

(19) 日本国特許庁(JP)

## (12) 特許公報(B2)

(11) 特許番号

特許第5934380号  
(P5934380)

(45) 発行日 平成28年6月15日(2016.6.15)

(24) 登録日 平成28年5月13日(2016.5.13)

(51) Int.Cl.

H04N 19/597 (2014.01)  
G06T 15/00 (2011.01)

F 1

H04N 19/597  
G06T 15/00 501

請求項の数 17 (全 12 頁)

(21) 出願番号 特願2014-545881 (P2014-545881)  
 (86) (22) 出願日 平成23年12月29日 (2011.12.29)  
 (65) 公表番号 特表2015-510288 (P2015-510288A)  
 (43) 公表日 平成27年4月2日 (2015.4.2)  
 (86) 國際出願番号 PCT/US2011/067902  
 (87) 國際公開番号 WO2013/101095  
 (87) 國際公開日 平成25年7月4日 (2013.7.4)  
 審査請求日 平成26年6月10日 (2014.6.10)

(73) 特許権者 593096712  
 インテル コーポレイション  
 アメリカ合衆国 95054 カリフォルニア州 サンタクララ ミッション カレッジ ブールバード 2200  
 (74) 代理人 100107766  
 弁理士 伊東 忠重  
 (74) 代理人 100070150  
 弁理士 伊東 忠彦  
 (74) 代理人 100091214  
 弁理士 大貫 進介  
 (72) 発明者 アケニネーモレル, トマス ゲー.  
 スウェーデン国 エス-277 38 エム. ルンド ロセンヴァーゲン 13

最終頁に続く

(54) 【発明の名称】可変の深さ圧縮

## (57) 【特許請求の範囲】

## 【請求項 1】

グラフィクスプロセッサを使用して、タイルの深さデータを圧縮するステップであって、前記タイルについて予測した深さ値と実際の深さ値との残差値を符号化するのに使用されるビット数、及び前記タイルのXベクトルとYベクトルのデルタ値に使用されるビット数を、前記深さデータに応じて変化させることによって、前記深さデータを圧縮するステップ

を含む、方法。

## 【請求項 2】

前記タイル内のアンカーポイントに基づく符号化を使用するステップを含む、請求項1に記載の方法。 10

## 【請求項 3】

前記Xベクトル及び前記Yベクトルの前記デルタ値のビット数を選択するステップと、深さごとの前記残差値にビット数を割り当てるステップとを含む、請求項1に記載の方法。

## 【請求項 4】

前記Xベクトル及び前記Yベクトルの前記デルタ値のビット数、並びに深さごとの前記残差値のビット数についての各々の選択を可能にするステップを含む、請求項3に記載の方法。

## 【請求項 5】

複数の選択可能なモードを提供するステップを含み、各モードは、前記 X ベクトル及び前記 Y ベクトルの前記デルタ値、並びに深さごとの前記残差値に対して、特定のビット数を指定する、請求項 1 に記載の方法。

**【請求項 6】**

前記タイル内のアンカーポイントについて、前記 X ベクトル及び前記 Y ベクトルのデルタ値に基づくデルタベクトルを表すのに必要な最少の数のビットを計算するステップを含む、請求項 5 に記載の方法。

**【請求項 7】**

プロセッサによって実行されると、

タイルの深さデータに応じて、前記タイルについて予測した深さ値と実際の深さ値との残差値を符号化するのに使用されるビット数、及び前記タイルの X ベクトルと Y ベクトルのデルタ値に使用されるビット数を変化させることによって、前記深さデータを圧縮する命令を記憶している、非一時的なコンピュータ読み取り可能媒体。 10

**【請求項 8】**

前記タイル内のアンカーポイントに基づく符号化を使用する命令を更に記憶している、請求項 7 に記載の媒体。

**【請求項 9】**

前記 X ベクトル及び前記 Y ベクトルの前記デルタ値のビット数を選択し、深さごとの前記残差値のビット数を割り当てる命令を更に記憶している、請求項 7 に記載の媒体。 20

**【請求項 10】**

前記 X ベクトル及び前記 Y ベクトルの前記デルタ値のビット数、並びに深さごとの前記残差値のビット数についての各々の選択を可能にする命令を更に記憶している、請求項 9 に記載の媒体。

**【請求項 11】**

複数の選択可能モードを提供する命令を更に備え、各モードは、前記 X ベクトル及び前記 Y ベクトルの前記デルタ値、並びに深さごとの前記残差値に対して、特定のビット数を指定する、請求項 7 に記載の媒体。

**【請求項 12】**

前記タイル内のアンカーポイントについて、前記 X ベクトル及び前記 Y ベクトルのデルタ値に基づくデルタベクトルを表すのに必要な最少の数のビットを計算する命令を更に含む、請求項 11 に記載の媒体。 30

**【請求項 13】**

タイルの深さデータに応じて、前記タイルについて予測した深さ値と実際の深さ値との残差値を符号化するのに使用されるビット数、及び前記タイルの X ベクトルと Y ベクトルのデルタ値に使用されるビット数を変化させることによって、前記深さデータを圧縮するプロセッサと、

前記プロセッサに結合されるストレージと  
を備える、装置。

**【請求項 14】**

前記プロセッサは、前記タイル内のアンカーポイントに基づく符号化を使用する、請求項 13 に記載の装置。 40

**【請求項 15】**

前記プロセッサは、前記 X ベクトル及び前記 Y ベクトルの前記デルタ値に使用するビット数を選択し、深さごとの前記残差値にビット数を割り当てる、請求項 13 に記載の装置。

**【請求項 16】**

前記プロセッサは、前記 X ベクトル及び前記 Y ベクトルの前記デルタ値、並びに深さごとの前記残差値のビット数についての各々の選択を可能にする、請求項 15 に記載の装置。

**【請求項 17】**

10

20

30

40

50

前記プロセッサは、複数の選択可能なモードを提供し、各モードは、前記Xベクトル及び前記Yベクトルの前記デルタ値、並びに深さごとの前記残差値に対して、特定のビット数を指定する、請求項1-3に記載の装置。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、一般にグラフィック処理に関し、より具体的には、モーションブラーのラスタ化のための深さバッファの圧縮に関する。

【背景技術】

【0002】

モーションブラーのラスタ化は、動いている物体をより正確に表すこと、特に、物体が十分早く動くときに観察されるブラーをより正確に表すことを試みる。モーションブラーを無視すると、シーン内の基本要素の深さに関する情報は容易に圧縮され得る。特に、各基本要素の深さは、画像化デバイスからその基本要素までの距離として決定され得る。モーションブラーを生じないときは、この深さ情報を低減するためのアルゴリズムが存在する。

【発明の概要】

【発明が解決しようとする課題】

【0003】

しかしながら、モーションブラーを生じるとき、深さ情報の圧縮を試みることはかなり複雑な操作である。

【0004】

グラフィクスプロセッサは、増加したメモリ帯域幅の使用による電力消費の増加に非常に敏感になる傾向がある。また、メモリ帯域幅はそれ自体では、最近のグラフィクスプロセッサにおいて希少なリソースである。現在、モーションブラーのラスタ化及びフィールドの深さを取得し、対話的な又はリアルタイムのレンダリングベースでこの情報を取得することがより重要視されている。これはかなりのメモリ帯域幅の使用を要し、このことは電力消費の増加に関与する。

【課題を解決するための手段】

【0005】

一部の実施形態は、図面に関連して説明される。

【0006】

一部の実施形態によると、深さ圧縮に割り当てるビット数を、考慮事項(consideration)の数に基づいて可変に変更することができる。したがって、深さデータをより効率的な方法で圧縮することができる。

【図面の簡単な説明】

【0007】

【図1】本発明の一実施形態の概略図である。

【図2】一実施形態に係るアンカーポイントのモード選択についてのフローチャートである。

【図3】一実施形態に係る動的な残差値ビット割り当てについてのフローチャートである。

【図4】一実施形態のシステム図である。

【発明を実施するための形態】

【0008】

深さバッファ圧縮は、メモリ帯域幅の使用を低減する1つの技法である。典型的に、タイル内の任意の基本要素の深さ $z$ は、典型的には三角形である、その基本要素上で線形である。これは、安価な圧縮及び圧縮解除アルゴリズムを構築するのに用いられることがある。

【0009】

10

20

30

40

50

平面符号化は、この線型的関係を利用する試みる。ラスタライザは、正確な平面方程式をコンプレッサに供給し、従って残余項は必要とされない。しかしながら、モーションブラーを生じるとき、線形関数に従って各頂点が動く可能性がある。サンプルにおける深さ関数は、有理三次関数である。この関数は、概して、簡単な予測関数を使用してタイル全体にわたって深さを予測するのを難しくする。その結果、標準的な深さバッファ圧縮技術、特に正確な平面方程式を利用するものは、そのような雑音の多いバッファを圧縮できない可能性がある。

#### 【0010】

タイルと呼ばれるピクセルのブロックは、独立に処理され得る。現在の深さ圧縮スキーマは、モーションブラー及びフィールドの深さに明示的には対処しておらず、したがって、これらのスキーマは時間コンポーネント及びレンズパラメータを有していない。10

#### 【0011】

一般に、深さ圧縮アルゴリズムは、3つの共通のステップを使用する。これらの3つの共通のステップは、クラスタ化と、予測関数生成と、残差符号化 (residual encoding) と呼ばれることがある。クラスタ化は、タイル内のサンプルのセットが異なるレイヤ、例えば背面レイヤと前面レイヤとに属するときに使用される。この場合、同じ予測関数を使用してタイル内の全て深さを圧縮するのは困難である。

#### 【0012】

クラスタ化ステップは、タイルのサンプルを2つ又はそれ以上のレイヤに分けることを試みる。この場合、各レイヤ内のサンプルは典型的に、共通の平面内にあること等の何らかの特性を共有する。サンプルを2つ又は複数のレイヤに分割する目標は、理想的には、全てのサンプルを单一のレイヤ内に圧縮することに比べて圧縮がより簡単になることがある。20

#### 【0013】

予測関数生成では、各レイヤが、そのレイヤ自身の予測関数を生成する。目標は、深さサンプル及び可能であればそのサンプルの固定座標を使用して、安価な関数を用いて各サンプルの深さを予測する予測関数を作成することである。例えばピクセルごとに小さな変位を有する長方形がタイルにレンダリングされていると想定する。これはおそらく変位した深さが存在することになる場所について良好な推測であるので、予測関数として、その長方形の平面を使用してもよい。その推測が完全ではないとしても、不完全性を次のステップにおいて対処することができる。30

#### 【0014】

グラフィクスプロセッサの共通の要件は、深さバッファが非損失的であることなので、残差符号化は、タイルの圧縮解除の間ににより正確な深さを再構築するのを可能にする。残差は、予測関数と実際のサンプルの深さとの間の差である。良好な予測関数が与えられると、サンプルの深さとこの予測関数との残差は小さい。結果として、これらの残差を、比較的少ないビットを使用して符号化することができる。そして(レイヤの数が少ない場合)良好な圧縮比を達成することができ、予測関数に必要なストレージは小さい。

#### 【0015】

一部の実施形態において、深さバッファ圧縮は、アンカーポイントに基づく符号化を使用することがあるが、平面符号化を含め、他の技法が使用されることもある。平面データの導関数又はデルタベクトルに基づく平面方程式及び平面符号化の構築は、実際の導関数が、これらの導関数を記憶するために割り当てられたビット量に比べて非常に大きいという意味で、極端な平面 (extreme plane) に対処することができない。したがって、実際の平面表現の正確性は低下する可能性がある。この正確性の低下は、タイル内のデルタベクトルが評価されたポイントからの距離とともに増す。他方、デルタベクトルに多過ぎるビットをネイティブに割り当てるにより、深さ値の残差を格納するのに利用可能なビットの数が低減する。

#### 【0016】

したがって、一部の実施形態において、利用可能なビットをデルタベクトルに静的に割40

50

り当てる影響は、深さデータの性質に応じてビットの動的な割り当てを可能にすることによって低減され得る。

#### 【0017】

ラスタ化されたジオメトリに応じて、タイルは、多数の基本要素のうちのターゲットとすることができます、その結果、時間とともに深さ値の複雑な分散がもたらされる。1つ又は複数の平面方程式をこれらの深さ値の基本予測として使用して、残差値を、圧縮された深さデータフォーマットで符号化する。予測平面の性質の結果として、予測される深さ値は、より正しいことも、あまり正しくないこともあります、その結果、残差値の訂正ビットに対する必要性が変化することとなる。

#### 【0018】

現在の深さ圧縮機構は、全てのタイル位置に対して等しい数の残差ビットを静的に割り当てるが、実際の深さ値とあまり一致しない可能性がある。一部の実施形態において、利用可能なビットを、残差訂正ビットに静的に割り当てる影響は、この静的な割り当ての代わりに、個々のタイル位置に対して残差ビットの数を動的に割り当てる可能にすることによって低減される。

#### 【0019】

アンカーポイントに基づく符号化の実施形態において、深さデータの圧縮機構は、タイル内の1つ又は複数のポイントを候補アンカーポイントとして選ぶことによって、ラスタ化された基本要素の平面表現を検出する。アンカーポイントの評価は、深さデータ値に基づく、ローカルの平面的なX及びY導関数の計算を含む。予測平面表現 ( $z = p(x, y) = a + b * x + c * y$ ) は、圧縮された深さデータにおいて3つの値： $a$ 、 $b$  及び  $c$  として符号化される。ここで、 $a$  はアンカーポイントにおける深さ値であり、 $b$  は  $dZ/dX$  であり、 $c$  は  $dZ/dY$  である。値  $b$  及び値  $c$  は、デルタ値、すなわち「ベクトル」である。

#### 【0020】

タイル内の各深さ  $z(x, y)$  について、残差深さ値が、予測した深さ値と実際の深さ値との差として格納される。値  $d$  は残差値である。

#### 【0021】

デルタ値及び残差値に割り当てるビット数を選択することによって、タイルの深さデータに対する合計ビット量を、深さ値の正確性を損なうことなく低減することができる。一部の実施形態において、深さデータの最終的な圧縮 (net compression) が達成され得る。

#### 【0022】

デルタモードの概念を紹介する。デルタモードは、X、Y及びRの特定の組み合わせを指定する。ここで、XはXベクトルのデルタ値のビット数であり、YはYベクトルのデルタ値のビット数であり、Rは深さごとの残差ビットの数である。圧縮において、これらのモードは、タイル内の各アンカーポイントの選択に利用可能である。

#### 【0023】

ある変形において、各アンカーポイントが、該アンカーポイント自身の利用可能なモードのセットを有していることがあります、これらのモードは、圧縮の間に動的に作成されるか、圧縮の結果によって動的にガイドされる。

#### 【0024】

別の変形では、符号化を簡潔にするために、両方向 (XとY) に同じビット数を使用することができる。以下では、圧縮を開始する前に、静的に作成されて格納されるモードテーブルを検討する。デルタモードの例示的なセットである表1を参照されたい。この例では、B = 5 1 2、N = 4、M = 3、A = 3 2、T = 3 2とする。

#### 【0025】

【表1】

モード	X	Y	R
1	3	3	14
2	6	6	13
3	9	9	11
4	8	10	11
5	10	8	11
6	13	13	10
7	11	17	9
8	17	11	9

表1. 例示的なデルタモードのテーブル

## 【0026】

圧縮についてタイルの次元とビット量が与えられるとき、デルタ値と残差値に利用可能なビット数の間にトレードオフが存在する。必ずしもX、Y及びRの全ての組み合わせが関連するわけではない。Rが少なくとも1ビット増加した場合に、X及び/又はYが低減することにのみ意味がある。これは、利用可能なモードの数を制限する。

10

## 【0027】

以下の条件式は、タイルの圧縮を可能にするために満たす必要がある条件である。

$$T * \log_2(N) + N * (M + A + X + Y) + (T - N * 3) * R < B$$

ここで、

B = 圧縮を可能にする合計ビット量

N = アンカーポイントの数

M = アンカーポイントについて使用されるモードを表す $\log_2$ (モード数)ビット

A = アンカーポイントの深さ値に使用されるビット数

X = Xベクトルのデルタ値に使用されるビット数

Y = Yベクトルのデルタ値に使用されるビット数

30

T = タイル内の深さの合計数

R = 残差深さ値に使用されるビット数

## 【0028】

項 $T * \log_2(N)$ は、深さがどの平面方程式を使用しているかを示すために、タイル内の深さごとに $\log_2(N)$ ビットを有する。項 $(T - N * 3)$ は、アンカーポイントの深さ並びに一次導関数( $dZ/dX$ 及び $dZ/dY$ )が正確であること、すなわち、これらが残差ビットのストレージを要しないことを暗示する。別の実施形態では、アンカーポイントも一次導関数も正確でなく、これは、これらの深さに残差ヒットが必要であり、そして上記不等式の最後の項を $(T - N * 3)$ の代わりにTにする必要がある。アンカーポイントは正確であるが、導関数は正確でない場合、最後の項は代わりに $(T - N)$ になる。上記の例では、アンカ値並びに導関数は正確に、多くの場合妥当な想定である正しい深さ値に到達する。

40

## 【0029】

一実施形態によると、各アンカーポイントについてモードを選択するためのアルゴリズムは、次の通りである。

## 【0030】

2Mの行を有するテーブルを構築する。各方向(X及びY)について、割り当てられた複数のデルタビットを格納する。例えば表1を参照されたい。次いで、N個のアンカーポイントの位置が与えられる。 $N = 1, 2, 3, \dots, T$ (通常は1と4の間)である。次に、各アンカーポイントについて、デルタベクトルを表す必要な最小ビット数が計算さ

50

れる。これは、アンカーポイントの深さから  $dZ / dX$  の右（又は左）近傍まで、そして  $dZ / dY$  の上（又は下）近傍までのデルタベクトルを単に計算し、どのくらい多くのデルタビットが必要であるかを（最初の及び最上位セットビットを見つけることによって）検査することにより行われることがある。最大数の残差ビットを有する対応モードは、このアンカーポイントのモードとして選択される。

#### 【0031】

この簡単なスキーマの利点は、より多くのタイルが圧縮され、最終的に、深さについての（グラフィクスプロセッサにおける帯域幅の大きい消費である）メモリ帯域幅の利用を低減することが可能になることである。例えば一度に深さごとに数残差ビットしか必要としない、大きなデルタベクトルを有するタイルを圧縮することが可能である。同様に、多くの残差ビットを要する小さなデルタベクトルを伴うタイルも圧縮することができる。10

#### 【0032】

不十分な正確性及び／又は精度のために、そして、極端な平面表現又は複雑なタイルデータを簡単にするために、平面予測は、正しい深さ値に調整するために、タイル位置ごとに残差ビットを必要とする。予測における不正確性の多さに応じて、異なる数の残差ビットが、予測と正しい値との間の差を符号化するのに実際に必要とされる。

#### 【0033】

タイル位置ごとに 1 つのエントリを有し、どれくらい多くの残差ビットがその位置に必要とされるかのインジケータを格納する残差モードマスクの使用を提案する。これは、各位置についてできるだけ少ないビットを使用し、その結果として必要なビットの最小の合計が得られるのを助ける。20

#### 【0034】

各位置について使用される残差ビットの可能な数を、適切な値に静的に割り当てることができ、又はタイルデータに基づいて動的に計算することができる。

#### 【0035】

動的に割り当てられる残差モードでは、一実施形態において、アルゴリズムは以下のように機能する。

#### 【0036】

タイルの圧縮の間に、予測の深さを実際の深さと比較して、残差値を計算する。次いで最短モード（すなわち、必要とされる差をキャプチャする、最少の数のビットのモード）を選択して残差を符号化する。次に、対応するモード値が残差モードマスクに入れられる。最後に残差モードマスクが、符号化されたタイルデータに格納される。30

#### 【0037】

この技法を、例を用いて説明する。

#### 【0038】

#### 【表2】

モード	残差ビット
00=0	2
01=1	8
10=2	12
11=3	16

表2. 4つのモードを有する例示的な残差モードテーブル

#### 【0039】

【表3】

	X0	X1	X2	X3
Y0	00	00	01	10
Y1	00	01	11	00
Y2	00	10	01	00
Y3	01	00	00	00

表3. 対応する符号化を伴う $4 \times 4$ ピクセルの例示的なタイル

10

## 【0040】

この例において、残差のビットの合計数は90 ( $9 \times 2 + 4 \times 8 + 2 \times 12 + 16$ )である。これは、2ビットを使用するモード00のインスタンスが9つ存在し、他も同様に計算できるためである。全ての残差値の値を格納すると256 ( $16 \times 16$ )ビットになる。

## 【0041】

この例では、各深さ値に対して2ビット値を有する。しかしながら、各 $2 \times 2$ 深さに対して2ビット値を使用することを選んでもよい（使用される2ビット値の数を減らすため）。一般に、各W×H深さは、モードを示すのにQビットを使用することができる。

## 【0042】

静的な割り当てのために、タイルごとに、選択を行うための幾つかのモードを供給することができる。例えばあるモードは、全体的に統一的な方法で残差ビットを深さ全体に広げるモードとすることができます。別のモードは、アンカーポイントに近い少ないビットと、アンカーポイントから離れた多くのビットとを使用することがあり、深さはタイル内に位置する。

## 【0043】

アンカーポイントに基づく符号化ベースの圧縮技術を用いる実施形態を説明してきたが、動的な残差ビット割り当てを、残差を符号化する任意の圧縮器に使用することもできる。動的なデルタ値ビット割り当ては、他の予測器での使用も可能である。例えば（4つの深さ値に加えてそのX, Y位置から作成される）双線パッチを予測器として用いる場合、そのパッチを、1つのアンカーポイント及び2つのデルタベクトルとして、そし第4のポイントに対する残差値として符号化することができる。この第4のポイントに対する残差値は、最初の3つのポイントからの平面方程式と第4のポイントとの間の差である。これら2つのデルタベクトルに使用したビットの数と、第4のポイント用の残差ビットとを、アンカーポイントに基づく符号化について上述した方法と同様の方法で動的に割り当てることができる。同じことが他の予測機能にも容易に当てはまる。

30

## 【0044】

図1を参照すると、装置10は、ラスタライザ12及び一連のピクセルパイプライン14a～14nを含み得る。深さテストユニット16a～16nが、各ピクセルパイプラインに対して提供され得る。深さキャッシュ及び選別ユニット18は、深さ値の選別に使用され得る。圧縮がコンプレッサ20において行われ、圧縮解除がデコンポレッサ24において行われ得る。（表1に類似の）デルタモードテーブル22も提供され得る。最後に、ランダムアクセスメモリ26が圧縮及び圧縮解除エンジン用に提供され得る。

40

## 【0045】

図2を参照すると、アンカーポイントのためのモード選択シーケンス30が、ハードウェア、ソフトウェア及び/又はファームウェアで実装され得る。ソフトウェア及びファームウェアの実施形態では、このシーケンスは、磁気、光又は半導体メモリのような、非一時的なコンピュータ読取可能媒体に格納されるコンピュータ実行命令によって実装され得る。

## 【0046】

50

アンカーポイントのためのモード選択は、ブロック 32において示されるように、デルタビットについてのテーブルを構築することによって開始される。次いで、ブロック 34において、アンカーポイントの位置が与えられる。最後に、ブロック 36において、アンカーポイントごとにそれぞれデルタベクトルを表すビットが計算される。

#### 【0047】

動的な残差値ビット割り当てのためのシーケンス 38は、ソフトウェア、ファームウェア及び／又はハードウェアで実装され得る。ソフトウェア及びファームウェアの実施形態では、このシーケンスは、磁気、光又は半導体メモリのような、非一時的なコンピュータ読み取り可能媒体に格納されるコンピュータ実行命令によって実装され得る。

#### 【0048】

シーケンスは、ブロック 40において示されるように、予測される深さを実際の深さと比較することによって開始する。次いでブロック 42に示されるように、残差値が計算される。

#### 【0049】

次いで、残差値ビットを符号化する最短モードが選択される（ブロック 44）。モード値を、残差モードマスク内に入れる（ブロック 46）。最後に、ブロック 48に示されるように、残差モードマスクを、符号化されたタイルデータに格納する。

#### 【0050】

図 4 に示されるように、コンピュータシステム 130 は、ハードドライブ 134 と、バス 104 によってチップセットコアロジック 110 に結合される取外し可能媒体 136 とを含み得る。コンピュータシステムは、スマートフォン、タブレット又はモバイルインターネットデバイスのようなスマートモバイルデバイスを含め、任意のコンピュータシステムとすることができる。キーボード及びマウス 120、あるいは他の従来的なコンポーネントを、バス 108 を介してチップセットコアロジック 110 に結合してもよい。コアロジックは、バス 105 を介してグラフィクスプロセッサ 112 に結合し、一実施形態では、中央プロセッサに結合してもよい。グラフィクスプロセッサ 112 は、バス 106 によってフレームバッファ 114 にも結合され得る。フレームバッファ 114 は、バス 107 によってディスプレイ画面 118 に結合され得る。一実施形態において、グラフィクスプロセッサ 112 は、单一命令複数データ（S I M D）アーキテクチャを使用するマルチスレッドのマルチコアパラレルプロセッサとすることができる。

#### 【0051】

ソフトウェア実装の場合、適切なコードが、（139において示されるように）メインメモリ 132 又はグラフィクスプロセッサ内の任意の利用可能なメモリを含め、任意の適切な半導体、磁気又は光メモリに格納され得る。したがって、一実施形態において、図 2 及び図 3 のシーケンスを実行するコードは、一実施形態において、メモリ 132 のような非一時的機械又はコンピュータ読み取り可能媒体 130、及び／又はグラフィクスプロセッサ 112、及び／又は中央プロセッサ 100 内に格納され、プロセッサ 100 及び／又はグラフィクスプロセッサ 112 によって実行され得る。

#### 【0052】

図 2 及び図 3 はフローチャートである。一部の実施形態において、これらのフローチャートに示されるシーケンスは、ハードウェア、ソフトウェア又はファームウェアで実装され得る。ソフトウェアの実施形態では、半導体メモリ、磁気メモリ又は光メモリのような非一時的なコンピュータ読み取り可能媒体が命令を格納するのに使用され、プロセッサによって実行されて、図 2 及び図 3 に示されるシーケンスを実施し得る。

#### 【0053】

本明細書で説明されるグラフィクス処理の技法は、様々なハードウェアアーキテクチャにおいて実装されることがある。例えばグラフィクスの機能性をチップセット内に一体化してもよい。あるいは、別個のグラフィクスプロセッサを使用してもよい。更に別の実施形態として、グラフィクス機能を、マルチコアプロセッサを含め汎用目的のプロセッサによって実装してもよい。

10

20

30

40

50

## 【0054】

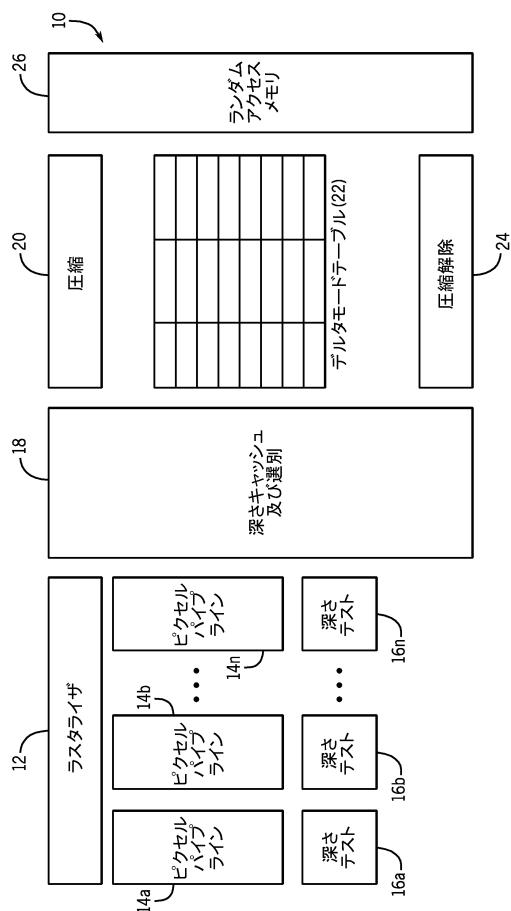
本明細書を通じて「一実施形態」又は「実施形態」への言及は、実施形態に関連して説明される特定の特徴、構造又は特性が、本発明に包含される少なくとも1つの実装に含まれることを意味する。したがって、「一実施形態」又は「実施形態において」というフレーズの使用は、必ずしも同じ実施形態を示すものではない。さらに、特定の特徴、構造又は特性は、説明された特定の実施形態以外の他の適切な形式でも実施されることもあり、そのような全ての形式は、本願の特許請求の範囲に含まれ得る。

## 【0055】

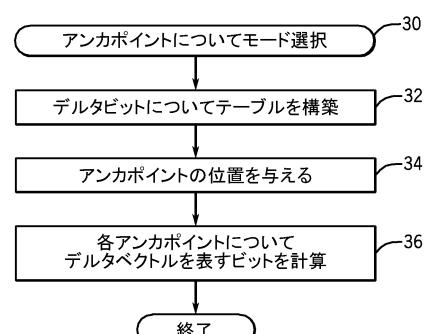
本発明は、限られた数の実施形態に関して説明されているが、当業者は、これの実施形態から様々な修正及び変形を理解するであろう。添付の特許請求の範囲は、本発明の精神及び範囲内にある限りにおいて、そのような変更及び変形を網羅するように意図される。

10

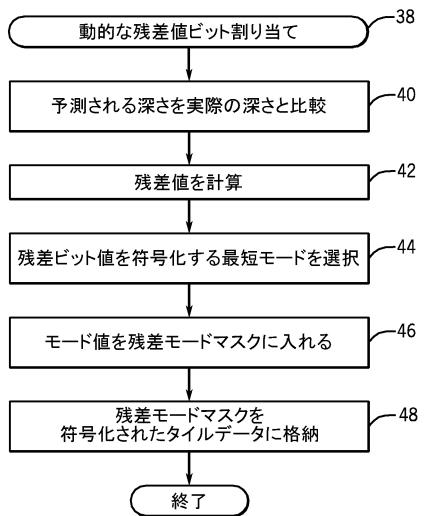
【図1】



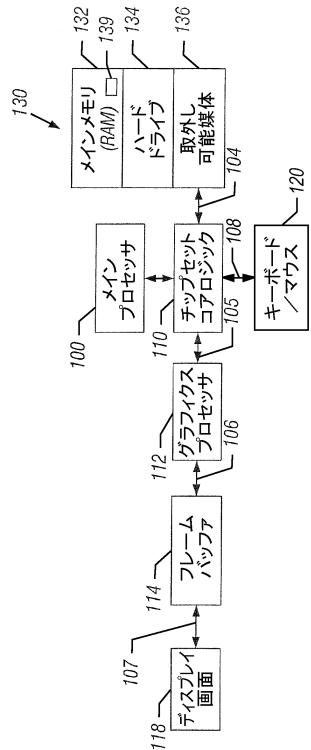
【図2】



【図3】



【図4】



---

フロントページの続き

(72)発明者 ニルソン , イーム コー .

スウェーデン国 エス - 2 2 6 3 9 エム . ルンド スヴェンスカ ヴァーゲン 4

(72)発明者 アンデルソン , マグヌス

スウェーデン国 エス - 2 5 3 7 5 エム ヘルシングボルグ カプリフォルガタン 3 7

(72)発明者 ハッセルグレン , ヨン エヌ .

スウェーデン国 エス - 2 1 8 3 2 エム ブルケルフロストランド スバルヴァーゲン 4

審査官 岩井 健二

(56)参考文献 特開2010-218548 (JP, A)

特開2009-151758 (JP, A)

特開2005-341555 (JP, A)

特開2002-152776 (JP, A)

特開平09-037246 (JP, A)

米国特許第6580427 (US, B1)

(58)調査した分野(Int.Cl., DB名)

H 04 N 19 / 00 - 19 / 98

G 06 T 15 / 00 - 15 / 87