

(19) 日本国特許庁(JP)

(12) 公表特許公報(A)

(11) 特許出願公表番号

特表2018-503919

(P2018-503919A)

(43) 公表日 平成30年2月8日(2018.2.8)

(51) Int.Cl. F I テーマコード (参考)
G 0 6 T 3/40 (2006.01) G 0 6 T 3/40 7 0 5 5 B 0 5 7

審査請求 未請求 予備審査請求 有 (全 34 頁)

(21) 出願番号	特願2017-539400 (P2017-539400)	(71) 出願人	595020643
(86) (22) 出願日	平成28年1月14日 (2016.1.14)		クゥアルコム・インコーポレイテッド
(85) 翻訳文提出日	平成29年9月20日 (2017.9.20)		QUALCOMM INCORPORATED
(86) 国際出願番号	PCT/US2016/013468		ED
(87) 国際公開番号	W02016/122896		アメリカ合衆国、カリフォルニア州 92
(87) 国際公開日	平成28年8月4日 (2016.8.4)		121-1714、サン・ディエゴ、モア
(31) 優先権主張番号	62/108,986		ハウス・ドライブ 5775
(32) 優先日	平成27年1月28日 (2015.1.28)	(74) 代理人	100108855
(33) 優先権主張国	米国 (US)		弁理士 蔵田 昌俊
(31) 優先権主張番号	14/836,449	(74) 代理人	100109830
(32) 優先日	平成27年8月26日 (2015.8.26)		弁理士 福原 淑弘
(33) 優先権主張国	米国 (US)	(74) 代理人	100158805
			弁理士 井関 守三
		(74) 代理人	100112807
			弁理士 岡田 貴志

最終頁に続く

(54) 【発明の名称】 ベイヤーマッピングを備えたグラフィックス処理ユニット

(57) 【要約】

グラフィックス処理のための方法および製造物であって、その中で、第1のラインの生ベイヤーデータおよび第2のラインの生ベイヤーデータが受け取られる。第1のラインの生ベイヤーデータおよび第2のラインの生ベイヤーデータの複数の重複していない 2×2 配列の各 2×2 配列は、複数のテクセルを提供するために、別個の対応するテクセルとしてマッピングされる。少なくとも1つの動作が、複数のテクセルのうちの少なくとも1つに対して実行される。

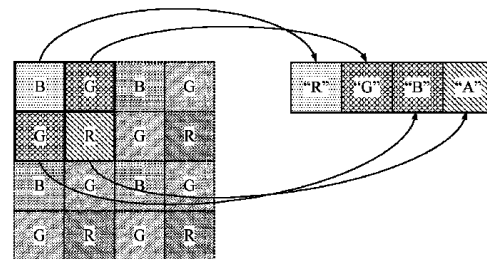


FIG. 9

【特許請求の範囲】**【請求項 1】**

グラフィックス処理のための方法であって、

第 1 のラインの生バイヤーデータおよび第 2 のラインの生バイヤーデータを受け取ることと、

複数のテクセルを提供するために、別個の対応するテクセルとして、前記第 1 のラインの生バイヤーデータおよび前記第 2 のラインの生バイヤーデータの複数の重複してない 2×2 配列の各 2×2 配列をマッピングすることと、

前記複数のテクセルのうちの少なくとも 1 つに対して少なくとも 1 つの動作を実行することと

を備える方法。

10

【請求項 2】

前記少なくとも 1 つの動作は、バイヤービニング動作、バイリニアフィルタ動作、読取り動作、または収集動作のうちの少なくとも 1 つを含む、請求項 1 に記載の方法。

【請求項 3】

前記複数のテクセルの各テクセルは、R G B A フォーマットである、請求項 1 に記載の方法。

【請求項 4】

前記複数のテクセルのうちの少なくとも 1 つに対して前記少なくとも 1 つの動作を実行した後、画像信号プロセッサに前記複数のテクセルを送信することをさらに備える、請求項 1 に記載の方法。

20

【請求項 5】

前記複数のテクセルを提供するために、別個の対応するテクセルとして、前記第 1 のラインの生バイヤーデータおよび前記第 2 のラインの生バイヤーデータの複数の重複してない 2×2 配列の各 2×2 配列を前記マッピングすることは、アプリケーションプログラムインタフェースを介してアクセス可能である、請求項 1 に記載の方法。

【請求項 6】

前記生バイヤーデータが M I P I フォーマットであるかどうかについての決定を行うことと、

前記生バイヤーデータが M I P I フォーマットであると決定された場合、前記生バイヤーデータをアンパックするために、グラフィックス処理ユニットのテクスチャパイプラインにおけるハードウェアを用いることと

をさらに備える、請求項 1 に記載の方法。

30

【請求項 7】

前記テクスチャパイプラインにおける前記ハードウェアによって実行される前記生バイヤーデータを前記アンパックすることは、対応するテクスチャデータタイプを有する、請求項 6 に記載の方法。

【請求項 8】

前記テクスチャパイプラインにおける前記ハードウェアによって実行される前記生バイヤーデータを前記アンパックすることは、アプリケーションプログラムインタフェースを介してアクセス可能である、請求項 6 に記載の方法。

40

【請求項 9】

グラフィックス処理のためのデバイスであって、

キャッシュと、

前記キャッシュから第 1 のラインの生バイヤーデータ、および前記キャッシュから第 2 のラインの生バイヤーデータを受け取り、

複数のテクセルを提供するために、別個の対応するテクセルとして、前記第 1 のラインの生バイヤーデータおよび前記第 2 のラインの生バイヤーデータの複数の重複してない 2×2 配列の各 2×2 配列をマッピングし、

前記複数のテクセルのうちの少なくとも 1 つに対して少なくとも 1 つの動作を実行す

50

る

ように構成されたテクスチャパイプラインと
を備えるデバイス。

【請求項 10】

前記テクスチャパイプラインは、前記少なくとも 1 つの動作が、ベイヤープニング動作、バイリニアフィルタ動作、読取り動作、または収集動作のうちの少なくとも 1 つを含むように構成される、請求項 9 に記載のデバイス。

【請求項 11】

前記テクスチャパイプラインは、前記複数のテクセルの各テクセルが、R G B A フォーマットであるように構成される、請求項 9 に記載のデバイス。

10

【請求項 12】

画像信号プロセッサをさらに備え、ここにおいて、前記画像信号プロセッサは、テクスチャプロセッサが前記複数のテクセルのうちの少なくとも 1 つに対して前記少なくとも 1 つの動作を実行した後に、前記複数のテクセルを受け取るように構成される、請求項 9 に記載のデバイス。

【請求項 13】

前記テクスチャパイプラインは、アプリケーションプログラムインタフェースを介してアクセス可能である手法で、前記複数のテクセルを提供するために、別個の対応するテクセルとして、前記第 1 のラインの生ベイヤータおよび前記第 2 のラインの生ベイヤータの複数の重複してない 2×2 配列の各 2×2 配列をマッピングするように構成される、請求項 9 に記載のデバイス。

20

【請求項 14】

前記テクスチャパイプラインは、
前記生ベイヤータが M I P I フォーマットであるかどうかについての決定を行い、
前記生ベイヤータが M I P I フォーマットであると決定された場合、前記生ベイヤータをアンパックする
ようにさらに構成される、請求項 9 に記載のデバイス。

【請求項 15】

前記テクスチャパイプラインは、前記アンパックすることが、対応するテクスチャデータタイプを有するように、前記生ベイヤータをアンパックすることが達成されるように構成される、請求項 14 に記載のデバイス。

30

【請求項 16】

前記テクスチャパイプラインは、前記生ベイヤータをアンパックすることがアプリケーションプログラムインタフェースを介してアクセス可能であるように構成される、請求項 14 に記載のデバイス。

【請求項 17】

前記キャッシュは、レベル 2 キャッシュであり、前記テクスチャパイプラインは、ローカルキャッシュと、
前記レベル 2 キャッシュから前記第 1 のラインの生ベイヤータ、および前記レベル 2 キャッシュから前記第 2 のラインの生ベイヤータを受け取り、前記複数のテクセルを提供するために、別個の対応するテクセルとして、前記第 1 のラインの生ベイヤータおよび前記第 2 のラインの生ベイヤータの前記複数の重複してない 2×2 配列の各 2×2 配列をマッピングし、前記ローカルキャッシュに前記複数のテクセルを送信するように構成されたアンパッキング回路と
を含む、請求項 9 に記載のデバイス。

40

【請求項 18】

前記アンパッキング回路は、
前記生ベイヤータが M I P I フォーマットであるかどうかについての決定を行い、
前記生ベイヤータが M I P I フォーマットであると決定された場合、前記生ベイヤータをアンパックする

50

ようにさらに構成される、請求項 17 に記載のデバイス。

【請求項 19】

前記アンパッキング回路は、別個のキャッシュを含む、請求項 17 に記載のデバイス。

【請求項 20】

グラフィックス処理のためのデバイスであって、

第 1 のラインの生ベイヤードータおよび第 2 のラインの生ベイヤードータを受け取るための手段と、

複数のテクセルを提供するために、別個の対応するテクセルとして、前記第 1 のラインの生ベイヤードータおよび前記第 2 のラインの生ベイヤードータの複数の重複していない 2×2 配列の各 2×2 配列をマッピングするための手段と、

前記複数のテクセルのうちの少なくとも 1 つに対して少なくとも 1 つの動作を実行するための手段と

を備えるデバイス。

【請求項 21】

前記少なくとも 1 つの動作は、ベイヤービニング動作、バイリニアフィルタ動作、読取り動作、または収集動作のうちの少なくとも 1 つを含む、請求項 20 に記載のデバイス。

【請求項 22】

前記複数のテクセルの各テクセルは、R G B A フォーマットである、請求項 20 に記載のデバイス。

【請求項 23】

前記複数のテクセルのうちの少なくとも 1 つに対して前記少なくとも 1 つの動作を実行した後、画像信号プロセッサに前記複数のテクセルを送信するための手段をさらに備える、請求項 20 に記載のデバイス。

【請求項 24】

前記複数のテクセルを提供するために、別個の対応するテクセルとして、前記第 1 のラインの生ベイヤードータおよび前記第 2 のラインの生ベイヤードータの複数の重複していない 2×2 配列の各 2×2 配列を前記マッピングすることは、アプリケーションプログラムインタフェースを介してアクセス可能である、請求項 20 に記載のデバイス。

【請求項 25】

前記生ベイヤードータが M I P I フォーマットであるかどうかについての決定を行うための手段と、

前記生ベイヤードータが M I P I フォーマットであると決定された場合、前記生ベイヤードータをアンパックするために、グラフィックス処理ユニットのテクスチャパイプラインにおけるハードウェアを用いるための手段と

をさらに備える、請求項 20 に記載のデバイス。

【請求項 26】

前記テクスチャパイプラインにおける前記ハードウェアによって実行される前記生ベイヤードータを前記アンパックすることは、対応するテクスチャデータタイプを有する、請求項 25 に記載のデバイス。

【請求項 27】

前記テクスチャパイプラインにおける前記ハードウェアによって実行される前記生ベイヤードータを前記アンパックすることは、アプリケーションプログラムインタフェースを介してアクセス可能である、請求項 25 に記載のデバイス。

【発明の詳細な説明】

【関連出願の説明】

【0001】

[0001]本願は、その内容全体が参照により本明細書に組み込まれる、2015年1月28日に提出された米国仮特許出願第62/108,986号の利益を主張する。

【技術分野】

【0002】

10

20

30

40

50

[0002]本開示は、グラフィックス処理に関し、具体的には、グラフィックス処理ユニット（GPU）による生画像データ（raw image data）の処理に関する。

【背景技術】

【0003】

[0003]デジタルカメラによって使用されるCMOS画像センサの出力は、典型的に、ベイヤー生データ（Bayer raw data）である。ベイヤー生データは、典型的に、50%緑、25%赤、および25%青であるフィルタパターンを含むベイヤーフィルタからキャプチャされた、さらなる処理なしの、生画像データである。ベイヤーデータは、データがデモザイキング（demosaicing）のようなさらなる処理以前にセンサによって提供されるフォーマットであるという点で、「生（raw）」である。画像センサは、典型的に、任意の所
10
与のロケーション（location）における1色のみを感知する - 上記で説明されたように、ベイヤーフィルタは、典型的に、50%緑センサ、25%赤センサ、および25%青センサであるフィルタパターンを使用する。画像が表示される前に、それは典型的に、それによってフルカラー画像が各ロケーションにおいて1色のみをキャプチャするセンサからの画像から再構成される、デモザイキングのような様々な処理を行う。

【0004】

[0004]ベイヤー生画像データはまた、MIPフォーマットであり得、これは、各行の終わりまでパディングのない、6 - 14ビットの深度における生センサデータの標準化された密なパッキングである（ここで、「各行（each row）」は、画像データの各行を意味し、ここで、画像データは、行および列において存在する2次元配列である）。MIP
20
フォーマットのデータは、よりコンパクトであるが、データ構造アラインメントの目的のために通常使用されるパディングを欠く。「データ構造アラインメント」は、メモリにおける特定のデータ構造がどこで（when）開始および終了するかを決定することが可能になるように、データがメモリにおいてアクセスおよび配置（arranged）される方法を指す。標準データフォーマットは、データ構造アラインメントを確実にするためにパディングを使用する。MIPフォーマットにおけるデータは、各行の終わりまでパディングを有しないので、MIPフォーマットデータのストライド長は、予め決定されなければならない（例えば、アラインされた128または256ビット）。

【発明の概要】

【0005】

[0005]いくつかの例では、グラフィックス処理のためのデバイスが、キャッシュと、キャッシュから第1のラインの生ベイヤーデータ、およびキャッシュから第2のラインの生ベイヤーデータを受け取り、複数のテクセル（texels）を提供するために、別個の対応するテクセルとして、第1のラインの生ベイヤーデータおよび第2のラインの生ベイヤーデータの複数の重複してない2×2配列の各2×2配列をマッピングし、複数のテクセルのうちの少なくとも1つに対して少なくとも1つの動作（operation）を実行するように構成（arranged）されたテクスチャパイプラインとを備える。

【0006】

[0006]いくつかの例では、グラフィックス処理のための方法が、第1のラインの生ベイヤーデータおよび第2のラインの生ベイヤーデータを受け取ることと、複数のテクセルを
40
提供するために、別個の対応するテクセルとして、第1のラインの生ベイヤーデータおよび第2のラインの生ベイヤーデータの複数の重複してない2×2配列の各2×2配列をマッピングすることと、複数のテクセルのうちの少なくとも1つに対して少なくとも1つの動作を実行することとを備える。

【0007】

[0007]いくつかの例では、グラフィックス処理のためのデバイスが、第1のラインの生ベイヤーデータおよび第2のラインの生ベイヤーデータを受け取るための手段と、複数のテクセルを提供するために、別個の対応するテクセルとして、第1のラインの生ベイヤーデータおよび第2のラインの生ベイヤーデータの複数の重複してない2×2配列の各2×2配列をマッピングするための手段と、複数のテクセルのうちの少なくとも1つに対して
50

少なくとも 1 つの動作を実行するための手段とを備える。

【 0 0 0 8 】

【0008】本開示の 1 つまたは複数の態様の詳細は、添付の図面および以下の説明において示される。本開示で説明される技法の他の特徴、目的、および利点は、説明および図面から、および特許請求の範囲から明らかになるであろう。

【図面の簡単な説明】

【 0 0 0 9 】

【図 1】【0009】図 1 は、カメラセンサを示す概念図である。

【図 2】【0010】図 2 は、生センサデータの処理を示す概念図である。

【図 3】【0011】図 3 は、M I P I 1 0、M I P I 1 2、および M I P I 1 4 フォーマットにおけるビットパッキングの例を示す概念図である。

【図 4】【0012】図 4 は、バイヤーデータのための実例的なレイアウトを示す概念図である。

【図 5】【0013】図 5 は、本開示の技法による実例的なデバイスを例示するブロック図である。

【図 6】【0014】図 6 は、本開示の技法による実例的なプロセスを例示するフロー図である。

【図 7】【0015】図 7 は、本開示の技法をインプリメントし得る実例的なシステムを例示するブロック図である。

【図 8】【0016】図 8 は、本開示の技法をインプリメントし得る図 7 のシステムの例を例示するブロック図である。

【図 9】【0017】図 9 は、本開示の技法による図 6 のプロセスの動作 (acts) のうちの 1 つの例に従って、各バイヤーパターンをテクセルにマッピングする例を例示する概念図である。

【図 1 0】【0018】図 1 0 は、本開示の技法による図 9 のマッピングされたテクセルをアドレス指定 (addressing) するためのプロセスを例示する概念図である。

【図 1 1】【0019】図 1 1 は、本開示の技法による図 6 のプロセスの動作 (acts) のうちの 1 つの例に従って、テクセルに対して収集動作を実行する例を例示する概念図である。

【図 1 2】【0020】図 1 2 は、本開示の技法による図 8 のシステムの一部の例を例示するブロック図である。

【図 1 3】【0021】図 1 3 は、本開示の技法による、図 7 のデバイスの例を示す機能ブロック図である。

【発明の詳細な説明】

【 0 0 1 0 】

【0022】いくつかの例では、グラフィックス処理のためのデバイスが、キャッシュおよびテクスチャパイプラインを備える。(例えば、GPUの)テクスチャパイプラインは、キャッシュから第 1 のラインの生バイヤーデータ、およびキャッシュから第 2 のラインの生バイヤーデータを受け取るように構成される。テクスチャパイプラインは、複数のテクセルを提供するために、別個の対応するテクセルとして、第 1 のラインの生バイヤーデータおよび第 2 のラインの生バイヤーデータの複数の重複してない 2×2 配列の各 2×2 配列をマッピングするようにさらに構成される。「 2×2 (two by two)」は、 2×2 配列が 1 つのラインにおける 2 つの隣接する画素 (pixels) (例えば、第 1 のラインにおける 2 つの隣接する画素) および隣接するラインにおける 2 つの隣接する画素 (例えば、第 2 のラインにおける 2 つの隣接する画素) を含むように、2 次元の画像データを指す。 2×2 パターンは、例えば、このような 2×2 配列のテクセルを例示する以下の図 9 - 図 1 1 に関して、以下でより詳細に説明される。

【 0 0 1 1 】

【0023】別個の対応するテクセルとして、第 1 のラインの生バイヤーデータおよび第 2 のラインの生バイヤーデータの複数の重複してない 2×2 配列の各 2×2 配列をマッピングすることは、マッピング後に、各個別の画素が、データに対して実行される動作の基本単

10

20

30

40

50

位として扱われるのとは対照的に、 2×2 マトリクスにおける4つの画素の各まとまり (bundle) が、データに対して実行される動作の基本単位として扱われるように、生バイヤーデータの各 2×2 配列における4つの画素がひとまとめにされる (bundled) ことを意味する。複数のテクセルを提供するために、別個の対応するテクセルとして、第1のラインの生バイヤーデータおよび第2のラインの生バイヤーデータの複数の重複していない 2×2 配列の各 2×2 配列をマッピングすることは、定義上、テクセルに 2×2 配列をマッピングすることのより短い参照手段として「バイヤースウィズリング (Bayer swizzling)」と呼ばれる。(バイヤースウィズルの具体例が、以下の図9に例示され、図9に関してより詳細に説明される。) テクスチャパイプラインは、複数のテクセルのうちの少なくとも1つに対して少なくとも1つの動作を実行するようにさらに構成される。

10

【0012】

[0024] テクスチャパイプラインは、より詳細に説明されるように、例えば、位相検出オートフォーカス (PDAF) センサ、赤青緑クリア (RGBC) センサ、インターリーブドハイダイナミックレンジ (iHDR) センサ、コンピュータショナルフォトグラフィ (computational photography)、およびライトフィールドコンピューション (light field computation) などの、ある特定のカスタムセンサアプリケーションのためのある特定の前処理タスクを遂行するために、これら動作を実行する。バイヤースウィズリングは、このようなカスタムセンサアプリケーションのために実行されるある特定の動作が、個々の画素のレベルではなく、各々が画素の 2×2 配列に対応するテクセルのレベルで実行されるので、これら動作が、より一層効率的になることを可能にする。バイヤースウィズリング後は、アクセスされる各テクセルが4つのバイヤー画素を含むので、読取り動作は、さもなければ4回の読取りを必要とするであろうものを、1回の読取りで遂行することができる。同様の理由で、バイヤースウィズリング後のこれらテクセルに対する収集動作は、さもなければ4回のロードを必要とするであろうものを、1回のロードで遂行することができる。

20

【0013】

[0025] さらに、バイヤースウィズリング後、これらテクセルは、以下でより詳細に説明される理由から、テクスチャパイプラインにおけるバイリニアフィルタリング (bilinear filtering) の効率的なパフォーマンスによく適している (バイリニアフィルタリングは、ここでの後続するセクションにおいて説明される)。このようにして、バイヤースウィズリング後、バイヤービニング (Bayer Binning) を含むいくつかのアプリケーションは、テクスチャパイプラインのバイリニアフィルタリング能力を使用して、テクスチャパイプラインにおいて非常に効率的に実行されることができる。追加として、バイヤースウィズリングは、グラフィックス処理ユニット (GPU) におけるシェダプロセッサ上の負荷が低減されるように、バイヤーデータをひとまとめにし、ここで、テクスチャパイプラインおよびシェダプロセッサは、いくつかの例では、両方ともGPUに含まれる。シェダプロセッサは、以下でより詳細に説明される。テクスチャパイプラインがテクセルに対して実行する動作の具体例およびこのような実例的な動作のさらなる詳細が、ここでの後のセクションにおいて詳細に説明される。

30

【0014】

[0026] テクスチャパイプラインはまた、生MIP I データをアンパックする (unpack) ために使用されることができる。(例えば、位相検出オートフォーカス (PDAF) センサ、赤青緑クリア (RGBC) センサ、インターリーブドハイダイナミックレンジ (iHDR) センサ、コンピュータショナルフォトグラフィ、およびライトフィールドコンピューションを含む) ある特定のカスタムセンサアプリケーションにとっては、これらカスタムアプリケーションによって必要とされる動作を実行する前にアンパックされたMIP I データへのアクセスを有することが有利であり得る。テクスチャパイプラインにおいてMIP I データをアンパックすることは、MIP I データをアンパックする従来の方法に対して、著しい速度の優位性を有することができる。

40

【0015】

50

[0027]上記で説明されたように、M I P I データは、密にパックされている - 各行の終りまで M I P I データにおけるパディングがない。M I P I データが標準のパディング (normal padding) を欠いているので、データ構造アラインメントは困難であり、M I P I データのストライド長は、予め決定されなければならない。M I P I データを「アンパックする」ことは、データが、予め決定されたストライド長を必要とせずに、プロセッサによるアクセスと適正にアラインメントされることができるよう、十分なデータ構造パディングを有するフォーマットに M I P I データを変換することを指す。ほとんどの標準的な (most standard) データフォーマットは、「アンパック」されている。アンパッキングを必要とするのは、(標準的な量のデータ構造パディングを欠く) M I P I のような密にパックされたデータフォーマットのみである。本開示によるテクスチャパイプラインの例によって実行される M I P I アンパッキングについてのさらなる詳細は、ここでの後のセクションにおいて説明される。

10

【 0 0 1 6 】

[0028]本説明のこの段階では、本開示のある特定の態様がより良く理解され得るように、ベイヤーデータおよび M I P I データなどの、様々なタイプの生データフォーマットをさらに説明することが有用であり得る。

【 0 0 1 7 】

[0029]このセクションにおける説明は、ここで説明される画像データフォーマット (例えば、ベイヤーおよび M I P I) のより詳細な説明を与える。デジタルカメラは、レンズと、光を検出し、カラー画素値 (例えば、R G B 値) を生成するように構成されたカメラセンサとを含み得る。図 1 は、カメラセンサの結果として得られたカラー画素値の例を示し、ここで、これらカラー画素値は、センサによる可視光の検出によって作成される。センサによって生成される、結果として得られたカラー画素値は、しばしば生センサデータと呼ばれる。以下でより詳細に説明されるように、生センサデータは、典型的に、ベイヤーフィルタモザイク画素フォーマットのような、密にパックされたフォーマットであり、または、M I P I データフォーマット (例えば、M I P I 1 0、M I P I 1 2、および M I P I 1 4) でパックされた、ベイヤーフィルタモザイク画素フォーマットを含む、生センサデータである。

20

【 0 0 1 8 】

[0030]カメラは、画像信号プロセッサをさらに含み得る。いくつかの例では、画像信号プロセッサは、レンズおよびカメラセンサと同じパッケージ中に共に含まれる。他の例では、画像信号プロセッサは、レンズおよびカメラセンサとは別個にパッケージ化され得る。いくつかの例では、画像信号プロセッサは、グラフィックス処理ユニット (G P U) を含むシステムオンアチップ (S o C) 上に含まれ得る。画像信号プロセッサは、生センサデータを受け取り、生センサデータを圧縮されたデータフォーマット (例えば、J P E G ファイル) に変換し、ピクチャファイルにおいて、結果として得られた圧縮されたデータを記憶するように構成され得る。他の例では、画像信号プロセッサは、生センサデータを保持し、別個のファイルにおいて生センサデータを保存するように構成され得る。図 2 は、画像プロセッサを備えた実例的なカメラシステムを示す。

30

【 0 0 1 9 】

[0031]しばしば、密にパックされたフォーマットにおける生センサデータは、(例えば、画像信号プロセッサによって) 追加の処理が実行される前にアンパックされる。生ベイヤーフィルタモザイクセンサデータは、典型的に、1 画素当たり 1 0 または 1 2 ビットであり、密にパックされている (すなわち、各行の終りまで、いかなるパディングもなしにパックされる)。

40

【 0 0 2 0 】

[0032]ここで、M I P I データフォーマットが説明される。M I P I アライアンスは、モバイル業界向けのインタフェース仕様を開発する団体である。M I P I アライアンスによって開発された実例的な仕様は、カメラシリアルインタフェース (C S I) 仕様およびディスプレイシリアルインタフェース (D S I) 仕様を含む。C S I 仕様は、カメラセン

50

サと画像信号プロセッサの間のインタフェースのためのプロトコルを指定する。C S Iの一般的に使用されるバージョンが、C S I - 2である。D S I仕様は、ディスプレイプロセッサとディスプレイパネルの間のインタフェースのためのプロトコルを指定する。

【 0 0 2 1 】

[0033] M I P Iフォーマットは、6 - 14ビットの深度の生センサデータの密なパッキングを規格化する (standardize)。生センサデータのためのM I P Iフォーマットは、10ビット、12ビット、および14ビットのフォーマットを含む。図3は、M I P I 10、M I P I 12、およびM I P I 14フォーマットにおけるビットパッキングの例を示す。M I P Iフォーマットの各々について、各行の画素データは、行の終わりにおけるパディングとともに、密にパックされている。

10

【 0 0 2 2 】

[0034] 10ビットのフォーマットを使用する1つの例が、R A W 10フォーマット、またはM I P I 10と呼ばれる。図3に示されるように、M I P I 10フォーマットでは、4画素 (40ビット) ごとに、5バイトのチャンク (chunks) にパックされる。各画素の8つの最上位ビット (M S B) が最初にパックされ、その後、2つの最下位ビット (L S B) が、最終バイトに連結される。パケットデータサイズは、5バイトにおける4画素である (4 pixels in 5 bytes)。

【 0 0 2 3 】

[0035] 12ビットのフォーマットを使用する1つの例が、R A W 12フォーマット、またはM I P I 12と呼ばれる。図3に示されるように、M I P I 12フォーマットでは、2画素 (24ビット) ごとに、3バイトのチャンクにパックされる。各画素の8つの最上位ビット (M S B) が最初にパックされ、その後、4つの最下位ビット (L S B) が、最終バイトに連結される。パケットデータサイズは、3バイトにおける2画素である。

20

【 0 0 2 4 】

[0036] 14ビットのフォーマットを使用する1つの例が、R A W 14フォーマット、またはM I P I 14と呼ばれる。図3に示されるように、M I P I 14フォーマットでは、4画素 (56ビット) ごとに、7バイトのチャンクにパックされる。各画素の8つの最上位ビット (M S B) が最初にパックされ、その後、6つの最下位ビット (L S B) が、最終バイトに連結される。パケットデータサイズは、7バイトにおける4画素である。

【 0 0 2 5 】

30

[0037] 図4は、M I P Iフォーマットのための実例的なレイアウトを示す。M I P I 10フォーマットでは、アドレス指定の計算 (addressing calculation) は、次のように実行される。各行またはスキャンライン (例えば、「B G B G B G . . .」または「G R G R G R . . .」) は、終わりにおいてパディングされる。したがって、ストライドは、予め決定されなければならない (例えば、アラインされた128または256ビット)。各行内では、画素 i について: $(i / 4) * 5 = 5$ バイトセグメントのバイトアドレス。

【 0 0 2 6 】

[0038] この計算は、加算器および組合せ論理回路を使用してインプリメントされ得る。しかしながら、計算を実行することの複雑さは、5バイトの境界をまたぐ (cross) 2画素が必要とされる場合に増大される。その場合には、2つの5バイトのチャンクがフェッチされなければならない。

40

【 0 0 2 7 】

[0039] 関連のある画像データフォーマットが説明されたので、本開示によるデバイスおよび方法の例の大まかな説明 (high-level description) が、ここで説明される。このセクションの意図は、実例的なデバイスおよび方法の大まかな説明を与えることであるので、実例的なデバイスおよび方法が含まれ得る特定の環境の例および図5のデバイス501内の特定の实例的な要素のさらなる詳細、ならびに実例的なデバイスおよび方法によって達成される利益は、このセクションでは説明されず、その代わりに、ここでの開示の他のセクションにおいて説明される。

【 0 0 2 8 】

50

【0040】図5は、グラフィックス処理のためのデバイス501の例を例示するブロック図である。デバイス501は、キャッシュ530およびテクスチャパイプライン523を含む。テクスチャパイプライン523は、キャッシュ530から第1のラインの生ベイヤータ、およびキャッシュ530から第2のラインの生ベイヤータを受け取るように構成される。テクスチャパイプライン523は、複数のテクセルを提供するために、別個の対応するテクセルとして、第1のラインの生ベイヤータおよび第2のラインの生ベイヤータの複数の重複してない 2×2 配列の各 2×2 配列をマッピングするようにさらに構成される。テクスチャパイプライン523は、複数のテクセルのうちの少なくとも1つに対して少なくとも1つの動作を実行するようにさらに構成される。

【0029】

【0041】いくつかの例では、テクスチャパイプライン523は、GPUの一部である。いくつかの例では、第1および第2のラインのデータは、画像センサによって生成された、隣接するラインの生画像データであり、「 2×2 」は、 2×2 配列が1つのラインにおける2つの隣接する画素（例えば、第1のラインにおける2つの隣接する画素）および隣接するラインにおける2つの隣接する画素（例えば、第2のラインにおける2つの隣接する画素）を含むように、2次元の画像データを指す。

【0030】

【0042】図6は、グラフィックス処理のためのプロセス630の例を例示するフロー図である。開始ブロック後、テクスチャパイプライン（例えば、図5のテクスチャパイプライン523）は、第1のラインの生ベイヤータおよび第2のラインの生ベイヤータを受け取る（631）。いくつかの例では、第1および第2のラインのデータは、画像センサによって生成された、隣接するラインの生画像データである。次に、テクスチャパイプラインは、複数のテクセルを提供するために、別個の対応するテクセルとして、第1のラインの生ベイヤータおよび第2のラインの生ベイヤータの複数の重複してない 2×2 配列の各 2×2 配列をマッピングする（632）。複数のテクセルを提供するために、別個の対応するテクセルとして、第1のラインの生ベイヤータおよび第2のラインの生ベイヤータの複数の重複してない 2×2 配列の各 2×2 配列をマッピングするために、テクスチャパイプラインによって実行されるマッピングは、「ベイヤースウィズリング」としてここで定義される。ベイヤースウィズリング後、テクスチャパイプラインは、複数のテクセルのうちの少なくとも1つに対して少なくとも1つの動作を実行する（633）。その後、プロセスは、戻りブロックに進み、ここで、他の処理が再開される。

【0031】

【0043】図7は、本開示の1つまたは複数の態様をインプリメントするように構成され得る実例的なシステム/デバイスを例示するブロック図である。図7に示されるように、システム/デバイス2は、パーソナルコンピュータ、デスクトップコンピュータ、ラップトップコンピュータ、タブレットコンピュータ、コンピュータワークステーション、ビデオゲームプラットフォームまたはコンソール、例えば、セルラまたは衛星電話のようなモバイル電話、ランドライン電話、インターネット電話、ポータブルビデオゲームデバイスまたは携帯情報端末（PDA）のようなハンドヘルドデバイス、パーソナルミュージックプレーヤ、ビデオプレーヤ、ディスプレイデバイス、テレビジョン、テレビジョンセットトップボックス、サーバ、中間ネットワークデバイス、メインフレームコンピュータ、任意のモバイルデバイス、あるいはグラフィカルデータを処理するおよび/または表示するその他任意のタイプのデバイスであり得る。図7の例では、デバイス2は、中央処理ユニット（CPU）6、システムメモリ10、およびGPU12を含み得る。デバイス2はまた、ディスプレイプロセッサ14、トランシーバモジュール3、ユーザインタフェース4、およびディスプレイ8を含み得る。トランシーバモジュール3およびディスプレイプロセッサ14は、両方がCPU6および/またはGPU12と同じ集積回路（IC）の一部であり得るか、両方がCPU6および/またはGPU12を含むICまたは複数のICの外部にあり得るか、またはCPU6および/またはGPU12を含むICの外部にあるIC内に形成され得る。

10

20

30

40

50

【 0 0 3 2 】

[0044] デバイス 2 は、明確さを目的として図 7 に示されていない追加のモジュールまたはユニットを含み得る。例えば、デバイス 2 は、デバイス 2 がモバイルワイヤレス電話である例において電話通信を実施するために、そのいずれもが図 7 に示されていないスピーカおよびマイクロホン、またはデバイス 2 がメディアプレーヤである場合にはスピーカを含み得る。デバイス 2 はまた、ビデオカメラを含み得る。デバイス 2 はまた、画像信号プロセッサ (ISP) を含み得る。さらに、デバイス 2 において示される様々なモジュールおよびユニットは、デバイス 2 のすべての例において必ずしも必要ではない。例えば、ユーザインタフェース 4 およびディスプレイ 8 は、デバイス 2 が外部ユーザインタフェースまたはディスプレイとインタフェースするように装備されたデスクトップコンピュータまたは他のデバイスである例では、デバイス 2 の外部にあり得る。

10

【 0 0 3 3 】

[0045] ユーザインタフェース 4 の例は、それに限定されるものではないが、トラックボール、マウス、キーボード、および他のタイプの入力デバイスを含む。ユーザインタフェース 4 はまた、タッチスクリーンであり得、ディスプレイ 8 の一部として組み込まれ得る。トランシーバモジュール 3 は、デバイス 2 と別のデバイスまたはネットワークとの間のワイヤレスまたはワイヤード通信を可能にするための回路を含み得る。トランシーバモジュール 3 は、変調器、復調器、増幅器、およびワイヤードまたはワイヤレス通信のための他のそのような回路を含み得る。

20

【 0 0 3 4 】

[0046] CPU 6 は、実行のためにコンピュータプログラムの命令を処理するように構成された中央処理ユニット (CPU) のような、マイクロプロセッサであり得る。CPU 6 は、デバイス 2 の動作を制御する汎用または専用プロセッサを備え得る。ユーザは、CPU 6 に、1 つまたは複数のソフトウェアアプリケーションを実行することを行わせるために、デバイス 2 に入力を提供し得る。CPU 6 上で実行されるソフトウェアアプリケーションは、例えば、オペレーティングシステム、ワードプロセッサアプリケーション、電子メールアプリケーション、スプレッドシートアプリケーション、メディアプレーヤアプリケーション、ビデオゲームアプリケーション、グラフィカルユーザインタフェースアプリケーション、または別のプログラムを含み得る。追加として、CPU 6 は、GPU 12 の動作を制御するための GPU ドライバ 22 を実行し得る。ユーザは、キーボード、マウス、マイクロホン、タッチパッド、またはユーザインタフェース 4 を介してデバイス 2 に結合された別の入力デバイスのような、1 つまたは複数の入力デバイス (図示せず) を介してデバイス 2 への入力を提供し得る。

30

【 0 0 3 5 】

[0047] CPU 6 上で実行されるソフトウェアアプリケーションは、ディスプレイ 8 へのグラフィックスデータのレンダリングを生じさせるように CPU 6 に命令する 1 つまたは複数のグラフィックスレンダリング命令を含み得る。いくつかの例では、ソフトウェア命令は、例えば、OpenGL (登録商標) (Open Graphics Library) API、OpenGL ES (Open Graphics Library Embedded Systems) API、Direct3D API、X3D API、RenderMan API、WebGL API、あるいはその他任意の公的または独自の標準グラフィックス API などの、グラフィックスアプリケーションプログラムインタフェース (API) に準拠し得る。グラフィックスレンダリング命令を処理するために、CPU 6 は、GPU 12 に、グラフィックスデータのレンダリングの一部または全てを実行することを行わせるために、(例えば、GPU ドライバ 22 を通じて) GPU 12 に 1 つまたは複数のグラフィックスレンダリングコマンドを発行し得る。いくつかの例では、レンダリングされるグラフィックスデータは、例えば、点、線、三角形、四角形、トライアングルストリップ (triangle strips) などの、グラフィックスプリミティブのリストを含み得る。

40

【 0 0 3 6 】

[0048] 他の例では、CPU 6 上で実行されるソフトウェア命令は、GPU 12 に、G

50

P U ハードウェアの高度な並列性によって実行されるように適用可能な (applicable to be executed) 、より一般的な計算を実行するための汎用シェーダを実行させ得る。このような汎用アプリケーションは、いわゆる汎用グラフィックス処理ユニット (G P G P U) であり得、O p e n C L のような、汎用 A P I に準拠し得る。

【 0 0 3 7 】

[0049] G P U 1 2 は、ディスプレイ 8 に 1 つまたは複数のグラフィックスプリミティブをレンダリングするためにグラフィックス動作を実行するように構成され得る。したがって、C P U 6 上で実行中のソフトウェアアプリケーションのうちの 1 つがグラフィックス処理を必要とするとき、C P U 6 は、ディスプレイ 8 へのレンダリングのために、G P U 1 2 にグラフィックスコマンドおよびグラフィックスデータを提供し得る。グラフィックスデータは、例えば、描画コマンド、状態情報、プリミティブ情報、テクスチャ情報などを含み得る。G P U 1 2 は、いくつかの事例では、C P U 6 に比べて、複雑なグラフィック関連動作のより効率的な処理を提供する高度な並列構造を用いて構築され得る。例えば、G P U 1 2 は、並列的に (in a parallel manner) 複数の頂点または画素に対して動作するように構成された、複数のシェーダユニットのような複数の処理要素を含み得る。G P U 1 2 の高度な並列性は、いくつかの事例では、G P U 1 2 が、グラフィックス画像 (例えば、G U I および 2 次元 (2 D) および / または 3 次元 (3 D) グラフィックスシーン) を、C P U 6 を使用してディスプレイ 8 にこれらのシーンを直接描画するよりも迅速に、ディスプレイ 8 上に描画することを可能にし得る。

【 0 0 3 8 】

[0050] G P U 1 2 は、いくつかの事例では、デバイス 2 のマザーボードに組み込まれ得る。他の事例では、G P U 1 2 は、デバイス 2 のマザーボードにおけるポートでインストールされるグラフィックスカード上に存在し得るか、さもなければデバイス 2 と相互作用するように構成された周辺デバイス内に組み込まれ得る。G P U 1 2 は、1 つまたは複数のマイクロプロセッサ、特定用途向け集積回路 (A S I C) 、フィールドプログラマブルゲートアレイ (F P G A) 、デジタルシグナルプロセッサ (D S P) 、または他の同等の集積回路またはディスクリート論理回路などの、1 つまたは複数のプロセッサを含み得る。G P U 1 2 はまた、G P U 1 2 がマルチコアプロセッサと呼ばれ得るように、1 つまたは複数のプロセッサコアを含み得る。

【 0 0 3 9 】

[0051] G P U 1 2 は、グラフィックスメモリ 4 0 に直接結合され得る。したがって、G P U 1 2 は、バスを使用することなく、グラフィックスメモリ 4 0 からデータを読み取り、また、それにデータを書き込み得る。換言すると、G P U 1 2 は、オフチップメモリの代わりに、ローカル記憶装置を使用してローカルにデータを処理し得る。このようなグラフィックスメモリ 4 0 は、オンチップメモリと呼ばれ得る。これは、G P U 1 2 がバスを介してデータを読み取るおよび書き込む必要性、これは重いバストラフィックを経験し得る、を除去することによって、G P U 1 2 がより効率的な手法で動作することを可能にする。しかしながら、いくつかの事例では、G P U 1 2 は、別個のメモリを含まず、代わりに、バスを介してシステムメモリ 1 0 を利用し得る。グラフィックスメモリ 4 0 は、例えば、ランダムアクセスメモリ (R A M) 、スタティック R A M (S R A M) 、ダイナミック R A M (D R A M) 、消去可能なプログラマブル R O M (E P R O M) 、電氣的に消去可能なプログラマブル R O M (E E P R O M (登録商標)) 、フラッシュメモリ、磁気データ媒体または光記憶媒体などの、1 つまたは複数の揮発性または不揮発性メモリまたは記憶デバイスを含み得る。グラフィックスメモリ 4 0 は、キャッシュ 3 0 を含むおよび / またはキャッシュ 3 0 に結合され得る。キャッシュ 3 0 は、図 5 のキャッシュ 5 3 0 の例として用いられ得る。いくつかの例では、キャッシュ 3 0 は、G P U 1 2 によって実行されるメモリフェッチング動作のためのレベル 2 キャッシュである (すなわち、G P U 1 2 におけるキャッシュミスは、キャッシュ 3 0 へのアクセスをもたらし得る) 。他の例では、キャッシュ 3 0 は、レベル 3 キャッシュまたは同様のものであり得る。

【 0 0 4 0 】

[0052]いくつかの例では、GPU 12は、システムメモリ10に完全に形成された画像を記憶し得る。ディスプレイプロセッサ14は、システムメモリ10から画像を取り出し、画像を表示するためにディスプレイ8の画素を点灯させる値を出力し得る。ディスプレイ8は、GPU 12によって生成される画像コンテンツを表示する、デバイス2のディスプレイであり得る。ディスプレイ8は、液晶ディスプレイ(LCD)、有機発光ダイオードディスプレイ(OLED)、陰極線管(CRT)ディスプレイ、プラズマディスプレイ、または別のタイプのディスプレイデバイスであり得る。

【 0 0 4 1 】

[0053]本開示の技法によると、GPU 12は、生ベイヤータを受け取り、複数のテクセルを提供するために、別個の対応するテクセルとして、第1のラインの生ベイヤータおよび第2のラインの生ベイヤータの複数の重複してない2×2配列の各2×2配列をマッピングするように構成され得る。追加として、本開示の技法によると、GPU 12は、カメラセンサデータがMPIのような密にパックされたフォーマットであるとき、カメラセンサデータをアンパックするようにさらに構成され得る。

【 0 0 4 2 】

[0054]例えば、GPU 12は、テクスチャパイプライン523を含み得、したがって、本開示で説明される実例的な技法を実行するように構成され得る。GPU 12は、グラフィックス処理のためのメカニズムを提供し、これは、テクセル処理を含む。したがって、GPU 12は、複数の画素の生ベイヤータが単一の基本単位(例えば、GPU 12のためのテクセル)として処理されることができるよう、本開示で説明される技法を実行するのによく適し得る。このようにして、テクセルを基本単位として処理するGPU 12の能力は、ベイヤータ画像データの4つの異なる画素を処理するために利用されることができ

【 0 0 4 3 】

[0055]図8は、システム800の例を例示するブロック図であり、これは、さらなる詳細において、図7のCPU 6、GPU 12、およびシステムメモリ10の実例的なインプリメンテーションを含む。図8に示されるように、CPU 6は、少なくとも1つのソフトウェアアプリケーション18、グラフィックスAPI 20、およびGPUドライバ22を含み得、これらの各々は、CPU 6上で実行される1つまたは複数のソフトウェアアプリケーションまたはサービスであり得る。

【 0 0 4 4 】

[0056]CPU 6およびGPU 12に利用可能なメモリは、システムメモリ10およびフレームバッファ16を含み得る。フレームバッファ16は、システムメモリ10の一部であり得るか、またはシステムメモリ10とは別個であり得る。フレームバッファ16は、レンダリングされた画像データを記憶し得る。

【 0 0 4 5 】

[0057]ソフトウェアアプリケーション18は、GPU 12の機能性を利用する任意のアプリケーションであり得る。例えば、ソフトウェアアプリケーション18は、GUIアプリケーション、オペレーティングシステム、ポータブルマッピングアプリケーション、工学および芸術アプリケーションのためのコンピュータ支援型設計プログラム、ビデオゲームアプリケーション、あるいは2Dまたは3Dグラフィックスを使用する別のタイプのソフトウェアアプリケーションであり得る。

【 0 0 4 6 】

[0058]ソフトウェアアプリケーション18は、グラフィカルユーザインタフェース(GUI)および/またはグラフィックスシーンをレンダリングするようにGPU 12に命令する1つまたは複数の描画命令を含み得る。例えば、描画命令は、GPU 12によってレンダリングされる1つまたは複数のグラフィックスプリミティブのセットを定義する命令を含み得る。いくつかの例では、描画命令は、集合的に、GUIにおいて使用される複数のウィンドウサーフェス(windowing surfaces)の全てまたは一部を定義し得る。追

加の例では、描画命令は、集合的に、アプリケーションによって定義されるモデル空間またはワールド空間内に1つまたは複数のグラフィックスオブジェクトを含むグラフィックスシーンの全てまたは一部を定義し得る。

【0047】

[0059]ソフトウェアアプリケーション18は、1つまたは複数のグラフィックスプリミティブを、表示可能なグラフィックス画像へとレンダリングするための1つまたは複数のコマンドをGPU12へ発行するために、グラフィックスAPI20を介して、GPUドライバ22を起動(involve)し得る。いくつかの例では、これらプリミティブは、グラフィックスメモリ40に記憶され得る。

【0048】

[0060]GPU12がCPU6からコマンドを受け取ると、グラフィックス処理パイプライン24は、コマンドを復号し、コマンドにおいて指定される動作を実行するようにグラフィックス処理パイプライン24を構成する。

【0049】

[0061]フレームバッファ16は、GPU12のためのデスティネーション画素(destination pixels)を記憶する。各デスティネーション画素は、一意のスクリーン画素ロケーションに関連付けられ得る。いくつかの例では、フレームバッファ16は、デスティネーション画素ごとに色成分およびデスティネーションアルファ値を記憶し得る。

【0050】

[0062]いくつかの例では、GPU12は、シェーダユニット46、グラフィックス処理パイプライン24、テクスチャパイプライン23、およびグラフィックスメモリ40を含む。いくつかの例では、テクスチャパイプライン23は、図5のテクスチャパイプライン523の例として用いられ得る。1つまたは複数のシェーダプログラムは、GPU12におけるシェーダユニット46上で実行され得る。シェーダユニット46は、シェーダプロセッサ48を含み得、その各々は、フェッチングおよび復号動作(fetching and decoding operations)のための1つまたは複数の構成要素、算術演算を行うための1つまたは複数の算術論理演算ユニット、1つまたは複数のメモリ、キャッシュ、およびレジスタを含み得る。

【0051】

[0063]GPU12は、グラフィックス処理パイプライン24における頂点シェーダステージ、ハルシェーダステージ(a hull shader stage)、ドメインシェーダステージ、ジオメトリシェーダステージ、およびピクセルシェーダステージのうちの1つまたは複数を実行するために、シェーダユニット46にコマンドを送ることによって、頂点シェーディング、ハルシェーディング、ドメインシェーディング、ジオメトリシェーディング、ピクセルシェーディング、および同様のことなどの様々なシェーディング動作を実行するようにシェーダユニット46を指示し得る。いくつかの例では、GPUドライバ22は、GPU12内に含まれる1つまたは複数のプログラマブルシェーダユニット上にシェーダプログラムをダウンロードするように構成され得る。シェーダプログラムは、例えば、OpenGLシェーディング言語(GLSL)、上位レベルシェーディング言語(HLSL)、Cg(C for Graphics)シェーディング言語などの、上位レベルシェーディング言語で記述され得る。コンパイルされたシェーダプログラムは、GPU12内のシェーダユニット46の動作を制御する1つまたは複数の命令を含み得る。例えば、シェーダプログラムは、頂点シェーダステージの機能(functions)を行うためにシェーダユニット46によって実行され得る頂点シェーダプログラム、ハルシェーダステージの機能を行うためにシェーダユニット46によって実行され得るハルシェーダプログラム、ドメインシェーダステージの機能を行うためにシェーダユニット46によって実行され得るドメインシェーダプログラム、ジオメトリシェーダステージの機能を行うためにシェーダユニット46によって実行され得るジオメトリシェーダプログラム、および/またはピクセルシェーダの機能を行うためにシェーダユニット46によって実行され得るピクセルシェーダプログラムを含み得る。頂点シェーダプログラムは、プログラマブル頂点シェーダユニットまたは

10

20

30

40

50

統合型 (unified) シェーダユニットの実行を制御し得、1つまたは複数の頂点当たり (per-vertex) の動作を指定する命令を含む。

【0052】

[0064]いくつかの例では、生ベイヤ画像データは、システムメモリ10へ転送され得る。いくつかの例では、デバイス800は、生ベイヤ画像データを提供する画像センサを有するデジタルカメラに結合され得るか、それを含み得るか、またはその一部であり得、ここで、生ベイヤ画像データは、システムメモリ10に送信され得る。いくつかの例では、システムメモリ10における生ベイヤ画像データは、キャッシュ30に送信され、キャッシュ30に記憶され得る。

【0053】

[0065]テクスチャパイプライン23は、テクスチャリング機能 (fuctions) を実行するハードウェアユニットのプログラマブルパイプラインである。テクスチャパイプライン23は、シェーダプロセッサ48と通信状態にある。いくつかの例では、テクスチャパイプライン23は、ローカルキャッシュ31を含み得る。いくつかの例では、ローカルキャッシュ31は、レベル1キャッシュである。この例では、ローカルキャッシュ31が「レベル1キャッシュ」であることは、メモリ転送動作が実行されるときにローカルキャッシュ31が最初にチェックされて、レベル1キャッシュにおいてキャッシュミスが存在した場合にのみ、他のロケーションがチェックされることを意味する。

【0054】

[0066]いくつかの例では、テクスチャパイプライン23は、図6のプロセス630の動作 (acts) のような、動作 (acts) を実行し得る。いくつかの例では、テクスチャパイプライン23は、いくつかの例における図6のプロセスに従って、キャッシュ30から第1のラインの生ベイヤデータ、およびキャッシュ30から第2のラインの生ベイヤデータを受け取り、複数のテクセルを提供するために、別個の対応するテクセルとして、第1のラインの生ベイヤデータおよび第2のラインの生ベイヤデータの複数の重複してない2x2配列の各2x2配列をマッピングし、複数のテクセルのうちの少なくとも1つに対して少なくとも1つの動作を実行し得る。

【0055】

[0067]例えば、いくつかの例では、テクスチャパイプライン23は、生ベイヤ画像データの2x2画素のセットにアクセスするために、キャッシュ30から2つのラインの生ベイヤ画像データを読み取る。次に、テクスチャパイプライン23は、例えば、図9に例示される (および図9 - 図11と併せて、直後に続くセクションでより詳細に説明される) 手法で、キャッシュ30からの生データにおける各2x2BGG R (青緑緑赤) パターンの画素を、単一のテクセルの「R G B A」フォーマットにマッピングし得る。このようにして、各2x2ベイヤパターンは、単一の「R G B A」色として扱われる。

【0056】

[0068]図9は、ベイヤパターンをテクセルにマッピングする例を例示する概念図である。いくつかの例では、入力データは、生ベイヤMIP Iデータであり、また、他の例では、入力データは、例えば、代わりに8ビットのフォーマットまたは16ビットのフォーマットである、MIP Iフォーマットでない生ベイヤデータである。いくつかの例では、2x2ベイヤパターンからのデータは、例えば、いくつかの例では、図9に例示されるように、常に「R」として左上をマッピングし、常に「G」として右上をマッピングするなどのように、常に同じ手法でマッピングされる。

【0057】

[0069]示されるように、ベイヤスウィズリングを実行するために、テクスチャパイプラインは、各2x2パターンの画素が別個のテクセルにマッピングされることができるよう、一度に2つのラインのデータを受け取る。対照的に、従来、各画素は、別個のテクセルであり、したがって、従来、一度に1つのラインのデータのみが受け取られる必要があった。

【0058】

10

20

30

40

50

[0070] テクスチャパイプラインは、ベイヤードータが、 2×2 B G G R (青緑緑赤) ベイヤーパターンとして知られるフォーマットである、図 9 に例示される例で示され、また図 1 にも例示されたような、典型的なベイヤーフォーマットにおける 2×2 配列当たり 2 つの緑画素、1 つの赤画素、および 1 つの青画素などの、 2×2 配列関数 (function) でパターン化されるので、 2×2 配列のためのベイヤースウィズリングを実行する。しかしながら、他のベイヤーパターンが本開示の技法に従って使用され得、それに限定されるものではないが、R G G B (赤緑緑青)、G B R G (緑青赤緑)、G R B G (緑赤青緑)、R G B C (赤青緑クリア)、および R G B W (赤青緑白) を含む。R G B C および R G B W ベイヤーフォーマットは、B G G R ベイヤーパターンと同様に 2×2 配列でパターン化され、したがって、これらフォーマットの各々は、本開示の技法によるここで説明されるようなベイヤー画素の各 2×2 配列を単一のテクセルにマッピングするベイヤースウィズリングから利益を得る。

10

【0059】

[0071] 図 10 は、図 9 のマッピングされたテクセルのアドレス指定を例示する概念図である。いくつかの例では、この場合のベイヤー順序における画像座標 (i, j) アドレス指定は、図 10 に示されるとおりである。例示されるように、テクセルのテクスチャの幅および高さは、標準ベイヤーテクセルフォーマットの半分になり得る。各テクセルインデックスは、4 つの色値の 2×2 マトリクスを含む。

【0060】

[0072] 2×2 ベイヤー画素を対応する R G B A テクセルにマッピングした後、いくつかの例では、テクスチャパイプラインは、これらテクセルのうちの少なくとも 1 つに対して少なくとも 1 つの動作を実行する。テクスチャパイプラインは、メモリフェッチ当たり、4 つの色値の 2×2 マトリクスを返す。

20

【0061】

[0073] さらに例として、ベイヤービニングは、テクスチャパイプラインのバイリニアフィルタリング能力を使用して実行され得、ここで、ベイヤービニングは、テクスチャパイプラインによって実行されるバイリニアフィルタリング動作を含み得る。テクスチャパイプラインはまた、R G B A テクセルに対して読み取りのような動作を実行し得る。他の例では、テクスチャパイプラインは、適応型ベイヤーフィルタリングおよび / またはベイヤー統計 (Adaptive Bayer Filtering and/or Bayer statistics) を実行し得、ここで、適応型ベイヤーフィルタリングおよび / またはベイヤー統計は、R G B A テクセルのうちの 1 つまたは複数に対して実行され得る「Gather 4」のような収集動作を含み得る。いくつかの例では、Gather 4 動作は、4 つのロードではなく、1 つロードで所望のチャンネルをフェッチするために使用され得る。このような収集動作の例が、図 11 に例示される。

30

【0062】

[0074] 図 11 は、テクセルに対して収集動作を実行するテクスチャパイプラインの例を例示する概念図である。図 11 は、 2×2 ベイヤー配列からスウィズリングされた「R G B A」テクセルに対するテクスチャパイプラインの動作 Gather の例の影響 (effect) を例示する。具体的には、図 11 は、B チャンネルに対して実行される収集動作を例示する。上記で説明されたように、各別個のテクセルは、動作が個々のベイヤー画素のレベルではなく、テクセルのレベルで実行されるように、各セットのアドレス座標が、単一のベイヤー画素ではなく、 2×2 ベイヤー配列テクセル全体を指し示す、別個の 2×2 ベイヤー配列である。例示される例で示されるように、収集動作の例は、4 つの隣接する 2×2 ベイヤー配列テクセルの各々から 2×2 ベイヤーパターンの単一のチャンネルをフェッチする。テクスチャパイプラインはまた、ベイヤードータに対して実行されることが可能なその他任意の機能を実行し得、特に、最近点サンプリング (Nearest and Point Sampling) を含み、これは、ベイヤードータが、個々の画素のレベルではなく、 2×2 配列の画素のレベルで動作されるという事実を除いて、従来の内容 (matter) で 2×2 ベイヤー配列に対して実行され得、これは動作をより効率的にする。

40

50

【 0 0 6 3 】

[0075] 上記および以下で説明されるように、バイリニアフィルタリングはまた、 2×2 ベイヤー配列からスウィズリングされた「R G B A」テクセルに対して実行されることができる。バイリニアフィルタリングは、それによって画素値が、同じ色の隣接する画素値に基づいて平均化されるプロセスである。リニアフィルタリングは、同じラインにおける隣接する画素値に基づく平均化を指す（したがって、水平に隣接する画素または垂直に隣接する画素のいずれかが、リニアフィルタリングで平均化される）。バイリニアフィルタリングは、垂直に隣接するものと水平に隣接するものの両方に基づいて画素値を平均化することを指す。バイリニアフィルタリングの使用は、隣接する画素に基づいて平均化することによって、アーティファクトをよりゆるやか（less severe）にし、その結果、外れ値を有する誤りのある画素の影響を低減させることによって、視覚アーティファクトの低減を達成する。

10

【 0 0 6 4 】

[0076] バイリニアフィルタリングは、テクスチャパイプラインのバイリニアフィルタリング機能を利用するために隣接する画素を平均化するとき、著しく少ないロードが必要とされるので、 2×2 ベイヤー配列からスウィズリングされた「R G B A」テクセルにより（with）、著しくより効率的である。

【 0 0 6 5 】

[0077] 図 1 2 は、実例的なデバイス 1 2 0 0 を例示する機能ブロック図であり、これは、図 8 のシステム 8 0 0 の一部として用いられ得る。デバイス 1 2 0 0 は、レベル 2 キャッシュ 1 2 3 0、テクスチャパイプライン 1 2 2 3 およびシェーダプロセッサ 1 2 4 8 を含み、これらは、それぞれ図 8 のキャッシュ 3 0、パイプライン 2 3、およびシェーダプロセッサ 4 8 の例として用いられ得る。テクスチャパイプライン 1 2 2 3 は、アンパックおよびベイヤー スウィズルブロック 1 2 7 1、レベル 1 キャッシュ 1 2 3 1、フォーマット変換ブロック 1 2 7 2、およびバイリニア最近 gather 4（bilinear nearest gather4）ブロック 1 2 7 3 を含む。キャッシュ 1 2 3 1 は、図 8 のキャッシュ 3 1 の例として用いられ得る。図 1 2 に示されるように、アンパックおよびベイヤー スウィズルブロック 1 2 7 1、レベル 1 キャッシュ 1 2 3 1、フォーマット変換ブロック 1 2 7 2、およびバイリニア最近 gather 4 ブロック 1 2 7 3 は、全てテクスチャパイプライン 1 2 2 3 に存在する。

20

30

【 0 0 6 6 】

[0078] いくつかの例では、ブロック 1 2 7 1 は、ブロック 1 2 7 1 に提供される画像データが M I P I データであるか否かを認識し（すなわち、画像データが M I P I フォーマットであるかどうかについての決定を行い）、画像データが M I P I 生データである場合、ブロック 1 2 7 1 におけるハードウェアを介して M I P I 生データをアンパックするように構成されたハードウェアを含む。ブロック 1 2 7 1 は、ハードウェアを含むアンパッキング回路であり得る。デバイス 1 2 0 0 のいくつかの例は、M I P I パックされたベイヤーデータまたはアンパックされたベイヤー画像のいずれかとして生ベイヤーデータを送ることが可能であるカメラと連携して使用され、したがって、デバイス 1 2 0 0 は、M I P I パックされたベイヤーデータおよびアンパックされたベイヤーデータの両方を効率的にサポートする。いくつかの例では、正規化データタイプおよび整数データタイプの両方が、ブロック 1 2 7 1 によってサポートされる。いくつかの例では、ブロック 1 2 7 1 は、16 ビットのデータフォーマット（すなわち、画素当たり 16 ビット）に M I P I 生データをアンパックする。このようにして、生ベイヤー M I P I データは、ブロック 1 2 7 1 がアンパッキングおよびベイヤー スウィズリングを実行した後、テクセル当たり 64 ビット（4 つの 16 ビット画素）にアンパックされる。他の例では、ブロック 1 2 7 1 は、16 ビットのフォーマット以外の適切なフォーマットに M I P I 生データをアンパックし得る。

40

【 0 0 6 7 】

[0079] いくつかの例では、テクスチャパイプライン 1 2 2 3 は、異なるタイプの処理用

50

の異なるタイプのテクスチャを識別するために、「テクスチャデータタイプ」または単に「テクスチャタイプ」と呼ばれるものを利用し、様々なテクスチャに、テクスチャに従って適切なテクスチャタイプを割り当て、その後、識別されたテクスチャデータタイプに基づいてグラフィック処理機能を実行する。しかしながら、いくつかの例では、テクスチャパイプライン 1 2 2 3 におけるハードウェアによって実行される機能は、テクスチャを実際には表さないデータにテクスチャデータタイプを割り当てることによってアクセスされ得、したがって、テクスチャパイプライン 1 2 2 3 におけるハードウェアは、割り当てられたテクスチャデータタイプに基づいて、このデータに対して機能を実行することができる。いくつかの例では、新しいテクスチャデータタイプが、生画像データに対して処理を実行するために、テクスチャパイプライン 1 2 2 3 のために割り当てられる。例えば、いくつかの例では、M I P I 生データは、1 つ（または複数）の別個のテクスチャデータタイプである。テクスチャパイプライン 1 2 2 3 は、全てのデータフォーマットについて、テクスチャとして読取りアクセスを提供するように構成される。

10

20

30

40

50

【 0 0 6 8 】

[0080]いくつかの例では、ブロック 1 2 7 1 は、ブロック 1 2 7 1 内に内蔵キャッシュを含む。いくつかの例では、レベル 2 キャッシュ 1 2 3 0 からの応答が、将来のレベル 1 キャッシュミスのために、ブロック 1 2 7 1 の内蔵キャッシュに記憶される。いくつかの例では、レベル 1 キャッシュ 1 2 3 1 におけるキャッシュミスは、アドレスをタグとして使用してブロック 1 2 7 1 における内蔵キャッシュを最初にチェックするであろう。このようにして、いくつかの例では、ブロック 1 2 7 1 の内蔵キャッシュは、それがレベル 1 キャッシングとレベル 2 キャッシングの間の中間であるので、レベル 1 . 5 キャッシュとして効果的に動作する。1 2 7 1 におけるキャッシュの使用は、いくつかの例では、レベル 2 キャッシュ 1 2 3 0 が、アンパックされた M I P I データを含むので、効率を増大させる。ブロック 1 2 7 1 におけるレベル 1 . 5 キャッシュが使用されない場合には、レベル 1 ローカルキャッシュ 1 2 3 1 のキャッシュミスに際して、レベル 2 キャッシュ 1 2 3 0 におけるデータが、密にパックされたフォーマットであり、したがって、アラインされていないので、レベル 2 キャッシュ 1 2 3 0 におけるキャッシュラインへの複数のアクセスが必要とされ得る。ブロック 1 2 7 1 のレベル 1 . 5 キャッシュにおけるデータがアンパックされている、ブロック 1 2 7 1 におけるレベル 1 . 5 キャッシュを使用することによって、レベル 2 キャッシュ 1 2 3 0 におけるアラインされていないデータへのフェッチの回数は低減される。また、従来、レベル 2 キャッシュ 1 2 3 0 は、スプリットティング (splitting) の要求 (例えば、アラインされていないデータをアラインする要求) を扱うであろう。しかしながら、ブロック 1 2 7 1 がアンパッキングを実行するので、レベル 2 キャッシュ 1 2 3 0 は、スプリットティングの要求を扱う必要がなく、これは、レベル 2 キャッシュ 1 2 3 0 の効率を増大させる。

【 0 0 6 9 】

[0081]生 M I P I データをアンパックするために、(図 8 の)グラフィックスメモリ 4 0 を使用することが可能であるが、これは、M I P I データが密にパックされており、したがって、アラインされていないので、追加の同期を必要とする。アンパックされた画素を保持するレベル 1 キャッシュ 1 2 3 1 を用いて (with)、テクスチャパイプライン 1 2 2 3 によって実行される M I P I アンパッキングは、データが利用可能であることを確実にするための (図 8 の)グラフィックスメモリ 4 0 との同期を必要とすることなく、テクスチャパイプライン 1 2 2 3 における複数のワークアイテムにわたって生 M I P I 画像データのアンパックされた画素を共有することを可能にする。特にレベル 1 キャッシュ 1 2 3 1 からデータを受け取ることは、生画像データに対する処理が並列に行われることができるように、テクスチャパイプライン 1 2 2 3 における複数のワークアイテムによるアクセスのためにアンパックされた画素を保持することを可能にし、これは増大された効率を可能にする。また、上記で説明された階層的なキャッシュ構造は、上述された理由から、増大された効率を可能にする。

【 0 0 7 0 】

[0082]いくつかの例では、フォーマット変換ブロック 1 2 7 2 は、データを F P (浮動小数点)フォーマットに変換する。ブロック 1 2 7 3 は、図 1 1 に関する説明において上記で説明されたように、バイリニア、最近点サンプリング、および Gather 4 機能などの様々な機能をテクセルに対して実行し得る。

【0071】

[0083]様々な例では、テクスチャパイプライン 1 2 2 3 は、B のみ、G のみ、および R のみの画素をフェッチするために、マッピングされたベイヤー R G B A テクセルに対してテクスチャ収集動作を使用し得る。追加で、テクスチャパイプライン 1 2 2 3 は、B、G、および R チャンネルに別々に適用され得るバイリニアフィルタリングを適用し得る。

【0072】

[0084]上記で説明されたように、ベイヤースウィズリングにより、単一の動作は、一度に 4 つのベイヤー画素に対して動作が (テクスチャパイプライン 1 2 2 3 によって) 実行されることを効果的に可能にし得る。例えば、ベイヤースウィズリング後の R G B A テクセルに対する読取り動作は、一度に 1 つのベイヤー画素ではなく、一度に 4 つのベイヤー画素を効果的に読み取り、Gather 4 動作は、4 つのロードではなく、1 つのロードにおいて所望のチャンネルをフェッチするために使用され得、以下同様である。

【0073】

[0085]いくつかの例では、デバイス 1 2 0 0 は、様々な機能が A P I を介してアクセス可能になるように、それによってそれが、A P I を介して、ブロック 1 2 7 1 によって実行されるアンパッキング機能およびベイヤー 2 x 2 画素から R G B A テクセルへのマッピング (すなわち、ベイヤースウィズリング) を含む、ブロック 1 2 7 1 の様々な機能を実行することが可能である A P I を (例えば、フレームワーク拡張を介して) 提供する。したがって、いくつかの例では、ブロック 1 2 7 1 は、テクスチャタイプを介してではなく、A P I に応答してアンパッキングを実行し得る。いくつかの例では、A P I は、テクスチャタイプの代わりに使用され得、一方で、他の例では、テクスチャタイプと A P I 拡張の両方が使用され得るとともに、A P I 拡張は、テクスチャデータを含む。他の例では、テクスチャパイプライン 1 2 2 3 は、アンパッキング機能およびベイヤースウィズリング機能のためにテクスチャデータタイプを使用する。いくつかの例では、A P I は、テクスチャタイプの代わりに使用され得、一方で、他の例では、テクスチャタイプと A P I 拡張の両方が使用され得るとともに、A P I 拡張は、テクスチャデータを含む。他の例は、A P I ではなく、テクスチャタイプを使用し得る。いくつかの例では、シェーダプロセッサ 1 2 4 8 は、デモザイキング動作を実行するためにテクセルをフェッチし得る。

【0074】

[0086]いくつかの例では、シェーダプロセッサ 1 2 4 8 は、その後、さらなる処理のために画像信号プロセッサ (I S P) (図示せず) に画像データを送信し得る。M I P I アンパッキングおよびベイヤースウィズリングを含む、テクスチャパイプライン 1 2 2 3 におけるハードウェアによって実行される様々な機能、およびアンパックされた「R G B A」テクセルに対して実行される様々な動作 (特に、収集機能およびバイリニアフィルタリングを含む) は、I S P によって実行される画像信号処理より前に実行される画像前処理として機能し得る。ここで説明されるテクスチャパイプライン 1 2 2 3 におけるハードウェアによって実行される前処理は、特に、P D A F (位相検出オートフォーカス)、R G B C (赤緑青クリア)、インターリーブドハイダイナミックレンジ (i H D R)、コンピュータショナルフォトグラフィ、およびライトフィールドコンピュータシオンと併せると、これらのアプリケーションが、M I P I アンパッキングおよび / またはバイリニアフィルタを使用するまたはそれらから利益を得て、したがって、より効率的な M I P I アンパッキングおよび / またはより効率的なバイリニアフィルタリングから利益を得るので、特に有用であり得る。テクスチャパイプライン 1 2 2 3 におけるハードウェアによって実行される様々な機能はまた、後処理として行われることができる。すなわち、機能が I S P で実行された後、画像データは、上記に説明された様々な動作の実行のために、テクスチャパイプライン 1 2 2 3 に送られることができる。

10

20

30

40

50

【 0 0 7 5 】

[0087]図 1 3 は、実例的なデバイス 1 3 0 2 の機能的な態様の例を示す機能ブロック図であり、これは、さらなる詳細において、カメラ 1 3 2 3 および図 7 のデバイス 2 の一部の例として用いられ得る。GPU 1 3 1 2 は、図 7 の GPU 1 2 の例として用いられ得る。いくつかの例では、生バッファ (raw buffers) 1 3 5 4 および YUV バッファ 1 3 5 8 は、各々が図 8 のシステムメモリ 1 0 の一部の例である。図 1 3 は、正確な構造ではなく、実例的な機能プロセスフローを示す機能ブロック図であり - 例えば、GPU 1 3 1 2 は、図 1 3 において 2 度示されている - これは 2 つの GPU があるのではなく、例示される特定の例では、図 1 3 は、機能ブロック図であり、同じ GPU 1 3 1 2 が機能プロセスフローにおける 2 つの別個のポイントにおいて機能を実行するからである。

10

【 0 0 7 6 】

[0088]カメラ 1 3 2 3 は、(例えば、現代のスマートフォン上に配置されるような)主センサ 1 3 5 0 およびフロントセンサ 1 3 5 2 を含む 1 つまたは複数のカメラセンサを含み得る。(例えば、MIPPI 1 0 ベイヤーパターンフォーマットにおける)センサ 1 3 5 0 および / またはセンサ 1 3 5 2 によって生成される生センサデータは、生バッファ 1 3 5 4 に記憶され得る。生センサデータは、1 つまたは複数のカメラシリアルインタフェース (CSI) デコーダ CSID によって生バッファ 1 3 5 4 にルーティングされ得る。CSI デコーダ CSID は、統計およびダウンスケールユニットを組み込み得る。本開示の技法によると、GPU 1 3 1 2 は、GPU 1 3 1 2 におけるテクスチャパイプラインを使用して、生センサデータをアンパックおよび / またはフィルタリングし、アンパックされたデータを生バッファ 1 3 5 4 に記憶するように構成され得る。その後、アンパックされたデータは、圧縮されたフォーマット (例えば、JPEG) で画像ファイルを生成するために、画像信号プロセッサ 1 3 5 6 によって使用され得る。その後、圧縮されたカメラデータは、YUV バッファ 1 3 5 8 に記憶され得る。その後、このようなデータは、GPU 1 3 1 2、図 7 のディスプレイプロセッサ 2 3、または図 7 の CPU 6 によってさらに操作され、および / または図 7 のディスプレイ 1 8 上に表示され得る (上記で説明されたように、図 1 3 のデバイスは、図 1 3 のデバイス 2 の一部として用いられ得る)。

20

【 0 0 7 7 】

[0089]例に依存して、ここで説明された任意の技法のある特定の動作 (acts) またはイベントは、異なるシーケンスで実行されることができ、概して追加、統合、または省略され得る (例えば、すべての説明された動作 (acts) またはイベントは、これら技法の実施に必ずしも必要ではない) ことが認識されるべきである。さらに、ある特定の例では、動作 (acts) またはイベントは、順次ではなく、例えば、マルチスレッド処理、割込み処理、または複数のプロセッサを通じて、同時並行 (concurrently) に実行され得る。

30

【 0 0 7 8 】

[0090]1 つまたは複数の例では、説明された機能は、ハードウェア、ソフトウェア、ファームウェア、またはこれらの任意の組合せでインプリメントされ得る。ソフトウェアでインプリメントされる場合、これら機能は、コンピュータ可読媒体上で 1 つまたは複数の命令またはコードとして記憶または送信され、ハードウェアベースの処理ユニットによって実行され得る。コンピュータ可読媒体は、例えば、通信プロトコルに従って、1 つの場所から別の場所へのコンピュータプログラムの転送を容易にする任意の媒体を含む通信媒体、またはデータ記憶媒体のような有形の媒体に対応するコンピュータ可読記憶媒体を含み得る。

40

【 0 0 7 9 】

[0091]このように、コンピュータ可読媒体は一般に、(1) それ自体が信号を伝搬することを含まない有形のコンピュータ可読記憶媒体、または (2) 信号または搬送波のような通信媒体に対応し得る。データ記憶媒体は、本開示で説明された技法のインプリメンテーションのための命令、コードおよび / またはデータ構造を取り出すために、1 つまたは複数のコンピュータまたは 1 つまたは複数のプロセッサによってアクセスされることができ任意の利用可能な媒体であり得る。コンピュータプログラム製品は、コンピュータ可

50

読媒体を含み得る。

【0080】

[0092] 限定ではなく例として、このようなコンピュータ可読記憶媒体は、RAM、ROM、EEPROM、CD-ROMまたは他の光ディスク記憶装置、磁気ディスク記憶装置、または他の磁気記憶デバイス、フラッシュメモリ、あるいは、データ構造または命令の形式で所望のプログラムコードを記憶するために使用されることができ、かつコンピュータによってアクセスされることができるその他任意の媒体を備えることができる。また、任意の接続は、厳密にはコンピュータ可読媒体と称される。例えば、命令が、同軸ケーブル、光ファイバケーブル、ツイストペア、デジタル加入者回線(DSL)、または赤外線、無線、およびマイクロ波などのワイヤレス技術を使用して、ウェブサイト、サーバ、またはその他の遠隔ソースから送信される場合には、この同軸ケーブル、光ファイバケーブル、ツイストペア、DSL、または赤外線、無線、およびマイクロ波などのワイヤレス技術は、媒体の定義に含まれる。

10

【0081】

[0093] しかしながら、コンピュータ可読記憶媒体およびデータ記憶媒体は、接続、搬送波、信号、または同様のものを含まないが、その代りに有形の記憶媒体に向けられることが理解されるべきである。ここで使用される場合、ディスク(disk)およびディスク(disc)は、コンパクトディスク(CD)、レーザーディスク(登録商標)、光ディスク、デジタル多目的ディスク(DVD)、フロッピー(登録商標)ディスクおよびBlu-ray(登録商標)ディスクを含み、ここでディスク(disks)は、通常磁氣的にデータを再生し、一方ディスク(discs)は、レーザーを用いて光学的にデータを再生する。上記の組合せもまた、コンピュータ可読媒体の範囲内に含まれるべきである。

20

【0082】

[0094] 命令は、1つまたは複数のデジタルシグナルプロセッサ(DSP)、汎用マイクロプロセッサ、特定用途向け集積回路(ASIC)、フィールドプログラマブル論理アレイ(FPGA)、または他の同等の集積されたまたはディスクリートな論理回路などの、1つまたは複数のプロセッサによって実行され得る。したがって、ここで使用される場合、「プロセッサ」という用語は、任意の前述の構造またはここで説明された技法のインプリメンテーションに適したその他任意の構造を指し得る。加えて、いくつかの態様では、ここで説明された機能は、符号化および復号のために構成された専用のハードウェアモジュールおよび/またはソフトウェアモジュール内に提供され得、または、組み合わされたコーデックに組み込まれ得る。また、これら技法は、1つまたは複数の回路または論理要素において完全にインプリメントされることができる。

30

【0083】

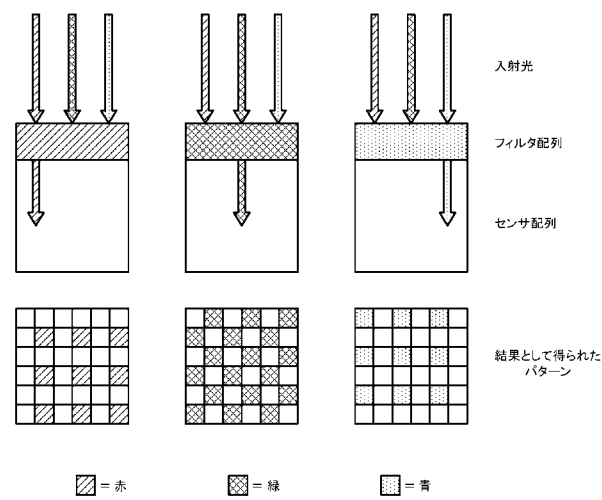
[0095] 本開示の技法は、ワイヤレスハンドセット、集積回路(IC)またはICのセット(例えば、チップセット)を含む、幅広い様々なデバイスまたは装置でインプリメントされ得る。様々な構成要素、モジュール、またはユニットが、開示された技法を実行するように構成されたデバイスの機能的な態様を強調するために、本開示において説明されているが、異なるハードウェアユニットによる実現を必ずしも必要とするわけではない。むしろ、上記で説明されたように、様々なユニットは、コーデックハードウェアユニットにおいて組み合わせられるか、または、適切なソフトウェアおよび/またはファームウェアと併せて、上記で説明されたような1つまたは複数のプロセッサを含む、相互運用ハードウェアユニットの集合によって提供され得る。

40

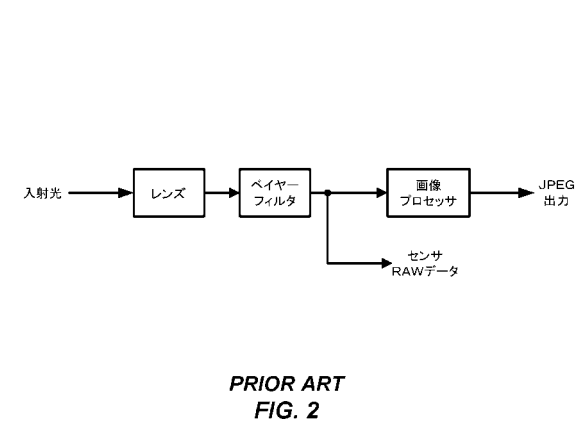
【0084】

[0096] 様々な例が説明された。これらおよび他の例は、以下の特許請求の範囲の範囲内にある。

【 図 1 】

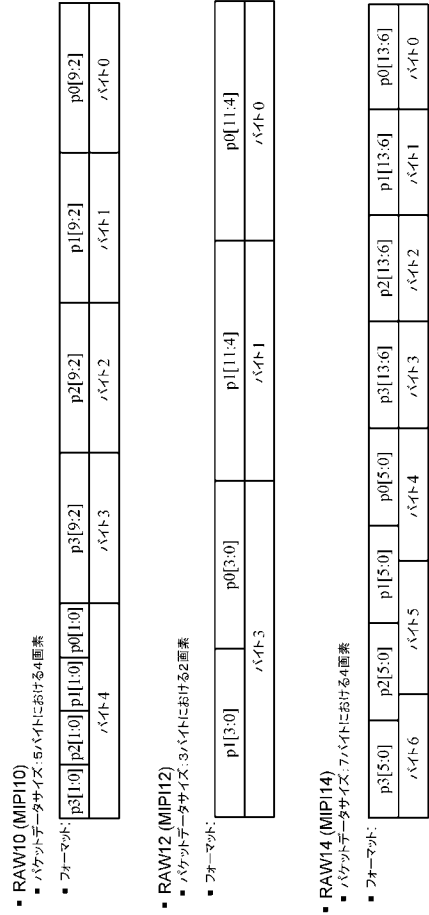


【 図 2 】



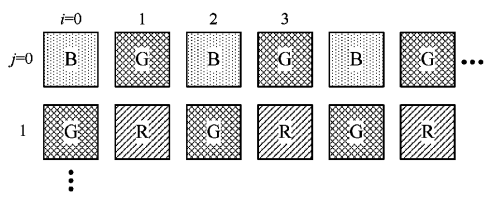
PRIOR ART
FIG. 1

【 図 3 】



PRIOR ART
FIG. 3

【 図 4 】



PRIOR ART
FIG. 4

【図 5】

501



FIG. 5

【図 6】

630

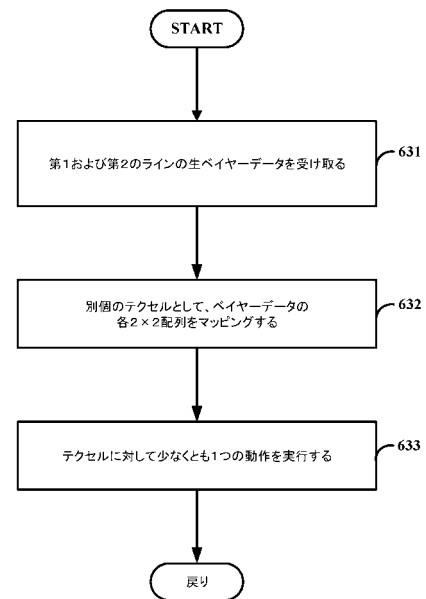


FIG. 6

【図 7】

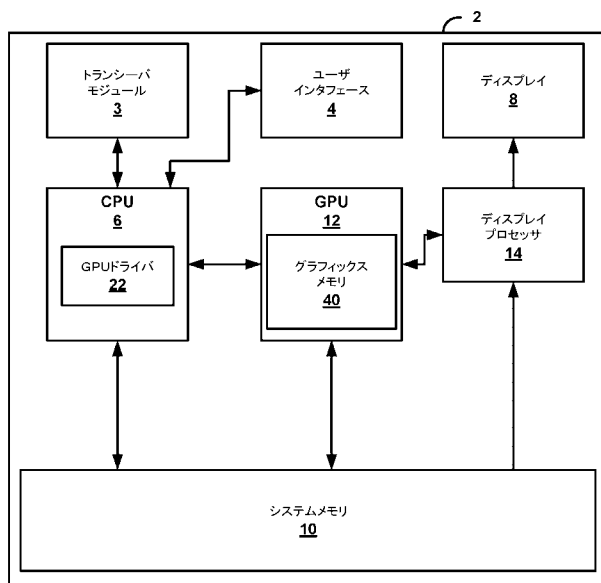


FIG. 7

【図 8】

800

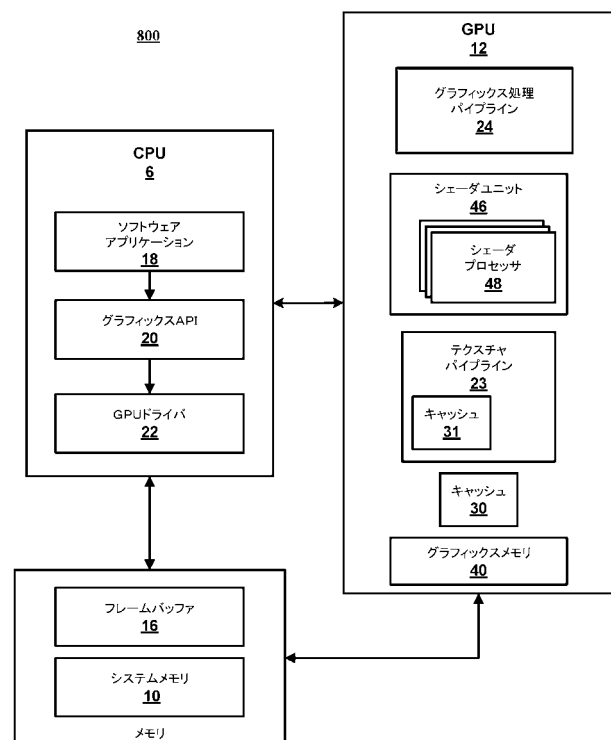


FIG. 8

【図 9】

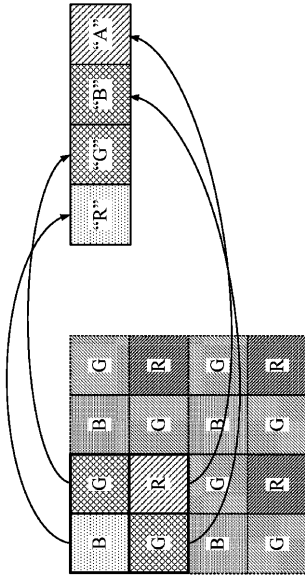


FIG. 9

【図 1 2】

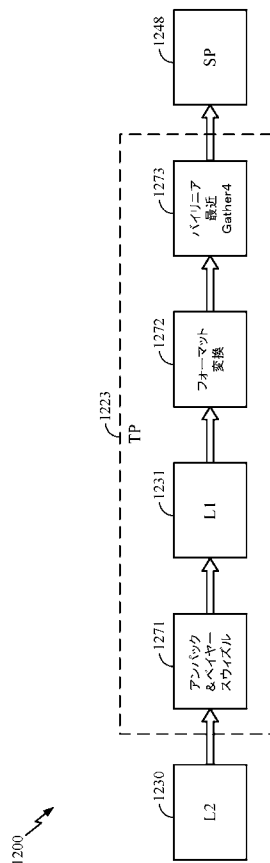


FIG. 12

【図 1 0】

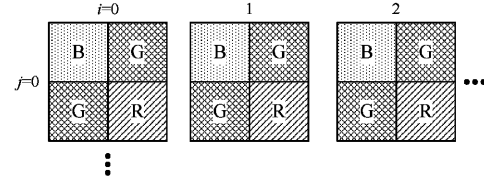


FIG. 10

【図 1 1】

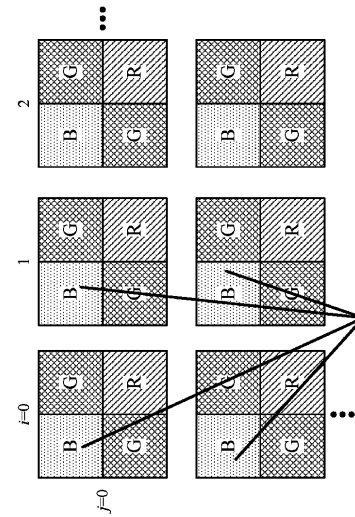


FIG. 11

【図 1 3】

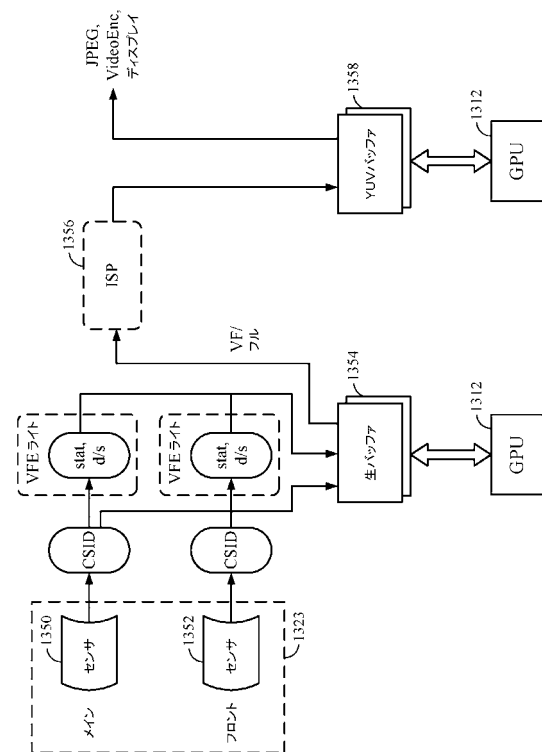


FIG. 13

【手続補正書】

【提出日】平成29年9月27日(2017.9.27)

【手続補正 1】

【補正対象書類名】特許請求の範囲

【補正対象項目名】全文

【補正方法】変更

【補正の内容】

【特許請求の範囲】

【請求項 1】

グラフィックス処理のための方法であって、

第 1 のラインの生バイヤーデータおよび第 2 のラインの生バイヤーデータを含む M I P I フォーマットにおける生バイヤーデータをキャッシュから受け取ることと、

グラフィックス処理ユニットのテクスチャパイプラインハードウェアを使用して、前記 M I P I フォーマットにおける前記生バイヤーデータをアンパックすることと、

複数のテクセルを提供するために、別個の対応するテクセルとして、前記第 1 のラインの生バイヤーデータおよび前記第 2 のラインの生バイヤーデータの複数の重複してない 2×2 配列の各 2×2 配列を、前記テクスチャパイプラインハードウェアを用いてマッピングすることと、

前記複数のテクセルのうちの少なくとも 1 つに対して少なくとも 1 つの動作を実行することと

を備える方法。

【請求項 2】

前記少なくとも 1 つの動作は、バイヤービニング動作、バイリニアフィルタ動作、読取り動作、または収集動作のうちの少なくとも 1 つを含む、請求項 1 に記載の方法。

【請求項 3】

前記複数のテクセルの各テクセルは、R G B A フォーマットである、請求項 1 に記載の方法。

【請求項 4】

前記複数のテクセルのうちの少なくとも 1 つに対して前記少なくとも 1 つの動作を実行した後、画像信号プロセッサに前記複数のテクセルを送信することをさらに備える、請求項 1 に記載の方法。

【請求項 5】

前記複数のテクセルを提供するために、別個の対応するテクセルとして、前記第 1 のラインの生バイヤーデータおよび前記第 2 のラインの生バイヤーデータの複数の重複してない 2×2 配列の各 2×2 配列を前記マッピングすることは、アプリケーションプログラムインタフェースを介してアクセス可能である、請求項 1 に記載の方法。

【請求項 6】

前記 M I P I P フォーマットにおける前記生バイヤーデータを前記アンパックすることは、対応するテクスチャデータタイプに関連する、請求項 1 に記載の方法。

【請求項 7】

前記 M I P I P フォーマットにおける前記生バイヤーデータを前記アンパックすることは、アプリケーションプログラムインタフェースを介してアクセス可能である、請求項 1 に記載の方法。

【請求項 8】

グラフィックス処理のためのデバイスであって、

キャッシュと、

第 1 のラインの生バイヤーデータおよび第 2 のラインの生バイヤーデータを含む M I P I フォーマットにおける生バイヤーデータを前記キャッシュから受け取り、

前記 M I P I フォーマットにおける前記生バイヤーデータをアンパックし、

複数のテクセルを提供するために、別個の対応するテクセルとして、前記第 1 のライ

ンの生ベイヤータおよび前記第 2 のラインの生ベイヤータの複数の重複していない 2×2 配列の各 2×2 配列をマッピングし、

前記複数のテクセルのうちの少なくとも 1 つに対して少なくとも 1 つの動作を実行する

ように構成されたテクスチャパイプラインハードウェアとを備えるデバイス。

【請求項 9】

前記テクスチャパイプラインハードウェアは、前記少なくとも 1 つの動作が、ベイヤータビニング動作、パイリニアフィルタ動作、読取り動作、または収集動作のうちの少なくとも 1 つを含むように構成される、請求項 8 に記載のデバイス。

【請求項 10】

前記テクスチャパイプラインハードウェアは、前記複数のテクセルの各テクセルが、RGB A フォーマットであるように構成される、請求項 8 に記載のデバイス。

【請求項 11】

画像信号プロセッサをさらに備え、ここにおいて、前記画像信号プロセッサは、テクスチャプロセッサが前記複数のテクセルのうちの少なくとも 1 つに対して前記少なくとも 1 つの動作を実行した後に、前記複数のテクセルを受け取るように構成される、請求項 8 に記載のデバイス。

【請求項 12】

前記テクスチャパイプラインハードウェアは、アプリケーションプログラムインタフェースを介してアクセス可能である手法で、前記複数のテクセルを提供するために、別個の対応するテクセルとして、前記第 1 のラインの生ベイヤータおよび前記第 2 のラインの生ベイヤータの複数の重複していない 2×2 配列の各 2×2 配列をマッピングするように構成される、請求項 8 に記載のデバイス。

【請求項 13】

前記テクスチャパイプラインハードウェアは、前記 MIP フォーマットにおける前記生ベイヤータをアンパックすることが、対応するテクスチャデータタイプに関連するように構成される、請求項 8 に記載のデバイス。

【請求項 14】

前記テクスチャパイプラインハードウェアは、前記 MIP フォーマットにおける前記生ベイヤータをアンパックすることがアプリケーションプログラムインタフェースを介してアクセス可能であるように構成される、請求項 8 に記載のデバイス。

【請求項 15】

前記キャッシュは、レベル 2 キャッシュであり、前記テクスチャパイプラインハードウェアは、

ローカルキャッシュと、

前記レベル 2 キャッシュから前記第 1 のラインの生ベイヤータ、および前記レベル 2 キャッシュから前記第 2 のラインの生ベイヤータを受け取り、前記複数のテクセルを提供するために、別個の対応するテクセルとして、前記第 1 のラインの生ベイヤータおよび前記第 2 のラインの生ベイヤータの前記複数の重複していない 2×2 配列の各 2×2 配列をマッピングし、前記ローカルキャッシュに前記複数のテクセルを送信するように構成されたアンパッキング回路と

を含む、請求項 8 に記載のデバイス。

【請求項 16】

前記アンパッキング回路は、

前記生ベイヤータが MIP フォーマットであるかどうかについての決定を行い、

前記生ベイヤータが MIP フォーマットであると決定された場合、前記生ベイヤータをアンパックする

ようにさらに構成される、請求項 15 に記載のデバイス。

【請求項 17】

前記アンパッキング回路は、別個のキャッシュを含む、請求項 1 5 に記載のデバイス。

【請求項 1 8】

グラフィックス処理のためのデバイスであって、

第 1 のラインの生ベイヤードータおよび第 2 のラインの生ベイヤードータを含む M I P
I フォーマットにおける生ベイヤードータを受け取るための手段と、

グラフィックス処理ユニットのテクスチャパイプラインハードウェアを使用して、前記
M I P I フォーマットにおける前記生ベイヤードータをアンパックするための手段と、

複数のテクセルを提供するために、別個の対応するテクセルとして、前記第 1 のライン
の生ベイヤードータおよび前記第 2 のラインの生ベイヤードータの複数の重複してない 2
× 2 配列の各 2 × 2 配列を、前記テクスチャパイプラインハードウェアを用いてマッピング
するための手段と、

前記複数のテクセルのうちの少なくとも 1 つに対して少なくとも 1 つの動作を実行する
ための手段と

を備えるデバイス。

【請求項 1 9】

前記少なくとも 1 つの動作は、ベイヤードビン動作、バイリニアフィルタ動作、読取
り動作、または収集動作のうちの少なくとも 1 つを含む、請求項 1 8 に記載のデバイス。

【請求項 2 0】

前記複数のテクセルの各テクセルは、R G B A フォーマットである、請求項 1 8 に記載
のデバイス。

【請求項 2 1】

前記複数のテクセルのうちの少なくとも 1 つに対して前記少なくとも 1 つの動作を実行
した後、画像信号プロセッサに前記複数のテクセルを送信するための手段をさらに備える
、請求項 1 8 に記載のデバイス。

【請求項 2 2】

前記複数のテクセルを提供するために、別個の対応するテクセルとして、前記第 1 のラ
インの生ベイヤードータおよび前記第 2 のラインの生ベイヤードータの複数の重複してな
い 2 × 2 配列の各 2 × 2 配列を前記マッピングすることは、アプリケーションプログラム
インタフェースを介してアクセス可能である、請求項 1 8 に記載のデバイス。

【請求項 2 3】

前記 M I P I P フォーマットにおける前記生ベイヤードータを前記アンパックすること
は、対応するテクスチャデータタイプに関連する、請求項 1 8 に記載のデバイス。

【請求項 2 4】

前記 M I P I P フォーマットにおける前記生ベイヤードータを前記アンパックすること
は、アプリケーションプログラムインタフェースを介してアクセス可能である、請求項 1
8 に記載のデバイス。

【手続補正 2】

【補正対象書類名】明細書

【補正対象項目名】0 0 8 4

【補正方法】変更

【補正の内容】

【0 0 8 4】

[0096] 様々な例が説明された。これらおよび他の例は、以下の特許請求の範囲の範囲内
にある。

以下に本願の出願当初の特許請求の範囲に記載された発明を付記する。

[C 1] グラフィックス処理のための方法であって、

第 1 のラインの生ベイヤードータおよび第 2 のラインの生ベイヤードータを受け取るこ
とと、

複数のテクセルを提供するために、別個の対応するテクセルとして、前記第 1 のライン
の生ベイヤードータおよび前記第 2 のラインの生ベイヤードータの複数の重複してない 2

× 2 配列の各 2 × 2 配列をマッピングすることと、

前記複数のテクセルのうちの少なくとも 1 つに対して少なくとも 1 つの動作を実行することと

を備える方法。

[C 2] 前記少なくとも 1 つの動作は、ベイヤービニング動作、バイリニアフィルタ動作、読取り動作、または収集動作のうちの少なくとも 1 つを含む、C 1 に記載の方法。

[C 3] 前記複数のテクセルの各テクセルは、R G B A フォーマットである、C 1 に記載の方法。

[C 4] 前記複数のテクセルのうちの少なくとも 1 つに対して前記少なくとも 1 つの動作を実行した後、画像信号プロセッサに前記複数のテクセルを送信することをさらに備える、C 1 に記載の方法。

[C 5] 前記複数のテクセルを提供するために、別個の対応するテクセルとして、前記第 1 のラインの生ベイヤーデータおよび前記第 2 のラインの生ベイヤーデータの複数の重複してない 2 × 2 配列の各 2 × 2 配列を前記マッピングすることは、アプリケーションプログラムインタフェースを介してアクセス可能である、C 1 に記載の方法。

[C 6] 前記生ベイヤーデータが M I P I フォーマットであるかどうかについての決定を行うことと、

前記生ベイヤーデータが M I P I フォーマットであると決定された場合、前記生ベイヤーデータをアンパックするために、グラフィックス処理ユニットのテクスチャパイプラインにおけるハードウェアを用いることと

をさらに備える、C 1 に記載の方法。

[C 7] 前記テクスチャパイプラインにおける前記ハードウェアによって実行される前記生ベイヤーデータを前記アンパックすることは、対応するテクスチャデータタイプを有する、C 6 に記載の方法。

[C 8] 前記テクスチャパイプラインにおける前記ハードウェアによって実行される前記生ベイヤーデータを前記アンパックすることは、アプリケーションプログラムインタフェースを介してアクセス可能である、C 6 に記載の方法。

[C 9] グラフィックス処理のためのデバイスであって、

キャッシュと、

前記キャッシュから第 1 のラインの生ベイヤーデータ、および前記キャッシュから第 2 のラインの生ベイヤーデータを受け取り、

複数のテクセルを提供するために、別個の対応するテクセルとして、前記第 1 のラインの生ベイヤーデータおよび前記第 2 のラインの生ベイヤーデータの複数の重複してない 2 × 2 配列の各 2 × 2 配列をマッピングし、

前記複数のテクセルのうちの少なくとも 1 つに対して少なくとも 1 つの動作を実行する

ように構成されたテクスチャパイプラインと

を備えるデバイス。

[C 1 0] 前記テクスチャパイプラインは、前記少なくとも 1 つの動作が、ベイヤービニング動作、バイリニアフィルタ動作、読取り動作、または収集動作のうちの少なくとも 1 つを含むように構成される、C 9 に記載のデバイス。

[C 1 1] 前記テクスチャパイプラインは、前記複数のテクセルの各テクセルが、R G B A フォーマットであるように構成される、C 9 に記載のデバイス。

[C 1 2] 画像信号プロセッサをさらに備え、ここにおいて、前記画像信号プロセッサは、テクスチャプロセッサが前記複数のテクセルのうちの少なくとも 1 つに対して前記少なくとも 1 つの動作を実行した後に、前記複数のテクセルを受け取るように構成される、C 9 に記載のデバイス。

[C 1 3] 前記テクスチャパイプラインは、アプリケーションプログラムインタフェースを介してアクセス可能である手法で、前記複数のテクセルを提供するために、別個の対応するテクセルとして、前記第 1 のラインの生ベイヤーデータおよび前記第 2 のラインの

生バイヤーデータの複数の重複してない 2×2 配列の各 2×2 配列をマッピングするように構成される、C 9 に記載のデバイス。

[C 1 4] 前記テキストチャイプラインは、

前記生バイヤーデータが M I P I フォーマットであるかどうかについての決定を行い、
前記生バイヤーデータが M I P I フォーマットであると決定された場合、前記生バイヤーデータをアンパックする

ようにさらに構成される、C 9 に記載のデバイス。

[C 1 5] 前記テキストチャイプラインは、前記アンパックすることが、対応するテキストチャデータタイプを有するように、前記生バイヤーデータをアンパックすることが達成されるように構成される、C 1 4 に記載のデバイス。

[C 1 6] 前記テキストチャイプラインは、前記生バイヤーデータをアンパックすることがアプリケーションプログラムインタフェースを介してアクセス可能であるように構成される、C 1 4 に記載のデバイス。

[C 1 7] 前記キャッシュは、レベル 2 キャッシュであり、前記テキストチャイプラインは、

ローカルキャッシュと、

前記レベル 2 キャッシュから前記第 1 のラインの生バイヤーデータ、および前記レベル 2 キャッシュから前記第 2 のラインの生バイヤーデータを受け取り、前記複数のテクセルを提供するために、別個の対応するテクセルとして、前記第 1 のラインの生バイヤーデータおよび前記第 2 のラインの生バイヤーデータの前記複数の重複してない 2×2 配列の各 2×2 配列をマッピングし、前記ローカルキャッシュに前記複数のテクセルを送信するように構成されたアンパッキング回路と

を含む、C 9 に記載のデバイス。

[C 1 8] 前記アンパッキング回路は、

前記生バイヤーデータが M I P I フォーマットであるかどうかについての決定を行い、
前記生バイヤーデータが M I P I フォーマットであると決定された場合、前記生バイヤーデータをアンパックする

ようにさらに構成される、C 1 7 に記載のデバイス。

[C 1 9] 前記アンパッキング回路は、別個のキャッシュを含む、C 1 7 に記載のデバイス。

[C 2 0] グラフィックス処理のためのデバイスであって、

第 1 のラインの生バイヤーデータおよび第 2 のラインの生バイヤーデータを受け取るための手段と、

複数のテクセルを提供するために、別個の対応するテクセルとして、前記第 1 のラインの生バイヤーデータおよび前記第 2 のラインの生バイヤーデータの複数の重複してない 2×2 配列の各 2×2 配列をマッピングするための手段と、

前記複数のテクセルのうちの少なくとも 1 つに対して少なくとも 1 つの動作を実行するための手段と

を備えるデバイス。

[C 2 1] 前記少なくとも 1 つの動作は、バイヤービニング動作、バイリニアフィルタ動作、読取り動作、または収集動作のうちの少なくとも 1 つを含む、C 2 0 に記載のデバイス。

[C 2 2] 前記複数のテクセルの各テクセルは、R G B A フォーマットである、C 2 0 に記載のデバイス。

[C 2 3] 前記複数のテクセルのうちの少なくとも 1 つに対して前記少なくとも 1 つの動作を実行した後、画像信号プロセッサに前記複数のテクセルを送信するための手段をさらに備える、C 2 0 に記載のデバイス。

[C 2 4] 前記複数のテクセルを提供するために、別個の対応するテクセルとして、前記第 1 のラインの生バイヤーデータおよび前記第 2 のラインの生バイヤーデータの複数の重複してない 2×2 配列の各 2×2 配列を前記マッピングすることは、アプリケーション

プログラムインタフェースを介してアクセス可能である、C 2 0 に記載のデバイス。

[C 2 5] 前記生バイヤーデータが M I P I フォーマットであるかどうかについての決定を行うための手段と、

前記生バイヤーデータが M I P I フォーマットであると決定された場合、前記生バイヤーデータをアンパックするために、グラフィックス処理ユニットのテクスチャパイプラインにおけるハードウェアを用いるための手段と

をさらに備える、C 2 0 に記載のデバイス。

[C 2 6] 前記テクスチャパイプラインにおける前記ハードウェアによって実行される前記生バイヤーデータを前記アンパックすることは、対応するテクスチャデータタイプを有する、C 2 5 に記載のデバイス。

[C 2 7] 前記テクスチャパイプラインにおける前記ハードウェアによって実行される前記生バイヤーデータを前記アンパックすることは、アプリケーションプログラムインタフェースを介してアクセス可能である、C 2 5 に記載のデバイス。

【 国際調査報告 】

INTERNATIONAL SEARCH REPORT

International application No

PCT/US2016/013468

A. CLASSIFICATION OF SUBJECT MATTER

INV. G06T3/40
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06T

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	MORGAN MCGUIRE: "Efficient, High-Quality Bayer Demosaic Filtering on GPUs", JOURNAL OF GRAPHICS TOOLS, ASSOCIATION FOR COMPUTING MACHINERY, NEW YORK, US, vol. 13, no. 4, 1 January 2008 (2008-01-01), pages 1-16, XP009189618, abstract page 6, paragraph 1 ----- -/--	1-27

☒ Further documents are listed in the continuation of Box C.☐ See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

2 May 2016

Date of mailing of the international search report

11/05/2016

Name and mailing address of the ISA/

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040,
Fax: (+31-70) 340-3016

Authorized officer

Pierfederici, A

INTERNATIONAL SEARCH REPORT

International application No

PCT/US2016/013468

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	<p>"glTexImage2D - specify a two-dimensional texture image",</p> <p>2006, pages 1-10, XP55266843, Retrieved from the Internet: URL:https://www.opengl.org/sdk/docs/man/html/glTexImage2D.xhtml [retrieved on 2016-04-20] page 3, paragraph 2</p>	1-27
X	<p>-----</p> <p>PATRIK GOORTS ET AL: "Raw Camera Image Demosaicing using Finite Impulse Response Filtering on Commodity GPU Hardware using CUDA", SIGMAP AND WINSYS 2012, 1 January 2012 (2012-01-01), pages 96-101, XP55267938, Title; page 98, left-hand column, paragraph 3 page 99, left-hand column, paragraph 3</p>	1-27
A	<p>-----</p> <p>SHANE RYOO ET AL: "Optimization principles and application performance evaluation of a multithreaded GPU using CUDA", PROCEEDINGS OF THE 13TH ACM SIGPLAN SYMPOSIUM ON PRINCIPLES AND PRACTICE OF PARALLEL PROGRAMMING , PPOPP '08, 1 January 2008 (2008-01-01), pages 73-82, XP055042951, New York, New York, USA page 77, left-hand column, last paragraph - page 78, left-hand column, paragraph 1</p>	1-27
A	<p>-----</p> <p>PATRIK GOORTS ET AL: "Optimal Data Distribution for Versatile Finite Impulse Response Filtering on Next-Generation Graphics Hardware Using CUDA", PARALLEL AND DISTRIBUTED SYSTEMS (ICPADS), 2009 15TH INTERNATIONAL CONFERENCE ON, IEEE, PISCATAWAY, NJ, USA, 8 December 2009 (2009-12-08), pages 300-307, XP031616972, page 300, left-hand column, paragraph 1 - right-hand column, paragraph 1 page 303, left-hand column, paragraph 2</p> <p>-----</p> <p>-/--</p>	1-27

INTERNATIONAL SEARCH REPORT

International application No

PCT/US2016/013468

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	RAGNAR LANGSETH ET AL: "An Evaluation of Debayering Algorithms on GPU for Real-Time Panoramic Video Recording", 2014 IEEE INTERNATIONAL SYMPOSIUM ON MULTIMEDIA, 1 December 2014 (2014-12-01), pages 110-115, XP55267940, page 112, left-hand column, paragraph 3 - right-hand column, paragraph 2 -----	1-27
A	Sarah J Fortune: "Darwin College Research Report GPU-Based Raw Digital Photo Manipulation", Darwin College, Cambridge University, 1 June 2010 (2010-06-01), pages 1-79, XP55267916, Retrieved from the Internet: URL:https://www.darwin.cam.ac.uk/drupal7/sites/default/files/Documents/publications/dcrr011.pdf [retrieved on 2016-04-22] page 23, paragraph 2 - page 29, paragraph 1; figure 2.8 -----	1-27

フロントページの続き

(81)指定国 AP(BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), EA(AM, AZ, BY, KG, KZ, RU, TJ, TM), EP(AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OA(BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG), AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US

(72)発明者 ユン、ジェイ・チュンスブ

アメリカ合衆国、カリフォルニア州 9 2 1 2 1 - 1 7 1 4、サン・ディエゴ、モアハウス・ドライブ 5 7 7 5

(72)発明者 リ、リアン

アメリカ合衆国、カリフォルニア州 9 2 1 2 1 - 1 7 1 4、サン・ディエゴ、モアハウス・ドライブ 5 7 7 5

(72)発明者 ガヌガパティ、ピジャイ

アメリカ合衆国、カリフォルニア州 9 2 1 2 1 - 1 7 1 4、サン・ディエゴ、モアハウス・ドライブ 5 7 7 5

(72)発明者 ジャン、シュジエ

アメリカ合衆国、カリフォルニア州 9 2 1 2 1 - 1 7 1 4、サン・ディエゴ、モアハウス・ドライブ 5 7 7 5

F ターム(参考) 5B057 BA02 BA12 CA12 CA16 CB12 CB16 CG10 CH05 CH09 CH14