0048:	a[3, 12]	a[3, 13]	a[3, 14]	a[3, 15]	a[3, 16]	a[3, 17]	a[3, 18]	a[3, 19]	a[4, 8]	a[4, 9]	a[4, 10]	a[4, 11]
0060:	a[4, 12]	a[4, 13]	a[4, 14]	a[4, 15]	a[4, 16]	a[4, 17]	a[4, 18]	a[4, 19]	a[5, 8]	a[5, 9]	a[5, 10]	a[5, 11]
0072:	a[5, 12]	a[5, 13]	a[5, 14]	a[5, 15]	a[5, 16]	a[5, 17]	a[5, 18]	a[5, 19]	a[6, 8]	a[6, 9]	a[6, 10]	a[6, 11]
0084:	a[6, 12]	a[6, 13]	a[6, 14]	a[6, 15]	a[6, 16]	a[6, 17]	a[6, 18]	a[6, 19]	a[7, 8]	a[7, 9]	a[7, 10]	a[7, 11]

40

【 0 0 4 6 】

反復 (5) 、ベースアドレス = 1 6

【表 7】

0000:	a[7,12]	a[7,13]	a[7,14]	a[7,15]	a[7,16]	a[7,17]	a[7,18]	a[7,19]	a[7,20]	a[7,21]	a[7,22]	a[7,23]
0012:	a[0,12]	a[0,13]	a[0,14]	a[0,15]	A[0,16]	a[0,17]	a[0,18]	a[0,19]	a[0,20]	a[0,21]	a[0,22]	a[0,23]
0024:	a[1,12]	a[1,13]	a[1,14]	a[1,15]	a[1,16]	a[1,17]	a[1,18]	a[1,19]	a[1,20]	a[1,21]	a[1,22]	a[1,23]
0036:	a[2,12]	a[2,13]	a[2,14]	a[2,15]	a[2,16]	a[2,17]	a[2,18]	a[2,19]	a[2,20]	a[2,21]	a[2,22]	a[2,23]
0048:	a[3,12]	a[3,13]	a[3,14]	a[3,15]	a[3,16]	a[3,17]	a[3,18]	a[3,19]	a[3,20]	a[3,21]	a[3,22]	a[3,23]
0060:	a[4,12]	a[4,13]	a[4,14]	a[4,15]	a[4,16]	a[4,17]	a[4,18]	a[4,19]	a[4,20]	a[4,21]	a[4,22]	a[4,23]
0072:	a[5,12]	a[5,13]	a[5,14]	a[5,15]	a[5,16]	a[5,17]	a[5,18]	a[5,19]	a[5,20]	a[5,21]	a[5,22]	a[5,23]
0084:	a[6,12]	a[6,13]	a[6,14]	a[6,15]	a[6,16]	a[6,17]	a[6,18]	a[6,19]	a[6,20]	a[6,21]	a[6,22]	a[6,23]

10

【 0 0 4 7 】

反復 (6) 、ベースアドレス = 2 0

【表 8】

0000:	a[6,24]	a[6,25]	a[6,26]	a[6,27]	a[7,16]	a[7,17]	a[7,18]	a[7,19]	a[7,20]	a[7,21]	a[7,22]	a[7,23]
0012:	a[7,24]	a[7,25]	a[7,26]	a[7,27]	a[0,16]	a[0,17]	a[0,18]	a[0,19]	A[0,20]	a[0,21]	a[0,22]	a[0,23]
0024:	a[0,24]	a[0,25]	a[0,26]	a[0,27]	a[1,16]	a[1,17]	a[1,18]	a[1,19]	a[1,20]	a[1,21]	a[1,22]	a[1,23]
0036:	a[1,24]	a[1,25]	a[1,26]	a[1,27]	a[2,16]	a[2,17]	a[2,18]	a[2,19]	a[2,20]	a[2,21]	a[2,22]	a[2,23]
0048:	a[2,24]	a[2,25]	a[2,26]	a[2,27]	a[3,16]	a[3,17]	a[3,18]	a[3,19]	a[3,20]	a[3,21]	a[3,22]	a[3,23]
0060:	a[3,24]	a[3,25]	a[3,26]	a[3,27]	a[4,16]	a[4,17]	a[4,18]	a[4,19]	a[4,20]	a[4,21]	a[4,22]	a[4,23]
0072:	a[4,24]	a[4,25]	a[4,26]	a[4,27]	a[5,16]	a[5,17]	a[5,18]	a[5,19]	a[5,20]	a[5,21]	a[5,22]	a[5,23]
0084:	a[5,24]	a[5,25]	a[5,26]	a[5,27]	a[6,16]	a[6,17]	a[6,18]	a[6,19]	a[6,20]	a[6,21]	a[6,22]	a[6,23]

20

【 0 0 4 8 】

反復 (7) 、ベースアドレス = 2 4

【表 9】

0000:	a[6,24]	a[6,25]	a[6,26]	a[6,27]	a[6,28]	a[6,29]	a[6,30]	a[6,31]	a[7,20]	a[7,21]	a[7,22]	a[7,23]
0012:	a[7,24]	a[7,25]	a[7,26]	a[7,27]	a[7,28]	a[7,29]	a[7,30]	a[7,31]	a[0,20]	a[0,21]	a[0,22]	a[0,23]
0024:	A[0,24]	a[0,25]	a[0,26]	a[0,27]	a[0,28]	a[0,29]	a[0,30]	a[0,31]	a[1,20]	a[1,21]	a[1,22]	a[1,23]
0036:	a[1,24]	a[1,25]	a[1,26]	a[1,27]	a[1,28]	a[1,29]	a[1,30]	a[1,31]	a[2,20]	a[2,21]	a[2,22]	a[2,23]
0048:	a[2,24]	a[2,25]	a[2,26]	a[2,27]	a[2,28]	a[2,29]	a[2,30]	a[2,31]	a[3,20]	a[3,21]	a[3,22]	a[3,23]
0060:	a[3,24]	a[3,25]	a[3,26]	a[3,27]	a[3,28]	a[3,29]	a[3,30]	a[3,31]	a[4,20]	a[4,21]	a[4,22]	a[4,23]
0072:	a[4,24]	a[4,25]	a[4,26]	a[4,27]	a[4,28]	a[4,29]	a[4,30]	a[4,31]	a[5,20]	a[5,21]	a[5,22]	a[5,23]
0084:	a[5,24]	a[5,25]	a[5,26]	a[5,27]	a[5,28]	a[5,29]	a[5,30]	a[5,31]	a[6,20]	a[6,21]	a[6,22]	a[6,23]

40

【 0 0 4 9 】

反復 (8) 、ベースアドレス = 2 8

【表 10】

0000:	a[6,24]	a[6,25]	a[6,26]	a[6,27]	a[6,28]	a[6,29]	a[6,30]	a[6,31]	a[6,32]	a[6,33]	a[6,34]	a[6,35]
0012:	a[7,24]	a[7,25]	a[7,26]	a[7,27]	a[7,28]	a[7,29]	a[7,30]	a[7,31]	a[7,32]	a[7,33]	a[7,34]	a[7,35]
0024:	a[0,24]	a[0,25]	a[0,26]	a[0,27]	a[0,28]	a[0,29]	a[0,30]	a[0,31]	a[0,32]	a[0,33]	a[0,34]	a[0,35]
0036:	a[1,24]	a[1,25]	a[1,26]	a[1,27]	a[1,28]	a[1,29]	a[1,30]	a[1,31]	a[1,32]	a[1,33]	a[1,34]	a[1,35]
0048:	a[2,24]	a[2,25]	a[2,26]	a[2,27]	a[2,28]	a[2,29]	a[2,30]	a[2,31]	a[2,32]	a[2,33]	a[2,34]	a[2,35]
0060:	a[3,24]	a[3,25]	a[3,26]	a[3,27]	a[3,28]	a[3,29]	a[3,30]	a[3,31]	a[3,32]	a[3,33]	a[3,34]	a[3,35]
0072:	a[4,24]	a[4,25]	a[4,26]	a[4,27]	a[4,28]	a[4,29]	a[4,30]	a[4,31]	a[4,32]	a[4,33]	a[4,34]	a[4,35]
0084:	a[5,24]	a[5,25]	a[5,26]	a[5,27]	a[5,28]	a[5,29]	a[5,30]	a[5,31]	a[5,32]	a[5,33]	a[5,34]	a[5,35]

10

【0050】

反復(9)、ベースアドレス=32

【表 11】

0000:	a[5,36]	a[5,37]	a[5,38]	a[5,39]	a[6,28]	a[6,29]	a[6,30]	a[6,31]	a[6,32]	a[6,33]	a[6,34]	a[6,35]
0012:	a[6,36]	a[6,37]	a[6,38]	a[6,39]	a[7,28]	a[7,29]	a[7,30]	a[7,31]	a[7,32]	a[7,33]	a[7,34]	a[7,35]
0024:	a[7,36]	a[7,37]	a[7,38]	a[7,39]	a[0,28]	a[0,29]	a[0,30]	a[0,31]	a[0,32]	a[0,33]	a[0,34]	a[0,35]
0036:	a[0,36]	a[0,37]	a[0,38]	a[0,39]	a[1,28]	a[1,29]	a[1,30]	a[1,31]	a[1,32]	a[1,33]	a[1,34]	a[1,35]
0048:	a[1,36]	a[1,37]	a[1,38]	a[1,39]	a[2,28]	a[2,29]	a[2,30]	a[2,31]	a[2,32]	a[2,33]	a[2,34]	a[2,35]
0060:	a[2,36]	a[2,37]	a[2,38]	a[2,39]	a[3,28]	a[3,29]	a[3,30]	a[3,31]	a[3,32]	a[3,33]	a[3,34]	a[3,35]
0072:	a[3,36]	a[3,37]	a[3,38]	a[3,39]	a[4,28]	a[4,29]	a[4,30]	a[4,31]	a[4,32]	a[4,33]	a[4,34]	a[4,35]
0084:	a[4,36]	a[4,37]	a[4,38]	a[4,39]	a[5,28]	a[5,29]	a[5,30]	a[5,31]	a[5,32]	a[5,33]	a[5,34]	a[5,35]

20

【0051】

上でデモンストレーションされるように、この関連するプロセスは片内の列を事前選択された回数だけ論理的にシフトする効果を有する。特に、このシフトは動作メモリ166のブロック数に基づいて実行される。上の例では、第1セットの3片が記憶された後に、第2セットの片は、動作メモリに記憶されると、1列で下方にシフトされる。次の続くセットの3片が動作メモリ166に記憶されると、それらは2列で下方にシフトされる。このプロセスは、すべてのセルが画像の列において処理されるまで、(上の例では)3片のセットあたり1列毎に繰り返される。

30

【0052】

当業者は、例示的な実施の形態を複数セルの複数列に関して単に例として論じたことを理解するべきである。セルの他の構成を用いることもできる。例えば、同様の原則をセルのカラムに適用することができる。従って、様々な実施の形態はセルの列にのみ制限されることを意図するものではない。

【0053】

上で説明された利点に加えて、例示的な実施の形態は、ポインタの動向をおさえるために中央処理装置(CPU)105を必要としない。代わりに、簡単なモジュールコマンドとカウンタをある条件が満たされるまで(例えば、ベースポインタによって示されたアドレス位置と同じアドレス位置を読み取りポインタが示すまで)、ポインタを増加するのに使用することとしてもよい。従って、中央処理装置の使用が最適化される。さらに、別の1片がロードされている間(すなわち、パイプライン時)に画像データの1つのセルを処理することでシステム性能を改善することができる。

40

【0054】

発明の様々な実施の形態は少なくとも部分的に如何なる従来のコンピュータープログラミング言語でも実行することができる。例えば、いくつかの実施の形態は手続き上のプロ

50

グラミング言語（例えば、「C言語」）又はオブジェクト指向プログラミング言語（例えば、「C++」）で実行することができる。発明の他の実施の形態はプログラムされたハードウェア要素（例えば、適用業務に特化された集積回路、FPGA、およびデジタルシグナルプロセッサ）又は他の関連要素として実施されてもよい。

【0055】

代替の実施の形態では、開示されている装置と方法はコンピュータシステムによって使用されるコンピュータプログラム製品として実施されてもよい。そのような実施はコンピュータで読み込み可能なメディア（例えば、ディスク、CD-ROM、ROM、または固定ディスク）などの触知できるメディアに固定された一連の機械命令又はメディアを介してネットワークに接続された通信アダプタなどのモデムその他のインタフェース装置を介してコンピュータシステムに移送可能な一連の機械命令を含んでもよい。メディアは触知できるメディア（例えば、光学メディア若しくはアナログの通信ライン）又はワイヤレス技術（例えば、マイクロウェーブ、赤外線その他の伝送方式）で実行されるメディアのどちらかであってもよい。

10

【0056】

一連の機械命令は以前にシステムに関してここで説明された機能のすべてか一部を具体化する。多くのコンピュータアーキテクチャ又はオペレーティングシステムに使用する多くのプログラミング言語でそのような機械命令を書くことができることを当業者は理解すべきである。さらに、そのような命令は、半導体、磁気、光学その他のメモリ装置などの如何なるメモリ装置に記憶することができ、また、光学、赤外線、マイクロウェーブその他の伝送技術などの如何なる通信技術を用いて伝送できる。

20

【0057】

そのようなコンピュータプログラム製品は付随の印刷又は電子ドキュメンテーションを備えるリムーバブルメディア（例えば、真空パック包装したソフトウェア）として配付されてもよいし、コンピュータシステム（例えば、システムROMか固定ディスク）にプレロードされてもよいし、またはネットワーク（例えば、インターネットまたはWWW）の上でサーバ又は電子掲示板から配付されることが期待される。もちろん、発明のいくつかの実施の形態はソフトウェア（例えば、コンピュータプログラム製品）とハードウェアの両方の組合せとして実行されてもよい。発明のさらに別の実施の形態は完全にハードウェア、または完全にソフトウェア（例えば、コンピュータプログラム製品）として実行される。

30

【0058】

発明の様々な例示的な実施の形態が明らかにされたが、それが発明の真の範囲から逸脱することなく発明の長所のいくつかを達成できる様々な変形と変更をすることができることは当業者にとって明白である。これら及びその他の変更を添付の特許請求の範囲によってカバーすることを意図する。

【図面の簡単な説明】

【0059】

【図1】例示的な実施の形態を実行することができる例示的なコンピュータシステムを概略的に示す図である。

40

【図2】発明の例示的な実施の形態に従って処理することができるグラフィックイメージフレームを概略的に示す図である。

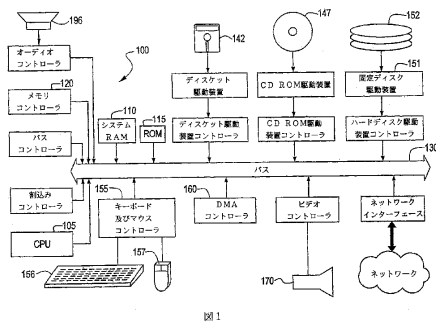
【図3】画像を処理に使用できる2個のメモリ装置を示す図である。

【図4】図4Aは、図3に示す第2メモリ（キャッシュメモリ）の構成の追加的詳細を示す図である。図4Bは、図4Aに示す第2メモリの別の論理的な概要を示す図である。

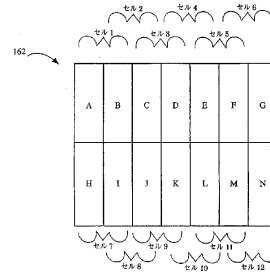
【図5】図2示す画像を処理する例示的な方法を示す図である。

【図6】図3、4A、および4Bに示す第2メモリを横移動する例示的な方法を示す図である。

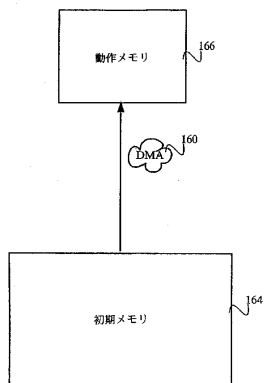
【図 1】



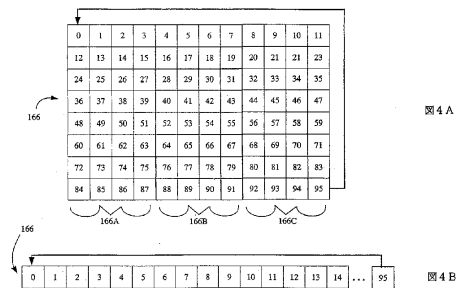
【図 2】



【図 3】



【図 4】



【図 5】

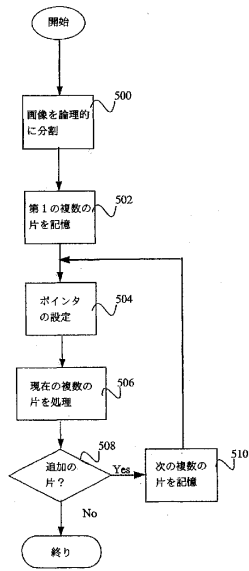


図 5

【図 6】

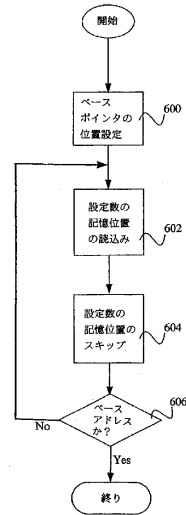


図 6

フロントページの続き

- (72)発明者 ニン、キー
アメリカ合衆国、マサチューセッツ州 02453、ワルサム、ナンバー2、リバー・ストリート
272
- (72)発明者 ホッフマン、マーク
アメリカ合衆国、マサチューセッツ州 02048、マンズフィールド、プレザント・ストリート
99
- (72)発明者 イ、ギャビー
アメリカ合衆国、マサチューセッツ州 02130、ジャマイカ・ブレイン、ナンバー21、エバ
ーグリーン・ストリート 50

審査官 多賀 実

- (56)参考文献 特開平11-288384 (JP, A)
特開昭63-271606 (JP, A)
特開平10-224797 (JP, A)
特開2000-076126 (JP, A)
特開平07-271966 (JP, A)
特開平05-204592 (JP, A)

(58)調査した分野(Int.Cl., DB名)

G06F12/02

G09G5/00-5/40

H04N7/32