

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第6884766号
(P6884766)

(45) 発行日 令和3年6月9日 (2021. 6. 9)

(24) 登録日 令和3年5月14日 (2021. 5. 14)

(51) Int. Cl.	F I
GO 6 F 12/02 (2006. 01)	GO 6 F 12/02 5 3 0 C
GO 6 F 12/00 (2006. 01)	GO 6 F 12/00 5 8 0
GO 6 F 12/04 (2006. 01)	GO 6 F 12/04 5 3 0
GO 6 T 1/20 (2006. 01)	GO 6 T 1/20 A

請求項の数 15 (全 28 頁)

(21) 出願番号	特願2018-513758 (P2018-513758)	(73) 特許権者	595020643
(86) (22) 出願日	平成28年8月1日 (2016. 8. 1)		クアルコム・インコーポレイテッド
(65) 公表番号	特表2018-529168 (P2018-529168A)		QUALCOMM INCORPORATED
(43) 公表日	平成30年10月4日 (2018. 10. 4)		アメリカ合衆国、カリフォルニア州 92
(86) 国際出願番号	PCT/US2016/044975		121-1714、サン・ディエゴ、モア
(87) 国際公開番号	W02017/048381		ハウス・ドライブ 5775
(87) 国際公開日	平成29年3月23日 (2017. 3. 23)	(74) 代理人	100108855
審査請求日	令和1年7月11日 (2019. 7. 11)		弁理士 蔵田 昌俊
(31) 優先権主張番号	14/857, 303	(74) 代理人	100109830
(32) 優先日	平成27年9月17日 (2015. 9. 17)		弁理士 福原 淑弘
(33) 優先権主張国・地域又は機関	米国 (US)	(74) 代理人	100158805
早期審査対象出願			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】 帯域幅圧縮グラフィックスデータの記憶

(57) 【特許請求の範囲】

【請求項 1】

方法であって、

少なくとも1つのプロセッサが、複数の帯域幅圧縮グラフィックスデータを、メモリ内のそれぞれの複数のブロックに記憶することと、

ここで、前記記憶することは、前記複数の帯域幅圧縮グラフィックスデータのうちの第1の1つまたは複数の帯域幅圧縮グラフィックスデータを、前記複数のブロックのうちの第1の1つまたは複数のブロックに記憶することと、前記複数の帯域幅圧縮グラフィックスデータのうちの第2の1つまたは複数の帯域幅圧縮グラフィックスデータを、前記複数のブロックのうちの第2の1つまたは複数のブロックに記憶することとを含み、

ここにおいて、前記複数のブロックの各々は、前記複数の帯域幅圧縮グラフィックスデータのうちの最大のものを完全に含むことができる、前記メモリにおいて均一の固定サイズであり、

ここにおいて、前記複数の帯域幅圧縮グラフィックスデータのうちの前記第1の1つまたは複数の帯域幅圧縮グラフィックスデータの各々は、前記固定サイズより小さいサイズを有し、

ここにおいて、前記複数の帯域幅圧縮グラフィックスデータのうちの前記第2の1つまたは複数の帯域幅圧縮グラフィックスデータの各々は、前記複数のブロックのうちの前記第2の1つまたは複数のブロックの各々に完全に記憶される、

10

20

前記少なくとも１つのプロセッサが、前記複数の帯域幅圧縮グラフィックスデータに関連付けられたデータを、前記プロセッサが、前記複数の帯域幅圧縮グラフィックスデータに対するグラフィックス動作の性能を最適化するために前記データを使用するように、前記複数のブロックのうちの前記第１の１つまたは複数のブロックが未使用空間を含むとの決定に少なくとも部分的に基づいて、前記複数の帯域幅圧縮グラフィックスデータのうちの前記第１の１つまたは複数の帯域幅圧縮グラフィックスデータを含む、前記複数のブロックのうちの前記第１の１つまたは複数のブロックの前記未使用空間に、記憶することとを備える方法。

【請求項２】

前記複数の帯域幅圧縮グラフィックスデータに関連付けられた前記データは、前記複数のブロックのうちの前記第１の１つまたは複数のブロックに記憶された前記複数の帯域幅圧縮グラフィックスデータのうちの前記第１の１つまたは複数の帯域幅圧縮グラフィックスデータについての深度データを備える、請求項１に記載の方法。

10

【請求項３】

前記少なくとも１つのプロセッサが、前記複数のブロックのうちの前記第２の１つまたは複数のブロックを完全に占有する、前記複数の帯域幅圧縮グラフィックスデータのうちの前記第２の１つまたは複数の帯域幅圧縮グラフィックスデータの各々についてのデフォルトの深度値に関連付けることをさらに備える、請求項２に記載の方法。

【請求項４】

前記複数の帯域幅圧縮グラフィックスデータに関連付けられた前記データは、前記複数のブロックのうちの前記第１の１つまたは複数のブロックに記憶された前記複数の帯域幅圧縮グラフィックスデータのうちの前記第１の１つまたは複数の帯域幅圧縮グラフィックスデータの各々を識別する１つまたは複数のハッシュコードを備える、または、

20

前記複数の帯域幅圧縮グラフィックスデータに関連付けられた前記データは、前記複数のブロックに記憶された前記複数の帯域幅圧縮グラフィックスデータの各々を識別するハッシュコードを備える、請求項１に記載の方法。

【請求項５】

前記複数の帯域幅圧縮グラフィックスデータに関連付けられた前記データは、前記複数のブロックに記憶された前記複数の帯域幅圧縮グラフィックスデータの各々を識別するハッシュコードと、前記複数のブロックに記憶された前記複数の帯域幅圧縮グラフィックスデータの各々についての深度データとを備える、請求項１に記載の方法。

30

【請求項６】

前記複数の帯域幅圧縮グラフィックスデータに関連付けられた前記データは、前記複数の帯域幅圧縮グラフィックスデータのレンダリングを最適化するために使用される前記複数の帯域幅圧縮グラフィックスデータに関連付けられたサーフェスデータを備える、請求項１に記載の方法。

【請求項７】

前記複数の帯域幅圧縮グラフィックスデータは、画像サーフェスの帯域幅圧縮部分を備える、請求項１に記載の方法。

【請求項８】

40

前記少なくとも１つのプロセッサは、グラフィックス処理ユニットを含む、請求項１に記載の方法。

【請求項９】

装置であって、

複数の帯域幅圧縮グラフィックスデータを、メモリ内のそれぞれの複数のブロックに記憶するための手段と、

ここで、前記記憶するための手段は、前記複数の帯域幅圧縮グラフィックスデータのうちの第１の１つまたは複数の帯域幅圧縮グラフィックスデータを、前記複数のブロックのうちの第１の１つまたは複数のブロックに記憶するための手段と、

前記複数の帯域幅圧縮グラフィックスデータのうちの第２の１つまたは複数の帯域幅

50

圧縮グラフィックスデータを、前記複数のブロックのうちの第2の1つまたは複数のブロックに記憶するための手段とを含み、

ここにおいて、前記メモリ内の前記複数のブロックの各々は、均一の固定サイズであり、

ここにおいて、前記複数の帯域幅圧縮グラフィックスデータのうちの前記第1の1つまたは複数の帯域幅圧縮グラフィックスデータの各々は、前記固定サイズより小さいサイズを有し、

ここにおいて、前記複数の帯域幅圧縮グラフィックスデータのうちの前記第2の1つまたは複数の帯域幅圧縮グラフィックスデータの各々は、前記複数のブロックのうちの前記第2の1つまたは複数のブロックの各々に完全に記憶される、

10

前記複数の帯域幅圧縮グラフィックスデータに関連付けられたデータを、前記複数の帯域幅圧縮グラフィックスデータに対するグラフィックス動作の性能を最適化するために、前記データが使用されるように、前記複数の帯域幅圧縮グラフィックスデータのうちの前記第1の1つまたは複数の帯域幅圧縮グラフィックスデータを含む、前記複数のブロックのうちの前記第1の1つまたは複数のブロックの未使用空間に記憶するための手段とを備える、装置。

【請求項10】

前記複数の帯域幅圧縮グラフィックスデータに関連付けられた前記データは、前記複数のブロックのうちの前記第1の1つまたは複数のブロックに記憶された前記複数の帯域幅圧縮グラフィックスデータのうちの前記第1の1つまたは複数の帯域幅圧縮グラフィックスデータについての深度データを備える、請求項9に記載の装置。

20

【請求項11】

前記複数のブロックのうちの前記第2の1つまたは複数のブロックを完全に占有する、前記複数の帯域幅圧縮グラフィックスデータのうちの前記第2の1つまたは複数の帯域幅圧縮グラフィックスデータの各々についてのデフォルトの深度値に関連付けるための手段をさらに備える、請求項10に記載の装置。

【請求項12】

前記複数の帯域幅圧縮グラフィックスデータに関連付けられた前記データは、前記複数のブロックに記憶された前記複数の帯域幅圧縮グラフィックスデータの各々についての深度データを備える、請求項9に記載の装置。

30

【請求項13】

前記複数の帯域幅圧縮グラフィックスデータに関連付けられた前記データは、前記複数のブロックのうちの前記第1の1つまたは複数のブロックに記憶された前記複数の帯域幅圧縮グラフィックスデータのうちの前記第1の1つまたは複数の帯域幅圧縮グラフィックスデータの各々を識別する1つまたは複数のハッシュコードを備える、請求項9に記載の装置。

【請求項14】

前記複数の帯域幅圧縮グラフィックスデータに関連付けられた前記データは、前記複数のブロックに記憶された前記複数の帯域幅圧縮グラフィックスデータの各々を識別するハッシュコードを備える、請求項9に記載の装置。

40

【請求項15】

前記複数の帯域幅圧縮グラフィックスデータに関連付けられた前記データは、前記複数のブロックに記憶された前記複数の帯域幅圧縮グラフィックスデータの各々を識別するハッシュコードと、前記複数のブロックに記憶された前記複数の帯域幅圧縮グラフィックスデータの各々についての深度データとを備える、請求項9に記載の装置。

【発明の詳細な説明】

【技術分野】

【0001】

[0001]本開示は、データ記憶に関し、より具体的には、帯域幅圧縮グラフィックスデータをメモリに記憶することに関する。

50

【背景技術】

【0002】

[0002]電子ディスプレイ上での視覚的表現のためのコンテンツを提供するデバイスは一般に、グラフィックス処理ユニット（GPU）を含む。GPUは、ディスプレイ上のコンテンツを表すピクセルをレンダリングする。GPUは、ディスプレイ上のピクセルごとに1つ以上のピクセル値を生成し、ラスタライズ段で生成されるフラグメントのフラグメントシェーディングを実行する提示のために各ピクセルをレンダリングするために、ディスプレイ上の各ピクセルについてのピクセル値に対してグラフィックス処理を実行する。

【発明の概要】

【0003】

[0003]本開示の技法は一般に、複数の帯域幅圧縮グラフィックスデータを、この複数の帯域幅圧縮グラフィックスデータに関連付けられた追加データとともにメモリに記憶するための技法に関する。複数の帯域幅圧縮グラフィックスデータは、サイズがさまざまであり得、複数の帯域幅圧縮グラフィックスデータは、複数の帯域幅圧縮グラフィックスデータのうちの最大の帯域幅圧縮グラフィックスデータを収納し得るメモリ内の均一なサイズのブロックに記憶される。したがって、複数の帯域幅圧縮グラフィックスデータをメモリ内の均一なサイズのブロックに記憶することは、複数の帯域幅圧縮グラフィックスデータを記憶するメモリ内のブロックのうちのいくつか中の残存した未使用空間に帰着する。メモリ内のブロックのうちのいくつか中のそのような未使用空間は、複数の帯域幅圧縮グラフィックスデータに関連付けられた深度データまたは複数の帯域幅圧縮グラフィックスデータの各々を識別するハッシュコードのような、複数の帯域幅圧縮グラフィックスデータに関連付けられた追加データを記憶するために利用されることができる。

【0004】

[0004]本開示の一例では、グラフィックス処理のための方法は、少なくとも1つのプロセッサが、複数の帯域幅圧縮グラフィックスデータを、メモリ内のそれぞれの複数のブロックに記憶することを含み得、ここにおいて、メモリ内の複数のブロックの各々は、均一の固定サイズであり、複数の帯域幅圧縮グラフィックスデータのうちの1つ以上は、固定サイズより小さいサイズを有する。プロセスは、少なくとも1つのプロセッサが、複数の帯域幅圧縮グラフィックスデータに関連付けられたデータを、複数の帯域幅圧縮グラフィックスデータのそれぞれの1つ以上を含む、複数のブロックのうちの1つ以上のブロックの未使用空間に記憶することをさらに含むことができる。

【0005】

[0005]本開示の別の例では、グラフィックスデータを処理するように構成された装置は、メモリを含むことができる。装置は、複数の帯域幅圧縮グラフィックスデータを、メモリ内のそれぞれの複数のブロックに記憶することと、ここにおいて、メモリ内の複数のブロックの各々は、均一の固定サイズであり、複数の帯域幅圧縮グラフィックスデータのうちの1つ以上は、固定サイズより小さいサイズを有する、複数の帯域幅圧縮グラフィックスデータに関連付けられたデータを、複数の帯域幅圧縮グラフィックスデータのそれぞれの1つ以上を含む、複数のブロックのうちの1つ以上のブロックの未使用空間に記憶することと、を行うように構成された少なくとも1つのプロセッサをさらに含むことができる。

【0006】

[0006]本開示の別の例では、装置は、複数の帯域幅圧縮グラフィックスデータを、メモリ内のそれぞれの複数のブロックに記憶するための手段を含み得、ここにおいて、メモリ内の複数のブロックの各々は、均一の固定サイズであり、複数の帯域幅圧縮グラフィックスデータのうちの1つ以上は、固定サイズより小さいサイズを有する。装置は、複数の帯域幅圧縮グラフィックスデータに関連付けられたデータを、複数の帯域幅圧縮グラフィックスデータのそれぞれの1つ以上を含む、複数のブロックのうちの1つ以上のブロックの未使用空間に記憶するための手段をさらに含むことができる。

【0007】

[0007] 1つ以上の例の詳細は、添付の図面および以下の説明において示される。他の特徴、目的、および利点は、本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】

【0008】

【図1】[0008]図1は、帯域幅圧縮グラフィカルデータをメモリに記憶するための、本開示の1つ以上の態様を実施するように構成され得る例となるコンピューティングデバイスを例示するブロック図である。

【図2】[0009]図2は、図1のCPU、GPU、およびシステムメモリの例となる実施をさらに詳細に例示するブロック図である。

【図3A】[0010]図3Aは、帯域幅圧縮グラフィカルデータをメモリに記憶するための例となる技法を例示する概念図である。

【図3B】図3Bは、帯域幅圧縮グラフィカルデータをメモリに記憶するための例となる技法を例示する概念図である。

【図3C】図3Cは、帯域幅圧縮グラフィカルデータをメモリに記憶するための例となる技法を例示する概念図である。

【図3D】図3Dは、帯域幅圧縮グラフィカルデータをメモリに記憶するための例となる技法を例示する概念図である。

【図3E】図3Eは、帯域幅圧縮グラフィカルデータをメモリに記憶するための例となる技法を例示する概念図である。

【図3F】図3Fは、帯域幅圧縮グラフィカルデータをメモリに記憶するための例となる技法を例示する概念図である。

【図4】[0011]図4は、帯域幅圧縮グラフィカルデータをメモリに記憶するための例となるプロセスを例示するフローチャートである。

【発明の詳細な説明】

【0009】

[0012]帯域幅圧縮グラフィックスデータは、それがコンピューティングデバイスのバスを通じてより速く転送され得るように圧縮されたグラフィックスデータである。コンピューティングデバイスのグラフィックス処理ユニット(GPU)が、サーフェスのようなグラフィックスデータに対してグラフィックス処理動作を実行するため、コンピューティングデバイスは、GPUとメモリとの間でまたは異なるメモリ間でバスを通じてサーフェスを転送することができる。たとえば、コンピューティングデバイスは、2つの異なるサーフェスを組み合わせる合成動作を、この合成動作を実行するためにメモリからGPUにこれら2つのサーフェスを転送し、結果として得られた合成サーフェスをGPUからメモリに転送し戻すことで実行することができる。ゆえに、圧縮を介してサーフェスのサイズを縮小することで、コンピューティングデバイスは、コンピューティングデバイスの構成要素間でサーフェスをより速く転送し得、それによって、コンピューティングデバイスの性能を高める。

【0010】

[0013]コンピューティングデバイスは、複数の帯域幅圧縮グラフィックスデータを生成するために、サーフェスをサブ領域に分割し、サーフェスのサブ領域の各々を圧縮することで、サーフェスの帯域幅圧縮を実行することができる。複数の帯域幅圧縮グラフィックスデータは、サーフェスのサブ領域間のコンテンツの違いによりサイズがさまざまであり得る。たとえば、コンピューティングデバイスは、単一色のピクセルを均一に含むサーフェスのサブ領域を、多くの異なる色のピクセルを含むサーフェスの別のサブ領域より比較的小さいサイズに圧縮することができ得る。

【0011】

[0014]コンピューティングデバイスは、複数の帯域幅圧縮グラフィックスデータを、メモリ内の、コンピューティングデバイスが割り当てる複数の均一なサイズのブロックに記憶することができる。ブロックの各々は、複数の帯域幅圧縮グラフィックスデータのうち

10

20

30

40

50

の最大のものを含むのに十分な大きさである。複数の帯域幅圧縮グラフィックスデータはサイズがさまざまであり得るのに対し、複数のブロックの各々は同じサイズであるため、複数の帯域幅圧縮グラフィックスデータを複数のブロックに記憶することは、ブロックに記憶されたそれぞれの帯域幅圧縮グラフィックスデータによって占有されていない未使用空間を各々有する、複数のブロックのうちの1つ以上のブロックに帰着する。

【0012】

[0015]本開示の態様にしたがって、コンピューティングデバイスは、複数の帯域幅圧縮グラフィックスデータに関連付けられた他のデータを、複数のブロックのうちの1つ以上のブロックの未使用空間に記憶することができる。たとえば、コンピューティングデバイスは、複数の帯域幅圧縮グラフィックスデータに関連付けられた深度データを、メモリ内の別個のエリア（たとえば、ブロック）に記憶しないで、代わりに、そのような深度データを、複数のブロックのうちの1つ以上のブロックの未使用空間に記憶することができる。同様に、コンピューティングデバイスは、複数の帯域幅圧縮グラフィックスデータの各々を識別するハッシュコードを、複数のブロックのうちの1つ以上のブロックの未使用空間に記憶することができる。このように、コンピューティングデバイスは、複数の帯域幅圧縮グラフィックスデータに関連付けられた追加のデータを記憶するために複数のブロック内の未使用空間を利用することができ、それによって、コンピューティングデバイスのメモリ利用効率を高める。

【0013】

[0016]コンピューティングデバイスが1つ以上のブロックの未使用空間に記憶し得るそれ以外のデータは、コンピューティングデバイスが、そのようなデータを、グラフィックスデータに対するグラフィックス動作の性能を最適化するために使用し得る点で、最適化サーフェスであり得る。たとえば、コンピューティングデバイスは、関連するグラフィックスデータをレンダリングする際のその性能を高めるために深度データを利用し得、コンピューティングデバイスは、グラフィックスデータに対する特定のグラフィカル動作のその性能を高めるためにハッシュコードを利用することができる。そのため、コンピューティングデバイスは、深度データまたはハッシュコード以外の任意の数の追加データを、1つ以上のブロックの未使用空間に記憶し得、それは、グラフィックスデータのレンダリングを最適化するために使用され得る追加の最適化サーフェスを記憶することを含む。

【0014】

[0017]図1は、帯域幅圧縮グラフィカルデータをメモリに記憶するための、本開示の1つ以上の態様を実施するように構成され得る例となるコンピューティングデバイスを例示するブロック図である。図1に示されるように、デバイス2は、ビデオデバイス、メディアプレーヤ、セットトップボックス、モバイル電話およびいわゆるスマートフォンのようなワイヤレスハンドセット、携帯情報端末（PDA）、デスクトップコンピュータ、ラップトップコンピュータ、ゲーミングコンソール、ビデオ会議ユニット、タブレットコンピューティングデバイス、等を含むがそれらに限定されないコンピューティングデバイスであり得る。図1の例では、デバイス2は、中央処理装置（CPU）6と、システムメモリ10と、GPU12とを含むことができる。デバイス2はまた、ディスプレイプロセッサ14と、トランシーバモジュール3と、ユーザインターフェース4と、ディスプレイ8とを含むことができる。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、CPU6および/またはGPU12を含む1つ以上のICの外部に両方があり得るか、CPU6および/またはGPU12を含むICの外部にあるIC内に形成されることができる。

【0015】

[0018]デバイス2は、明確さのために図1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むことができる。たとえば、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では電話通信を達成するためにスピーカおよびマイクロフォンを、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、それらはいずれも図1に

10

20

30

40

50

示されていない。デバイス 2 はまた、ビデオカメラを含み得る。さらに、デバイス 2 に示されるさまざまなモジュールおよびユニットは、必ずしも、デバイス 2 のすべての例に必要なというわけではない。たとえば、ユーザインターフェース 4 およびディスプレイ 8 は、デバイス 2 が、外部ユーザインターフェースまたはディスプレイとインターフェース接続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。

【 0 0 1 6 】

[0019]ユーザインターフェース 4 の例には、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が含まれるがそれらに限定されない。ユーザインターフェース 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ることができる。トランシーバモジュール 3 は、コンピューティングデバイス 2 と別のデバイスまたはネットワークとの間でのワイヤレスおよびワイヤード通信を可能にする回路を含むことができる。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むことができる。

【 0 0 1 7 】

[0020]CPU 6 は、実行のための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装置 (CPU) のような、マイクロプロセッサであり得る。CPU 6 は、コンピューティングデバイス 2 の動作を制御する汎用または専用プロセッサを備えることができる。ユーザは、1 つ以上のソフトウェアアプリケーションを実行することを CPU 6 に行わせるために、コンピューティングデバイス 2 に入力を提供することができる。CPU 6 上で実行されるソフトウェアアプリケーションには、たとえば、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ーフェースアプリケーション、または別のプログラムが含まれることができる。追加的に、CPU 6 は、GPU 12 の動作を制御するための GPU ドライバ 22 を実行することができる。ユーザは、キーボード、マウス、マイクロフォン、タッチパッド、またはユーザインターフェース 4 を介してコンピューティングデバイス 2 に結合された別の入力デバイスのような 1 つ以上の入力デバイス (図示されない) を介してコンピューティングデバイス 2 に入力を提供することができる。

【 0 0 1 8 】

[0021]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させるように CPU 6 に命令する 1 つ以上のグラフィックスレンダリング命令を含むことができる。いくつかの例では、ソフトウェア命令は、たとえば、OpenGL (Open Graphics Library) グラフィックスアプリケーションプログラミングインターフェース (API)、OpenGL ES (Open Graphics Library Embedded Systems) API、Direct3D API、X3D API、RenderMan API、WebGL API、あるいは任意の他の公のまたは専有の標準的なグラフィックス API のような、API に従うことができる。

【 0 0 1 9 】

[0022]ソフトウェアアプリケーションの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、グラフィックスデータのレンダリングのうちのいくつかまたはすべてを実行することを GPU 12 に行わせるために、1 つ以上のグラフィックスレンダリングコマンドを (たとえば、GPU ドライバ 22 を通じて) GPU 12 に発行することができる。いくつかの例では、レンダリングされるべきグラフィックスデータは、たとえば、点、線、三角形、四角形、トライアングルストリップストリップ (triangle strip)、等のグラフィックスプリミティブのリストを含むことができる。

【 0 0 2 0 】

[0023]GPU 12 は、ディスプレイ 8 に 1 つ以上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ることができる。ゆえに、CPU 6 上で実行される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

10

20

30

40

50

ス処理を必要とするとき、CPU 6 は、ディスプレイ 8 へのレンダリングのために、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 GPU 12 に提供することができる。グラフィックスデータは、たとえば、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、等を含むことができる。GPU 12 は、いくつかの事例では、CPU 6 より効率的な、複雑なグラフィック関連動作の処理を提供する高度な並列構造で築かれることができる。たとえば、GPU 12 は、並列的に、複数の頂点またはピクセルに作用するように構成された、シェーダユニットのような複数の処理要素を含むことができる。GPU 12 の高度に並列な性質は、いくつかの事例では、GPU 12 が、グラフィックス画像（たとえば、GUI および 2 次元（2D）および / または 3 次元（3D）グラフィックスシーン）をディスプレイ 8 上に、CPU 6 を用いてこれらのシーンを直接ディスプレイ 8 上に描画するより迅速に描画することを可能にすることができる。

10

【0021】

[0024] GPU 12 は、いくつかの事例では、コンピューティングデバイス 2 のマザーボードに一体化されることができる。他の事例では、GPU 12 は、コンピューティングデバイス 2 のマザーボード内のポートに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、コンピューティングデバイス 2 と相互作用するように構成された周辺デバイス内に他の方法で組み込まれることができる。GPU 12 は、1 つ以上の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ゲートアレイ（FPGA）、デジタルシグナルプロセッサ（DSP）、あるいは他の同等の集積回路またはディスクリット論理回路といった 1 つ以上のプロセッサを含むことができる。GPU 12 はまた、1 つ以上のプロセッサコアを含み得、それにより、GPU 12 は、マルチコアプロセッサと呼ばれることができる。

20

【0022】

[0025] GPU 12 は、グラフィックスメモリ 40 に直接結合されることができる。ゆえに、GPU 12 は、バスを使用することなく、グラフィックスメモリ 40 からデータを読み取り、それにデータを書き込むことができる。換言すると、GPU 12 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することができる。このようなグラフィックスメモリ 40 は、オンチップメモリと呼ばれることができる。これは、ヘビーバストラフィックを経験し得る、GPU 12 がバスを介してデータを読み取りおよび書き込む必要性をなくすことで、GPU 12 がより効率的に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、GPU 12 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 10 を利用することができる。グラフィックスメモリ 40 は、たとえば、ランダムアクセスメモリ（RAM）、静的メモリ（SRAM）、動的 RAM（DRAM）、消去可能なプログラマブル ROM（EPROM）、電気的消去可能なプログラマブル ROM（EEPROM（登録商標））、フラッシュメモリ、磁気データ媒体、あるいは光学記憶媒体のような、1 つ以上の揮発性および不揮発性メモリまたは記憶デバイスを含むことができる。

30

【0023】

[0026] いくつかの例では、GPU 12 は、完全形成画像（fully formed image）を、この画像が 1 つ以上のサーフェスであり得るシステムメモリ 10 に記憶することができる。サーフェスは、いくつかの例では、ピクセルの 2 次元ブロックであり得、ここで、ピクセルの各々が、カラー値を有することができる。本開示全体にわたって、グラフィックスデータという用語は、非限定的な例では、サーフェスまたはサーフェスの一部を含む。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 のピクセルを点灯させる値を出力することができる。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、コンピューティングデバイス 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。

40

【0024】

50

[0027]本開示の態様にしたがって、コンピューティングデバイス 2 は、システムメモリ 10 またはグラフィックスメモリ 40 のようなメモリ内の複数のブロックを割当てることができる。ここにおいて、メモリ内の複数のブロックの各々は、均一の固定サイズである。コンピューティングデバイス 2 は、複数の帯域幅圧縮グラフィックスデータを、メモリ内のそれぞれの複数のブロックにさらに記憶し得、ここにおいて、複数の帯域幅圧縮グラフィックスデータのうちの 1 つ以上は、固定サイズより小さいサイズを有する。コンピューティングデバイス 2 は、複数の帯域幅圧縮グラフィックスデータに関連付けられたデータを、複数の帯域幅圧縮グラフィックスデータのそれぞれの 1 つ以上を含む、複数のブロックのうちの 1 つ以上のブロックの未使用空間にさらに記憶することができる。

【0025】

10

[0028]図 2 は、図 1 の CPU 6、GPU 12、およびシステムメモリ 10 の例となる実施をさらに詳細に例示するブロック図である。図 2 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 と、グラフィックス API 20 と、GPU ドライバ 22 とを含み得、これらの各々は、CPU 6 上で実行される 1 つ以上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。

【0026】

[0029]CPU 6 および GPU 12 に利用可能なメモリには、システムメモリ 10、フレームバッファ 16、およびレンダターゲット 24 が含まれ得る。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、システムメモリ 10 とは別個であり得、レンダリングされた画像データを記憶することができる。GPU 12 もまた、レンダターゲット 24 への記憶のためのデータをレンダリングすることができる。フレームバッファ 16 と同様に、レンダターゲット 24 は、システムメモリ 10 の一部であり得るか、システムメモリ 10 とは別個であり得る。

20

【0027】

[0030]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。たとえば、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。

30

【0028】

[0031]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ーフェース (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 GPU 12 に命令する 1 つ以上の描画命令を含むことができる。たとえば、描画命令には、GPU 12 によってレンダリングされるべき 1 つ以上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が含まれることができる。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 surface) のすべてまたは一部を定義することができる。追加の例では、描画命令は、集合的に、これら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 つ以上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すべてまたは一部を定義することができる。

40

【0029】

[0032]ソフトウェアアプリケーション 18 は、1 つ以上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にレンダリングするための 1 つ以上のコマンドを GPU 12 に発するため、グラフィックス API 20 を介して、GPU ドライバ 22 を起動することができる。たとえば、ソフトウェアアプリケーション 18 は、GPU 12 にプリミティブ定義を提供するために、グラフィックス API 20 を介して、GPU ドライバ 22 を起動することができる。いくつかの事例では、プリミティブ定義が、たとえば、三角形、四角形、トライアングルファン (triangle fan)、トライアングルストリップ、等の描画プリミティブのリストの形式で、GPU 12 に提供されることができる。プリミティブ定義は、レンダリングされるべきプリミティブに関連付けられた 1 つ以上の頂点を指定

50

する頂点仕様 (vertex specification) を含むことができる。頂点仕様は、各頂点についての位置座標と、いくつかの事例では、この頂点に関連付けられた他の属性、たとえば、色座標、法線ベクトル、およびテクスチャ座標、とを含むことができる。プリミティブ定義はまた、プリミティブタイプ情報 (たとえば、三角形、四角形、トライアングルファン、トライアングルストリップ、等)、スケーリング情報、回転情報、等を含むことができる。ソフトウェアアプリケーション 18 によって GPU ドライバ 22 に発せられる命令に基づいて、GPU ドライバ 22 は、プリミティブをレンダリングするために GPU 12 が実行する 1 つ以上の動作を指定する 1 つ以上のコマンドを公式化することができる。GPU 12 が CPU 6 からコマンドを受けると、プロセッサ クラスタ 46 は、このコマンドを復号するためにグラフィックス処理パイプラインを実行し得、このコマンドにおいて指定されてい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を構成することができる。たとえば、グラフィックス処理パイプラインのコマンドエンジンは、プリミティブデータを読み取り、このデータを、グラフィックス処理パイプラインにおける他のグラフィックスパイプライン段が使用するためのプリミティブへと組み立てる。指定の動作を実行した後、GPU 12 は、レンダリングされたデータを、ディスプレイデバイスに関連付けられたフレームバッファ 16 にまたはレンダターゲット 24 のうちの 1 つに出力する。

10

【0030】

[0033] フレームバッファ 16 は、GPU 12 のための宛先ピクセルを記憶する。各宛先ピクセルは、一意的なスクリーンピクセルロケーションに関連付けられることができる。いくつかの例では、フレームバッファ 16 は、宛先ピクセルごとに色成分および宛先アルファ値を記憶することができる。たとえば、フレームバッファ 16 は、ピクセルごとに赤、緑、青、アルファ (RGBA) 成分を記憶し得、ここで、「RGB」成分は、カラー値に対応し、「A」成分は、宛先アルファ値に対応する。フレームバッファ 16 はまた、宛先ピクセルごとに深度値を記憶することができる。このように、フレームバッファ 16 は、グラフィックスデータ (たとえば、サーフェス) を記憶すると考えられ得る。フレームバッファ 16 およびシステムメモリ 10 は別個のメモリユニットであると例示されているが、他の例では、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。GPU 12 がフレームのピクセルのすべてをフレームバッファ 16 にレンダリングし終わると、フレームバッファは、完成したフレームを、表示のためにディスプレイ 8 に出力することができる。

20

30

【0031】

[0034] フレームバッファ 16 と同様に、レンダターゲット 24 の各々もまた、ピクセルについてのカラー値および / または深度値を含む、GPU 12 のための宛先ピクセルを記憶することができる。レンダターゲット 24 の各々は、フレームバッファ 16 と同数の一意的なピクセルロケーションについての情報を記憶し得るか、フレームバッファ 16 のようにその数の一意的なピクセルロケーションのサブセットを記憶することができる。

【0032】

[0035] プロセッサ クラスタ 46 は、1 つ以上のプログラマブル処理ユニット 42 および / または 1 つ以上の固定機能処理ユニット 44 を含むことができる。プログラマブル処理ユニット 42 は、たとえば、CPU 6 から GPU 12 上にダウンロードされる 1 つ以上のシェーダプログラムを実行するように構成されたプログラマブルシェーダユニットを含むことができる。いくつかの例では、プログラマブル処理ユニット 42 は、「シェーダプロセッサ」または「統合シェーダ」と呼ばれ得、ジオメトリ、頂点、ピクセル、または他のシェーディング動作を実行して、グラフィックスをレンダリングすることができる。シェーダユニットは、各々、フェッチおよび復号動作のための 1 つ以上の構成要素と、算術計算を実行するための 1 つ以上の ALU と、1 つ以上のメモリと、キャッシュと、レジスタとを含むことができる。

40

【0033】

[0036] GPU 12 は、グラフィックス処理パイプライン中の頂点シェーダ段、テッセレ

50

ーション段、ジオメトリシェーダ段、ラスタライズ段、およびフラグメントシェーダ段のうちの1つ以上を実行するよう求めるコマンドをプログラマブル処理ユニット42に送ること、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、フラグメントシェーディング、等のさまざまなシェーディング動作を実行するようプログラマブル処理ユニット42に指示することができる。いくつかの例では、GPUドライバ22は、1つ以上のシェーダプログラムをコンパイルすることと、コンパイルされたシェーダプログラムをGPU12内に含まれるプログラマブル処理ユニット42上にダウンロードすることとを、CPU6上で実行されるコンパイラに行わせることができる。シェーダプログラムは、たとえば、GLSL (OpenGL Shading Language)、HLSL (High Level Shading Language)、Cg (C for Graphics) シェーディング言語、OpenCL カーネル、等の高度なシェーディング言語で書かれることができる。コンパイルされたシェーダプログラムは、GPU12内のプログラマブル処理ユニット42の動作を制御する1つ以上の命令を含むことができる。たとえば、シェーダプログラムは、頂点シェーダ段の機能を実行するためにプログラマブル処理ユニット42によって実行され得る頂点シェーダプログラム、テッセレーション段の機能を実行するためにプログラマブル処理ユニット42によって実行され得るテッセレーションシェーダプログラム、ジオメトリシェーダ段の機能を実行するためにプログラマブル処理ユニット42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、フラグメントシェーダ段の機能を実行するためにプログラマブル処理ユニット42によって実行され得るフラグメントシェーダプログラムを含むことができる。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統合シェーダユニットの実行を制御し、1つ以上の頂点単位の (per-vertex) 動作を指定する命令を含むことができる。

【0034】

[0037] プロセッサクラスタ46はまた、固定機能処理ユニット44を含むことができる。固定機能処理ユニット44は、特定の機能を実行するためにハードウェア接続されているハードウェアを含むことができる。固定機能処理ユニット44は、たとえば、異なる機能を実行するように、1つ以上の制御信号を介して、構成可能であり得るが、固定機能ハードウェアは典型的に、ユーザコンパイルプログラムを受信することができるプログラムメモリを含まない。いくつかの例では、プロセッサクラスタ46内の固定機能処理ユニット44は、グラフィックス処理パイプラインのラスタライズ段の機能を実行するために、たとえば、深度テスト、シザーテスト、アルファブレンディング、低解像度深度テスト、等の、たとえば、ラスタ動作を実行する処理ユニットを含むことができる。

【0035】

[0038] グラフィックスメモリ40は、オンチップ記憶装置、またはGPU12の集積回路に物理的に組み込まれたメモリである。いくつかの事例では、グラフィックスメモリ40がオンチップであるため、GPU12は、システムバスを介してシステムメモリ10から値を読み取ることまたはシステムメモリ10に値を書き込むことより速くグラフィックスメモリ40から値を読み取ることまたはグラフィックスメモリ40に値を書き込むことができ得る。

【0036】

[0039] いくつかの例では、GPU12は、グラフィックスデータをレンダリングするために、遅延レンダリングモード (ピニングレンダリングまたはタイルベースレンダリングとも呼ばれる) にしたがって動作することができる。遅延レンダリングモードにしたがって動作するとき、GPU12内のプロセッサクラスタ46は、フレームを複数のタイルに分割するために、および、どのプリミティブが各タイル内にあるかを決定するために、最初にピニングパス (binning pass) (タイリングパス (tiling pass) としても知られている) を実行する。いくつかの例では、ピニングパスは、プリミティブがタイル内にあるか否かを示すことができる。他の例では、ピニングパスはまた、深度テストを含み、特定のプリミティブが、レンダリングされたタイルにおいて可視であるか否かを示すことができる。複数のタイルの各々について、次いで、プロセッサクラスタ46は、タイルのグラフィ

ックスデータ（ピクセルのカラー値）を、GPU 12に局所的に位置しているグラフィックスメモリ 40にレンダリングし、これは、各タイルをレンダリングするためにグラフィックス処理パイプラインを実行することを含む、完了すると、グラフィックスメモリ 40からフレームバッファ 16またはレンダターゲット 24のうちの1つに、レンダリングされたグラフィックスデータを読み取る。いくつかの例では、レンダリングされた各タイルが、ピクセルの2次元ブロックのピクセルのカラー値を含むため、タイルは、サーフェスと考えられ得るか、複数のタイルで構成されている最終的にレンダリングされた画像であるサーフェスの一部と考えられ得る。

【0037】

[0040] GPU 12は、各タイルをピクセルの複数のブロックに分割することができる。ピクセルのブロックのサイズは、低解像度バッファにおける1つの記憶場所に対応する、ディスプレイ 8上のピクセルのブロックのサイズに類似し得る。GPU 12は、各タイルのプリミティブをスクリーン空間に変換し得、互いに対してこれらプリミティブを前方から後方に順序付け得、以下を決定するために、現在のタイルのサブタイルをテストする：（1）各プリミティブが所与のサブタイル内に含まれているか、および（2）所与のサブタイルに含まれている場合、プリミティブのピクセルが、特定のサブタイル内の任意の他のプリミティブのピクセルによって塞がれているか。

【0038】

[0041]いくつかの例では、ビニングパス中、GPU 12はまた、複数のタイルの各々のピクセルのブロックについての低解像度z（LRZ）データを生成し得、そのようなLRZデータを、システムメモリ 10のような、メモリ内の低解像度バッファに記憶することができる。低解像度zとは、低解像度バッファが、複数のタイルの各々のピクセルごとというよりはむしろピクセルのブロックに関連付けられた深度データを記憶することを指す。低解像度バッファは、複数の記憶場所を有する2次元バッファであり得る。低解像度バッファ内の各記憶場所は、ディスプレイ 8上に表されるピクセルのブロックに対応することができる。いくつかの例では、低解像度バッファ内の記憶場所の数は、ディスプレイ 8上に表されることとなるピクセル数より少ないであろう。LRZデータは、ピクセルの所与のブロックについての最後部の深度値を含むピクセルのブロック（たとえば、ピクセルの2×2ブロック）についての深度データであり得る。タイルは、1つ以上のLRZデータに関連付けられ得る。たとえば、ピクセルの8×8ブロックであるタイルを考えると、このタイルは、タイルの所与の2×2ピクセルブロックに各々関連付けられている16個のLRZデータを含み得、16個のLRZデータの各々は、タイルの関連する2×2ピクセルブロックについての最後部の深度値を含むことができる。

【0039】

[0042] GPU 12は、LRZデータに関連付けられたピクセルのブロックを占有するプリミティブのピクセルの深度値を決定することに基づいてLRZデータを決定することができる。LRZデータが個々のピクセルではなく、ピクセルのブロックについての深度データであるため、GPU 12は、ピクセルのブロックごとのLRZデータの決定に保守的であり得る。たとえば、LRZデータがピクセルの2×2ブロック（p00、p01、p10、およびp11）である場合、GPU 12は、対応するLRZデータを、最後部のピクセル（すなわち、カメラから最も離れているピクセル）の深度データとなるように設定することができる。ピクセルp00、p01、p10、およびp11がそれぞれ、対応する深度値0.1、0.1、0.2、および0.15を有する場合、ここで、より低い値はより高い値よりカメラから離れている深度を表す、GPU 12は、そのピクセルブロックについてのLRZデータを0.1になるように設定することができる。

【0040】

[0043]レンダリングされたサーフェスを構成するピクセルの深度情報を用いて低解像度バッファを更新した後、GPU 12は、タイルごとに、低解像度バッファに記憶された深度値に基づいて、グラフィックスメモリ 40に画像をレンダリングすることができる。ピクセルをレンダリングするために、ディスプレイ上のピクセルごとに、GPU 12は、低

10

20

30

40

50

解像度バッファ内に記憶された深度値に基づいて、タイル内のどのプリミティブからどのピクセルをレンダリングするかを決定することができる。低解像度バッファ内に記憶された深度値に基づいて、プリミティブのピクセルが最終のシーンにおいて塞がれている(occluded)とGPU12が決定すると、GPU12は、それらの塞がれたピクセルに対するさらなるピクセルシェーディングまたはフラグメントシェーディング動作を実行しないと決定し得、それによって、GPU12の性能を高める。各タイルがグラフィックスメモリ40にレンダリングされた後、GPU12は、レンダリングされたタイルをグラフィックスメモリ40からメモリ26に転送することができる。このように、フレームバッファ16またはレンダターゲット24のうちの1つは、GPU12からのレンダリングされたタイルによってタイルごとに満たされ得、レンダリングされたタイルの各々をグラフィックスメモリからフレームバッファ16またはレンダターゲット24のうちの1つに転送しており、それによって、サーフェスをフレームバッファ16またはレンダターゲット24のうちの1つにレンダリングする。

【0041】

[0044] GPU12がレンダリングされたサーフェスに追加のプリミティブをレンダリングしようと試みるときに、GPU12は、それらのプリミティブのレンダリングを最適化するために、構築されたLRZデータをサーフェスに対して利用し得る。GPU12は、本開示の技法によってそれらのプリミティブをピクセルにラスタライズし得、塞がれているとGPU12が決定したピクセルを破棄するために低解像度深度テストを実行することができる。GPU12は、ピクセルごとに、ピクセルの深度値を、関連するLRZデータ(すなわち、テストされているピクセルのピクセルロケーションに関連付けられたLRZデータ)の深度値と比較し得、このピクセルの深度値が関連するLRZデータの深度値より小さい(たとえば、カメラからより離れている)場合、ピクセルを破棄することができる。これらの塞がれたピクセルを破棄することで、GPU12は、ピクセルシェーディング動作および同様のもののような、それらのピクセルに対する任意の追加のグラフィックスレンダリング動作の実行を省略し得、それによって、GPU12のグラフィックス処理性能を高める。

【0042】

[0045]いくつかの状況では、GPU12は、他のピクセルによって必然的に塞がれているピクセルを、それらのピクセルが個々のピクセルのピクセルレベルの深度テスト中に拒否され得るとしてもGPU12がLRZデータを使用してそれらのピクセルの低解像度テストを実行するとき、拒否しないであろう。たとえば、ピクセルの2x2ブロック(p00、p01、p10、およびp11)を表すLRZデータを考えると、LRZデータは、深度値0.1であり得、ここで、より低い値は、ピクセルp01が実際の深度値0.2を有し得るとしても、より高い値よりカメラから離れている深度を表す。その後、GPU12は、ピクセルp01と同じピクセルロケーションに、深度値0.15を有する新たなピクセルp01'を有するプリミティブをレンダリングするかを決定することができる。LRZデータが深度値0.1であるため、GPU12は、ピクセルp01の実際の深度値が0.2であるとしても、ピクセルp01'がLRZデータの深度値0.1より大きい深度値0.15を有するため、それでも、LRZデータに基づいて、新たなピクセルp01'に関連付けられたプリミティブが最終的にレンダリングされたサーフェスにおいて可視となると決定することができる。LRZデータに基づいた、ピクセルp01'が可視であるとのGPU12の決定により、GPU12が、ピクセルp01'が実際には最終的にレンダリングされたシーンにおいて可視ではないと決定するためにピクセルp01'に対してピクセルレベルの深度テストを実行し、ピクセルp01'を破棄する前に、GPU12は、ピクセルに対してグラフィックスレンダリング動作(たとえば、フラグメントシェーディング動作)を実行し得、それによって、ピクセルp01'のカラー値が、フレームバッファ16またはレンダターゲット24のうちの1つに書き込まれることを防ぐ。

【0043】

[0046] GPU12が、LRZデータを使用した低解像度深度テストの後に各ピクセルの

ピクセルレベルの深度テストを実行するため、L R Z データの使用は、オプションであると考えられ得る。低解像度深度テストは、G P U 1 2 がピクセルに対してピクセルシェーディング動作を実行する前に、それらのピクセルを破棄し得るが、G P U 1 2 は、依然として、G P U 1 2 が破棄されていないピクセルに対してピクセルシェーディング動作を実行した後に、それらの各ピクセルのピクセル単位の深度テストを最終的に実行することができる。ゆえに、L R Z データを使用した低解像度深度テストは、G P U 1 2 が、低解像度深度テストの結果として破棄される特定のピクセルに対してピクセルシェーディングを実行するのにその処理を費やすことを防ぐ G P U 1 2 の処理への最適化であると考えられ得る。そのため、G P U 1 2 は、G P U 1 2 がそのグラフィックス処理の一環として低解像度深度テストを実行しないとしても、グラフィックスデータをレンダリングするために依然として正確に実行することができる。

10

【 0 0 4 4 】

[0047] G P U 1 2 はまた、タイルベースハッシュコードが、それらのピクセルのブロックについての異なる色データを有するタイルを一意的に識別するように、各レンダリングされたタイルに含まれているピクセルのブロックの色データに基づいて、レンダリングされたタイルごとにタイルベースハッシュコードを決定することができる。上述したように、各レンダリングされたタイルは、ピクセルのブロック（たとえば、 8×8 ）であり、ここで、各ピクセルは、カラー値を有する。G P U 1 2 は、ピクセル値の異なるパターンを含むタイル（たとえば、赤いピクセルで完全に満たされているタイルおよび緑のピクセルで完全に満たされているタイル）を異なるタイルベースハッシュコードと関連付け得、ピクセル値の同じパターンを含むタイル（たとえば、赤いピクセルで各々完全に満たされている 2 つのタイル）を同じタイルベースハッシュコードと関連付けることができる。

20

【 0 0 4 5 】

[0048] G P U 1 2 が、第 1 のタイルから第 2 のタイルへの、タイルに対応する色データのビットブロック転送を実行するかを決定するとき、そのようなタイルベースハッシュコードは有益であり得る。第 1 のタイルおよび第 2 のタイルが各々同じタイルベースハッシュコードに関連付けられている場合、G P U 1 2 は、第 1 および第 2 のタイルがピクセルのそれらのそれぞれのブロックについて色データの同じセットを含むため、色データの実際の転送が発生する必要がなにことを決定し得、それによって、コンピューティングデバイス 2 の性能を高める。いくつかの例では、G P U 1 2 は、タイルのサイズより小さいピクセルのブロックについてのタイルベースハッシュコードを決定することができる。たとえば、タイルが 8×8 ブロックを備える場合、G P U 1 2 は、それでも、サーフェスのピクセルの 4×4 ブロックごとにタイルベースハッシュコードを決定することができる。このケースでは、各タイルは、それが含むピクセルの 4×4 ブロックごとに 4 つのタイルベースハッシュコードに関連付けられることができる。

30

【 0 0 4 6 】

[0049] 各レンダリングされたタイルが、フレームバッファ 1 6 またはレンダターゲット 2 4 のうちの 1 つへの記憶のためにグラフィックスメモリ 4 0 から転送されるため、G P U 1 2 は、バスを通じてタイルをフレームバッファ 1 6 またはレンダターゲット 2 4 のうちの 1 つにより効率的に動かすために、任意の適切な圧縮アルゴリズムを介して、各タイルを圧縮することができる。圧縮タイルの、結果として得られるサイズは、各タイルのコンテンツの変動性に基づいて異なり得る。いくつかの圧縮タイルは未圧縮タイルのサイズの数分の 1（fraction）であり得るが、他の圧縮タイルは、未圧縮タイルのサイズよりわずかに小さいまたは同じサイズであり得る、全く圧縮されていないであろう。ゆえに、複数の帯域幅圧縮タイルは、他の圧縮タイルの中の 1 つ以上の未圧縮タイルを含むことができる。

40

【 0 0 4 7 】

[0050] いくつかの例では、G P U 1 2 は、圧縮タイルごとにタイルベースハッシュコードを決定することができる。ゆえに、未圧縮タイルの基礎となる（underlying）サーフェスカラー値についてのタイルベースハッシュコードを生成するのではなく、G P U 1 2 は

50

、圧縮後の各タイルのデータに基づいてタイルベースハッシュコードを生成し得、それによって、複数の圧縮タイルに対するチェックサムとして機能する。この例では、2つのタイルベースハッシュコードは、圧縮後の2つの関連する圧縮タイルが同じである場合、同じであり得る。

【0048】

[0051]所与のレンダリングされた画像の未圧縮タイルがすべて同じサイズであるため、フレームバッファ16またはレンダターゲット24のうちの1つは、各々未圧縮タイルと同じサイズである固定サイズのブロックに、サーフェスの未圧縮タイルのすべてを記憶するのに十分な空間を有するように構成される。さらに、サーフェスを構成するタイルを圧縮することは、各特定のタイルのカラー値により変化する異なるサイズのタイルに帰着し得るため、GPU12は、特に、圧縮タイルを記憶することに対して、メモリ26中の可変サイズのカスタムブロックを割当てることができないであろう。したがって、GPU12は、各圧縮タイルがブロックのうちの1つに記憶されるように複数の圧縮タイルを複数のブロックに記憶することで、レンダリングされた画像の未圧縮タイルを記憶するために割当てられたのと同じ複数のブロックを利用することができる。

10

【0049】

[0052]メモリ26が、未圧縮タイルと各々同じサイズであるブロックに圧縮タイルを記憶するため、メモリ26は、実際には、未圧縮タイルの代わりに複数の圧縮タイルを記憶することで空間を節約してはいない。複数の圧縮タイルがメモリ26において未圧縮タイルより少ない空間を占め得るとしても、やはり、メモリ26内の同じ量の空間が、複数のブロックに記憶されるのが圧縮タイルであるか未圧縮タイルであるかにかかわらず、複数のブロックに対して確保される。

20

【0050】

[0053]したがって、GPU12が、圧縮タイルを複数のブロックに記憶するとき、複数のブロックは、複数の圧縮タイルの記憶によって占められない未使用空間を含むことができる。圧縮タイルが記憶される対応ブロックの全空間より少ない空間を占める各圧縮タイルの場合、対応ブロックは未使用空間を有することができる。そのため、本開示の技法によれば、GPU12は、複数の圧縮タイルから構成されるレンダリングされたサーフェスに関連付けられた追加データを記憶するために未使用空間を利用するように構成されることができる。たとえば、GPU12は、複数の圧縮タイルについてのLRZデータおよびタイルベースハッシュコードをメモリ26内の専用バッファに記憶しないで、そのようなデータを複数のブロックの未使用空間に記憶することができる。

30

【0051】

[0054]圧縮タイルを記憶するブロック内の未使用空間が保証されていないため、GPU12は、特定の圧縮タイルを記憶するブロックが未使用空間を有する場合にのみ、LRZデータおよびタイルベースハッシュコードのような、特定の圧縮タイルに関連付けられたデータを記憶することができ得る。しかしながら、圧縮タイルがブロックを完全に占有する場合、GPU12は、特定の圧縮タイルに関連付けられたデータをこのブロックに記憶することができないであろう。ゆえに、GPU12は、対応する圧縮タイルに関連付けられたサーフェスの各部分についてオプションであるデータを。ブロックの未使用空間に記憶することができ得る。

40

【0052】

[0055]GPU12は、複数のブロックが圧縮タイルを記憶することの結果として生じる、複数のブロックの各々における利用可能な未使用空間を決定することができる。たとえば、GPU12は、複数のブロック中のブロックのサイズを決定し得、圧縮タイルの各々のサイズを決定することができる。特定の圧縮タイルのサイズが複数のブロック中のブロックのサイズより小さいとGPU12が決定すると、GPU12は、この特定の圧縮タイルを記憶するブロックが未使用空間を有し得ることを決定することができる。

【0053】

[0056]複数のブロックのうちの1つ以上が未使用空間を含むとGPU12が決定するこ

50

とに回答して、GPU12は、GPU12がその性能を高めるために利用し得る最適化サーフェスを、複数のブロックのうちの1つ以上のブロックの未使用空間に記憶することができる。たとえば、LRZデータは、GPU12が、最終的にレンダリングされるサーフェスにおいて可視ではないプリミティブのラスター化を実行しないことを可能にすることでそれらのプリミティブを示すことよって、有益である。しかしながら、LRZデータなしで、GPU12は、最終的にレンダリングされるサーフェスにおいてプリミティブが可視であるかどうかにかかわらずそれらのプリミティブのラスター化を実行することで、依然として、所与のサーフェスを正確にレンダリングすることができる。そのため、LRZデータは、それがサーフェスをレンダリングするときGPU12の性能を高め得るが、GPU12がサーフェスを正確にレンダリングするのに不可欠な情報というわけではない。

10

【0054】

[0057]タイルベースハッシュコードは、それらが、GPU12の性能を高めるのに有益ではあるが、GPU12がグラフィックス動作を正確に実行するのに不可欠というわけではない点で、LRZデータに類似している。タイルベースハッシュコードなしで、GPU12は、依然として、色データのビットブロック転送のような機能を正確に実行し得るが、色データの同じブロックを有するサーフェスの部分間での色データの冗長転送を実行することができる。

【0055】

[0058]図3A - 図3Fは、帯域幅圧縮グラフィカルデータをメモリに記憶するための例となる技法を例示する概念図である。図3Aに示されるように、GPU12は、帯域幅圧縮グラフィックスデータ56A - 56N(「帯域幅圧縮グラフィックスデータ56」)を、システムメモリ10、フレームバッファ16、レンダターゲット24のうちの1つ以上、および同様のもののような、メモリ26中のブロック54A - 54N(「ブロック54」)に記憶することができる。帯域幅圧縮グラフィックスデータ56は、いくつかの例では各々、バスを通じておよびコンピューティングデバイス2の構成要素間で(たとえば、GPU12とメモリ26との間で)グラフィックスデータをより効率的に動かすためにGPU12によって圧縮されるレンダリングされたシーンまたはサーフェスを構成するタイル(たとえば、画像サーフェスの一部)であり得る。

20

【0056】

[0059]ブロック54は、メモリ26において連続しており、各々、帯域幅圧縮グラフィックスデータ56の各々を記憶するために同じ均一の固定サイズであり得る。いくつかの例では、帯域幅圧縮グラフィックスデータ56の各々が帯域幅圧縮タイルである場合、GPU12は、メモリ26において、ブロック54の1つ1つが帯域幅圧縮グラフィックスデータ56のうちの対応する1つを記憶し得るように、レンダリングされたサーフェスを構成するタイルの数と同じ数のブロック54を割当てることができる。

30

【0057】

[0060]ブロック54の各々が、レンダリングされたサーフェスの未圧縮グラフィックスデータを記憶するのに十分な大きさであるため、帯域幅圧縮グラフィックスデータ56をブロック54に記憶することは、ブロック54中に未使用空間が残存することに帰着し得る。図3Aの例では、未使用空間58A、58B、58C、58D、および58E(「未使用空間58」)は、ブロック54A、54B、54C、54E、および54Nがそれぞれの帯域幅圧縮グラフィックスデータ56A、56B、56C、56E、および56Nを記憶する場合、それぞれブロック54A、54B、54C、54E、および54Nに残存し得る。

40

【0058】

[0061]上述したように、GPU12は、各帯域幅圧縮グラフィックスデータ56のサイズをブロック54のブロックのサイズと比較することで、ブロック54の各ブロックが未使用空間58を有するかどうかを決定することができる。GPU12は、フラグサーフェス52A - 52N(「フラグサーフェス52」)を作成し、メモリ26に記憶し得、ここ

50

で、フラグサーフェス 5 2 の各々は、ブロック 5 4 のうちの 1 つに関連付けられている、ブロック 5 4 の対応ブロック中の未使用空間の量を示すことができる。

【 0 0 5 9 】

[0062] 図 3 A の例では、フラグサーフェス 5 2 は、ブロック 5 4 の対応ブロック中の未使用空間の量の割合を、4 分のいくつかであるかで、記憶することができる。フラグサーフェス 5 2 A は、未使用空間がブロック 5 4 A の $3/4$ を占めることを示すことができる。フラグサーフェス 5 2 B は、未使用空間がブロック 5 4 B の $1/2$ を占めることを示すことができる。フラグサーフェス 5 2 C は、未使用空間がブロック 5 4 C の $1/4$ を占めることを示すことができる。フラグサーフェス 5 2 D は、ブロック 5 4 D が未使用空間を有さないことを示すことができる。フラグサーフェス 5 2 E は、未使用空間がブロック 5 4 E の $1/4$ を占めることを示すことができる。フラグサーフェス 5 2 F は、ブロック 5 4 F が未使用空間を有さないことを示すことができる。フラグサーフェス 5 2 N は、未使用空間がブロック 5 4 N の $1/4$ を占めることを示すことができる。フラグサーフェス 5 2 もメモリ 2 5 に記憶されるため、メモリに帯域幅圧縮グラフィックスデータ 5 6 を記憶することは、メモリ 2 6 において、同等の未圧縮グラフィックスデータ 5 6 を記憶するより多くの空間を占めることができる。

10

【 0 0 6 0 】

[0063] 上述したように、GPU 1 2 は、帯域幅圧縮グラフィックスデータ 5 6 に関連付けられたデータを未使用空間 5 8 に記憶することができる。図 3 B に示されるように、GPU 1 2 は、未使用空間 5 8 を有するブロック 5 4 のブロックを、フラグサーフェス 5 2 に基づいて、決定し得、LRZ データ 6 0 A - 6 0 E をブロック 5 4 の未使用空間 5 8 A - 5 8 E に記憶することができる。LRZ データ 6 0 A - 6 0 E の各々は、固定サイズであり得る。ブロック 5 4 A、5 4 B、5 4 C、5 4 E、および 5 4 N だけがそれぞれの未使用空間 5 8 A - 5 8 E を有するため、GPU 1 2 は、図 3 B の例では、それぞれの帯域幅圧縮グラフィックスデータ 5 6 A、5 6 B、5 6 C、5 6 E、および 5 6 N についての深度情報を含む LRZ データ 6 0 A - 6 0 E だけをブロック 5 4 の未使用空間 5 8 に記憶することができる。ゆえに、帯域幅圧縮グラフィックスデータ 5 6 D および 5 6 F についての深度情報は、ブロック 5 4 の未使用空間 5 8 に記憶されない。

20

【 0 0 6 1 】

[0064] LRZ データ 6 0 A は、LRZ データ 6 0 A が、帯域幅圧縮グラフィックスデータ 5 6 A に対応するサーフェスの一部を構成するピクセルについての LRZ データを含み得る点で、帯域幅圧縮グラフィックスデータ 5 6 A に関連付けられ得る。例えば、帯域幅圧縮グラフィックスデータ 5 6 A が、ピクセルの特定の 8×8 ブロックに関するグラフィックスデータを含む場合、LRZ データ 6 0 A は、一例では、ピクセルの 8×8 ブロックの各 2×2 ピクセルブロックについての対応する複数の LRZ データを含むことができる。同様に、LRZ データ 6 0 B は、帯域幅圧縮グラフィックスデータ 5 6 B に対応するサーフェスの一部を構成するピクセルについての LRZ データを含み得、LRZ データ 6 0 C は、帯域幅圧縮グラフィックスデータ 5 6 C に対応するサーフェスの一部を構成するピクセルについての LRZ データを含み得、LRZ データ 6 0 D は、帯域幅圧縮グラフィックスデータ 5 6 E に対応するサーフェスの一部を構成するピクセルについての LRZ データを含み得、LRZ データ 6 0 E は、帯域幅圧縮グラフィックスデータ 5 6 N に対応するサーフェスの一部を構成するピクセルについての LRZ データを含むことができる。

30

40

【 0 0 6 2 】

[0065] 関連する LRZ データが関連するブロック 5 4 D および 5 4 に記憶されていない帯域幅圧縮グラフィックスデータ 5 6 D および 5 6 F について、GPU 1 2 は、デフォルトの深度値を、それぞれのブロック 5 4 D および 5 4 F を完全に占有する帯域幅圧縮グラフィックスデータ 5 6 D および 5 6 F の各々に関連付ける。デフォルトの深度値は、帯域幅圧縮グラフィックスデータ 5 6 D および 5 6 F に関連付けられたサーフェスの一部へとレンダリングされるべき追加のピクセルが、帯域幅圧縮グラフィックスデータ 5 6 D および 5 6 F に関連付けられたサーフェスの一部のピクセルの前にあると示す最後部の深度値

50

であり得、ゆえに、それらの追加のピクセルが最終的にレンダリングされるシーンにおいて実際に可視であるかどうかにかかわらず、可視であり得る。

【 0 0 6 3 】

[0066]帯域幅圧縮グラフィックスデータ 5 6 の各々についての深度情報を収容するために、G P U 1 2 は、複数の帯域幅圧縮グラフィックスデータ 5 6 についての深度情報をブロック 5 4 の単一ブロックの未使用空間 5 8 に記憶することができる。図 3 C に示されるように、G P U 1 2 は、複数の連続した帯域幅圧縮グラフィックスデータ 5 6 についての L R Z データを含むブロック 5 4 A の未使用空間 5 8 A に L R Z データ 6 0 F を記憶することができる。ブロック 5 4 の単一ブロックの未使用空間 5 8 に記憶される L R Z データは、帯域幅圧縮グラフィックスデータ 5 6 のうちの関連する帯域幅圧縮グラフィックスデータについての深度情報と、次の連続した指定数の帯域幅圧縮グラフィックスデータ 5 6 についての深度情報とを含むことができる。例えば、ブロック 5 4 の単一ブロックの未使用空間 5 8 に記憶される L R Z データが、帯域幅圧縮グラフィックスデータ 5 6 のうちの 6 つについての L R Z データを含み得る場合、L R Z データ 6 0 F は、帯域幅圧縮グラフィックスデータ 5 6 A - 5 6 F についての深度データを含むことができる。同様に、L R Z データ 6 0 G は、帯域幅圧縮グラフィックスデータ 5 6 N および次の 5 つの後続の帯域幅圧縮グラフィックスデータ 5 6 についての深度データを含むことができる。このように、ブロック 5 4 は、各帯域幅圧縮グラフィックスデータ 5 6 についての深度データを記憶することができる。

【 0 0 6 4 】

[0067]図 3 D に示されるように、G P U 1 2 はまた、タイルベースハッシュコード 6 2 A - 6 2 E をブロック 5 4 の未使用空間 5 8 A - 5 8 E に記憶することができる。タイルベースハッシュコード 6 2 A - 6 2 E の各々は、同じサイズであり得る。ブロック 5 4 A 、 5 4 B 、 5 4 C 、 5 4 E 、 および 5 4 N だけがそれぞれの未使用空間 5 8 A - 5 8 E を有するため、G P U 1 2 は、図 3 B の例では、それぞれの帯域幅圧縮グラフィックスデータ 5 6 A 、 5 6 B 、 5 6 C 、 5 6 E 、 5 6 N についてのカラー値を識別するタイルベースハッシュコード 6 2 A - 6 2 E だけを、ブロック 5 4 の未使用空間 5 8 に記憶することができる。ゆえに、帯域幅圧縮グラフィックスデータ 5 6 D および 5 6 F についてのタイルベースハッシュコードは、ブロック 5 4 の未使用空間 5 8 に記憶されない。

【 0 0 6 5 】

[0068]帯域幅圧縮グラフィックスデータ 5 6 の各々についてのタイルベースハッシュコードを収容するために、G P U 1 2 は、複数の帯域幅圧縮グラフィックスデータ 5 6 についてのタイルベースハッシュコードをブロック 5 4 の単一ブロックの未使用空間 5 8 に記憶することができる。図 3 E に示されるように、G P U 1 2 は、複数の連続した帯域幅圧縮グラフィックスデータ 5 6 についてのタイルベースハッシュコードを含むブロック 5 4 C の未使用空間 5 8 C にタイルベースハッシュコード 6 2 F を記憶することができる。ブロック 5 4 の単一ブロックの未使用空間 5 8 に記憶されるタイルベースハッシュコードは、帯域幅圧縮グラフィックスデータ 5 6 のうちの関連する帯域幅圧縮グラフィックスデータについてのタイルベースハッシュコードおよび次の連続した指定数の帯域幅圧縮グラフィックスデータ 5 6 または前の連続した数の帯域幅圧縮グラフィックスデータ 5 6 についてのタイルベースハッシュコードを含むことができる。例えば、ブロック 5 4 の単一ブロックの未使用空間 5 8 に記憶されたタイルベースハッシュコードが、帯域幅圧縮グラフィックスデータ 5 6 のうちの 3 つについての L R Z データを含み得る場合、タイルベースハッシュコード 6 2 F は、帯域幅圧縮グラフィックスデータ 5 6 A - 5 6 C の各々についてのタイルベースハッシュコードを含むことができる。同様に、タイルベースハッシュコード 6 2 G は、帯域幅圧縮グラフィックスデータ 5 6 N および前の 2 つの帯域幅圧縮グラフィックスデータ 5 6 についてのタイルベースハッシュコードを含むことができる。このように、ブロック 5 4 は、帯域幅圧縮グラフィックスデータ 5 6 の各々についてのタイルベースハッシュコードを記憶することができる。

【 0 0 6 6 】

[0069]いくつかの例では、GPU 12は、帯域幅圧縮グラフィックスデータ56に関連付けられた複数のタイプのデータをブロック54の未使用空間58に同時に記憶することができる。例えば、ブロック54の未使用空間58は、帯域幅圧縮グラフィックスデータ56の各々についての深度データおよびタイルベースハッシュコードの両方を記憶することができる。図3Fに示されるように、GPU 12は、LRZデータ60Hをブロック54Aの未使用空間58Aに、LRZデータ60Iをブロック54Eの未使用空間58Dに記憶することができる。GPU 12はまた、タイルベースハッシュコード62Hをブロック54の未使用空間58Cに、タイルベースハッシュコード62Iをブロック54Nの未使用空間58Eに記憶することができる。そのため、ブロック54の未使用空間58は、帯域幅圧縮グラフィックスデータ56についてのLRZデータおよびタイルベースハッシュコードの両方を同時に記憶することができる。

10

【0067】

[0071]GPU 12がLRZデータおよびタイルベースハッシュコードをブロック54の未使用空間58に記憶することができることを図3A - 3Fは例示するが、本開示は、必ずしも、LRZデータおよびタイルベースハッシュコードだけをブロック54の未使用空間58に記憶することに限定されるわけではない。むしろ、GPU 12は、帯域幅圧縮グラフィックスデータ56に関する任意の他のデータを、ブロック54の未使用空間58に記憶することができる。

【0068】

[0071]図4は、帯域幅圧縮グラフィカルデータをメモリに記憶するための例となるプロセスを例示するフローチャートである。図4に示されるように、プロセスは、GPU 12によって、複数の帯域幅圧縮グラフィックスデータ56を、メモリ26内のそれぞれの複数のブロック54に記憶することを含み得、ここにおいて、複数のブロック54の各々は、メモリ26内において均一の固定サイズであり、複数の帯域幅圧縮グラフィックスデータ56のうちの1つ以上は、固定サイズより小さいサイズを有する(102)。プロセスは、GPU 12によって、複数の帯域幅圧縮グラフィックスデータ56に関連付けられたデータを、複数の帯域幅圧縮グラフィックスデータ56のそれぞれの1つ以上を含む、複数のブロック54のうちの1つ以上のブロックの未使用空間58に記憶することをさらに含む(104)。

20

【0069】

[0072]いくつかの例では、複数の帯域幅圧縮グラフィックスデータ56に関連付けられたデータは、複数のブロック54のうちの1つ以上に記憶された複数の帯域幅圧縮グラフィックスデータ56のうちの1つ以上についての深度データを備える。いくつかの例では、複数の帯域幅圧縮グラフィックスデータ56のうちの第2の1つ以上は、複数のブロック54のうちの第2の1つ以上を完全に占有し得、プロセスは、GPU 12によって、複数の帯域幅圧縮グラフィックスデータ56のうちの第2の1つ以上の帯域幅圧縮グラフィックスデータの各々についてのデフォルトの深度値を関連付けることをさらに含むことができる。いくつかの例では、複数の帯域幅圧縮グラフィックスデータ56に関連付けられたデータは、複数のブロック54のうちの1つ以上に記憶された複数の帯域幅圧縮グラフィックスデータ56の各々についての深度データを備える。

30

40

【0070】

[0073]いくつかの例では、複数の帯域幅圧縮グラフィックスデータ56に関連付けられたデータは、複数のブロック54のうちの1つ以上に記憶された複数の帯域幅圧縮グラフィックスデータ56のうちの1つ以上の帯域幅圧縮グラフィックスデータの各々を識別する1つ以上のハッシュコードを備える。いくつかの例では、複数の帯域幅圧縮グラフィックスデータ56に関連付けられたデータは、複数のブロック54のうちの1つ以上に記憶された複数の帯域幅圧縮グラフィックスデータ56の各々を識別するハッシュコードを備える。

【0071】

[0074]いくつかの例では、複数の帯域幅圧縮グラフィックスデータ56に関連付けられ

50

たデータは、複数のブロック 5 4 のうちの 1 つ以上に記憶された複数の帯域幅圧縮グラフィックスデータ 5 6 の各々を識別するハッシュコードと、複数のブロック 5 4 のうちの 1 つ以上に記憶された複数の帯域幅圧縮グラフィックスデータ 5 6 の各々についての深度データとを備える。

【 0 0 7 2 】

[0075]いくつかの例では、複数の帯域幅圧縮グラフィックスデータ 5 6 に関連付けられたデータは、複数の帯域幅圧縮グラフィックスデータ 5 6 に関連付けられた最適化サーフェスを備える。いくつかの例では、複数の帯域幅圧縮グラフィックスデータ 5 6 は、画像サーフェスの帯域幅圧縮部分を備え得る。

【 0 0 7 3 】

[0076]いくつかの例では、G P U 1 2 によって、複数の帯域幅圧縮グラフィックスデータ 5 6 に関連付けられたデータを、複数の帯域幅圧縮グラフィックスデータ 5 6 のそれぞれの 1 つ以上を含む、複数のブロック 5 4 のうちの 1 つ以上のブロックの未使用空間に記憶することは、G P U 1 2 によって、複数のブロック 5 4 の 1 つ以上が未使用空間を含むと決定することと、複数のブロック 5 4 のうちの 1 つ以上が未使用空間を含むと決定したことに応答して、G P U 1 2 によって、複数の帯域幅圧縮グラフィックスデータ 5 6 に関連付けられたデータを、複数の帯域幅圧縮グラフィックスデータ 5 6 のそれぞれの 1 つ以上を含む、複数のブロック 5 4 のうちの 1 つ以上のブロックの未使用空間に記憶することとをさらに含むことができる。

【 0 0 7 4 】

[0077] 1 つ以上の例では、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み合わせにより実施されることができる。ソフトウェアにより実施される場合、これら機能は、コンピュータ読取可能な媒体において、1 つ以上の命令またはコードとして、記憶または送信されることができる。コンピュータ読取可能な媒体は、ある箇所から別の箇所へのコンピュータプログラムの移送を容易にする任意の媒体を含む通信媒体またはコンピュータデータ記憶媒体を含むことができる。データ記憶媒体は、本開示で説明された技法を実施するための命令、コード、および/またはデータ構造を取り出すために、1 つ以上のコンピュータまたは 1 つ以上のプロセッサによってアクセス可能な任意の利用可能な媒体であり得る。限定ではなく例として、このようなコンピュータ読取可能な媒体は、R A M、R O M、E E P R O M、C D - R O M 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、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または搬送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任意の他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、コンピュータ読取可能な媒体と称され得る。たとえば、ソフトウェアが、ウェブサイト、サーバ、または他のリモートソースから、同軸ケーブル、光ファイバーケーブル、ツイストペア、デジタル加入者回線 (D S L)、または赤外線、電波、およびマイクロ波のようなワイヤレス技術を使用して送信される場合、同軸ケーブル、光ファイバーケーブル、ツイストペア、D S L、または赤外線、電波、およびマイクロ波のようなワイヤレス技術は、媒体の定義に含まれる。本明細書で使用される場合、ディスク (disk) およびディスク (disc) は、コンパクトディスク (C D)、レーザーディスク (登録商標)、光ディスク、デジタル多用途ディスク (D V D)、フロッピー (登録商標) ディスク、およびブルーレイディスクを含み、ここで、ディスク (disk) は通常、磁氣的にデータを再生し、ディスク (disc) は、レーザーを用いて光学的にデータを再生する。上記の組み合わせもまた、コンピュータ読取可能な媒体の範囲内に含まれるべきである。

【 0 0 7 5 】

[0078]コードは、1 つ以上のデジタルシグナルプロセッサ (D S P)、汎用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論理アレイ (F P G A)、あるいは他の同等の集積回路またはディスクリットな論理回路のような 1 つ以上のプロセッサによって実行され得る。したがって、「プロセッサ」および「処理ユニ

10

20

30

40

50

ット」という用語は、本明細書で使用される場合、前述の構造または本明細書で説明された技法の実施に適した任意の他の構造のうちの任意のものを指し得る。加えて、いくつかの態様では、本明細書で説明された機能性は、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および/またはソフトウェアモジュール内で与えられ得るか、複合コードックに組み込まれ得る。また、本技法は、1つ以上の回路または論理要素により完全に実施され得る。

【0076】

[0079]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(すなわち、チップセット)を含む、幅広い種類のデバイスまたは装置により実施され得る。さまざまな構成要素、モジュール、またはユニットは、本開示では、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面を強調するように説明されているが、必ずしも異なるハードウェアユニットによる実現を必要とするわけではない。むしろ、上述したように、さまざまなユニットは、コードックハードウェアユニットに組み合わせられ得るか、適切なソフトウェアおよび/またはファームウェアとともに、上述した1つ以上のプロセッサを含む相互動作するハードウェアユニットの一群によって与えられ得る。

【0077】

[0080]さまざまな例が説明されている。これらの例および他の例は、以下の特許請求の範囲の範囲内である。

以下に本願発明の当初の特許請求の範囲に記載された発明を付記する。

[C1]

方法であって、

少なくとも1つのプロセッサが、複数の帯域幅圧縮グラフィックスデータを、メモリ内のそれぞれの複数のブロックに記憶することと、ここにおいて、前記メモリ内の前記複数のブロックの各々は、均一の固定サイズであり、前記複数の帯域幅圧縮グラフィックスデータのうちの1つ以上は、前記固定サイズより小さいサイズを有する、

前記少なくとも1つのプロセッサが、前記複数の帯域幅圧縮グラフィックスデータに関連付けられたデータを、前記複数の帯域幅圧縮グラフィックスデータの前記それぞれの1つ以上を含む、前記複数のブロックのうちの1つ以上のブロックの未使用空間に記憶することと

を備える方法。

[C2]

前記複数の帯域幅圧縮グラフィックスデータに関連付けられた前記データは、前記複数のブロックのうちの前記1つ以上に記憶された前記複数の帯域幅圧縮グラフィックスデータのうちの前記1つ以上についての深度データを備える、C1に記載の方法。

[C3]

前記少なくとも1つのプロセッサが、前記複数の帯域幅圧縮グラフィックスデータのうちの第2の1つ以上の帯域幅圧縮グラフィックスデータの各々についてのデフォルトの深度値に関連付けることをさらに備え、前記複数の帯域幅圧縮グラフィックスデータのうちの前記第2の1つ以上は、前記複数のブロックのうちの第2の1つ以上を完全に占有する、C2に記載の方法。

[C4]

前記複数の帯域幅圧縮グラフィックスデータに関連付けられた前記データは、前記複数のブロックのうちの前記1つ以上に記憶された前記複数の帯域幅圧縮グラフィックスデータの各々についての深度データを備える、C1に記載の方法。

[C5]

前記複数の帯域幅圧縮グラフィックスデータに関連付けられた前記データは、前記複数のブロックのうちの前記1つ以上に記憶された前記複数の帯域幅圧縮グラフィックスデータのうちの前記1つ以上の帯域幅圧縮グラフィックスデータの各々を識別する1つ以上のハッシュコードを備える、C1に記載の方法。

[C6]

前記複数の帯域幅圧縮グラフィックスデータに関連付けられた前記データは、前記複数のブロックのうちの前記1つ以上に記憶された前記複数の帯域幅圧縮グラフィックスデータの各々を識別するハッシュコードを備える、C 1に記載の方法。

[C 7]

前記複数の帯域幅圧縮グラフィックスデータに関連付けられた前記データは、前記複数のブロックのうちの前記1つ以上に記憶された前記複数の帯域幅圧縮グラフィックスデータの各々を識別するハッシュコードと、前記複数のブロックのうちの前記1つ以上に記憶された前記複数の帯域幅圧縮グラフィックスデータの各々についての深度データとを備える、C 1に記載の方法。

[C 8]

前記複数の帯域幅圧縮グラフィックスデータに関連付けられた前記データは、前記複数の帯域幅圧縮グラフィックスデータに関連付けられた最適化サーフェスを備える、C 1に記載の方法。

[C 9]

前記複数の帯域幅圧縮グラフィックスデータは、画像サーフェスの帯域幅圧縮部分を備える、C 1に記載の方法。

[C 1 0]

前記少なくとも1つのプロセッサが、前記複数の帯域幅圧縮グラフィックスデータに関連付けられた前記データを、前記複数の帯域幅圧縮グラフィックスデータの前記それぞれの1つ以上を含む、前記複数のブロックのうちの前記1つ以上のブロックの前記未使用空間に記憶することは、

前記少なくとも1つのプロセッサが、前記複数のブロックのうちの前記1つ以上が前記未使用空間を含むと決定することと、

前記複数のブロックのうちの前記1つ以上が前記未使用空間を含むと決定したことに応答して、前記少なくとも1つのプロセッサが、前記複数の帯域幅圧縮グラフィックスデータに関連付けられた前記データを、前記複数の帯域幅圧縮グラフィックスデータの前記それぞれの1つ以上を含む、前記複数のブロックのうちの前記1つ以上のブロックの前記未使用空間に記憶することと

をさらに備える、C 1に記載の方法。

[C 1 1]

前記少なくとも1つのプロセッサは、グラフィックス処理ユニットを含む、C 1に記載の方法。

[C 1 2]

グラフィックスデータを処理するように構成された装置であって、

メモリと、

少なくとも1つのプロセッサと

を備え、前記少なくとも1つのプロセッサは、

複数の帯域幅圧縮グラフィックスデータを、前記メモリ内のそれぞれの複数のブロックに記憶することと、ここにおいて、前記メモリ内の前記複数のブロックの各々は、均一の固定サイズであり、前記複数の帯域幅圧縮グラフィックスデータのうちの1つ以上は、前記固定サイズより小さいサイズを有する、

前記複数の帯域幅圧縮グラフィックスデータに関連付けられたデータを、前記複数の帯域幅圧縮グラフィックスデータの前記それぞれの1つ以上を含む、前記複数のブロックのうちの1つ以上のブロックの未使用空間に記憶することと

を行うように構成される、装置。

[C 1 3]

前記複数の帯域幅圧縮グラフィックスデータに関連付けられた前記データは、前記複数のブロックのうちの前記1つ以上に記憶された前記複数の帯域幅圧縮グラフィックスデータのうちの前記1つ以上についての深度データを備える、C 1 2に記載の装置。

[C 1 4]

前記少なくとも1つのプロセッサは、

前記複数の帯域幅圧縮グラフィックスデータのうちの第2の1つ以上の帯域幅圧縮グラフィックスデータの各々についてのデフォルトの深度値を関連付けるようにさらに構成され、前記複数の帯域幅圧縮グラフィックスデータのうちの第2の1つ以上は、前記複数のブロックのうちの第2の1つ以上を完全に占有する、C 1 3に記載の装置。

[C 1 5]

前記複数の帯域幅圧縮グラフィックスデータに関連付けられた前記データは、前記複数のブロックのうちの前記1つ以上に記憶された前記複数の帯域幅圧縮グラフィックスデータの各々についての深度データを備える、C 1 2に記載の装置。

[C 1 6]

前記複数の帯域幅圧縮グラフィックスデータに関連付けられた前記データは、前記複数のブロックのうちの前記1つ以上に記憶された前記複数の帯域幅圧縮グラフィックスデータのうちの前記1つ以上の帯域幅圧縮グラフィックスデータの各々を識別する1つ以上のハッシュコードを備える、C 1 2に記載の装置。

[C 1 7]

前記複数の帯域幅圧縮グラフィックスデータに関連付けられた前記データは、前記複数のブロックのうちの前記1つ以上に記憶された前記複数の帯域幅圧縮グラフィックスデータの各々を識別するハッシュコードを備える、C 1 2に記載の装置。

[C 1 8]

前記複数の帯域幅圧縮グラフィックスデータに関連付けられた前記データは、前記複数のブロックのうちの前記1つ以上に記憶された前記複数の帯域幅圧縮グラフィックスデータの各々を識別するハッシュコードと、前記複数のブロックのうちの前記1つ以上に記憶された前記複数の帯域幅圧縮グラフィックスデータの各々についての深度データとを備える、C 1 2に記載の装置。

[C 1 9]

前記複数の帯域幅圧縮グラフィックスデータに関連付けられた前記データは、前記複数の帯域幅圧縮グラフィックスデータに関連付けられた最適化サーフェスを備える、C 1 2に記載の装置。

[C 2 0]

前記複数の帯域幅圧縮グラフィックスデータは、画像サーフェスの帯域幅圧縮部分を備える、C 1 2に記載の装置。

[C 2 1]

前記少なくとも1つのプロセッサは、

前記複数のブロックのうちの前記1つ以上が前記未使用空間を含むと決定することと、前記複数のブロックのうちの前記1つ以上が前記未使用空間を含むと決定したことに応答して、前記複数の帯域幅圧縮グラフィックスデータに関連付けられた前記データを、前記複数の帯域幅圧縮グラフィックスデータの前記それぞれの1つ以上を含む、前記複数のブロックのうちの前記1つ以上のブロックの前記未使用空間に記憶することとを行うようにさらに構成される、C 1 2に記載の装置。

[C 2 2]

前記少なくとも1つのプロセッサは、グラフィックス処理ユニットを含む、C 1 2に記載の装置。

[C 2 3]

装置であって、

複数の帯域幅圧縮グラフィックスデータを、メモリ内のそれぞれの複数のブロックに記憶するための手段と、ここにおいて、前記メモリ内の前記複数のブロックの各々は、均一の固定サイズであり、前記複数の帯域幅圧縮グラフィックスデータのうちの1つ以上は、前記固定サイズより小さいサイズを有する、

前記複数の帯域幅圧縮グラフィックスデータに関連付けられたデータを、前記複数の帯域幅圧縮グラフィックスデータの前記それぞれの1つ以上を含む、前記複数のブロックの

10

20

30

40

50

うちの1つ以上のブロックの未使用空間に記憶するための手段と
を備える、装置。

[C 2 4]

前記複数の帯域幅圧縮グラフィックスデータに関連付けられた前記データは、前記複数の
ブロックのうちの前記1つ以上に記憶された前記複数の帯域幅圧縮グラフィックスデー
タのうちの前記1つ以上についての深度データを備える、C 2 3に記載の装置。

[C 2 5]

前記複数の帯域幅圧縮グラフィックスデータのうちの第2の1つ以上の帯域幅圧縮グラ
フィックスデータの各々についてのデフォルトの深度値に関連付けるための手段をさらに
備え、前記複数の帯域幅圧縮グラフィックスデータのうちの第2の1つ以上は、前記複数
のブロックのうちの第2の1つ以上を完全に占有する、C 2 4に記載の装置。

10

[C 2 6]

前記複数の帯域幅圧縮グラフィックスデータに関連付けられた前記データは、前記複数
のブロックのうちの前記1つ以上に記憶された前記複数の帯域幅圧縮グラフィックスデー
タの各々についての深度データを備える、C 2 3に記載の装置。

[C 2 7]

前記複数の帯域幅圧縮グラフィックスデータに関連付けられた前記データは、前記複数
のブロックのうちの前記1つ以上に記憶された前記複数の帯域幅圧縮グラフィックスデー
タのうちの前記1つ以上の帯域幅圧縮グラフィックスデータの各々を識別する1つ以上の
ハッシュコードを備える、C 2 3に記載の装置。

20

[C 2 8]

前記複数の帯域幅圧縮グラフィックスデータに関連付けられた前記データは、前記複数
のブロックのうちの前記1つ以上に記憶された前記複数の帯域幅圧縮グラフィックスデー
タの各々を識別するハッシュコードを備える、C 2 3に記載の装置。

[C 2 9]

前記複数の帯域幅圧縮グラフィックスデータに関連付けられた前記データは、前記複数
のブロックのうちの前記1つ以上に記憶された前記複数の帯域幅圧縮グラフィックスデー
タの各々を識別するハッシュコードと、前記複数のブロックのうちの前記1つ以上に記憶
された前記複数の帯域幅圧縮グラフィックスデータの各々についての深度データとを備え
る、C 2 3に記載の装置。

30

[C 3 0]

前記記憶するための手段は、
前記複数のブロックのうちの前記1つ以上が前記未使用空間を含むと決定するための手
段と、

前記複数のブロックのうちの前記1つ以上が前記未使用空間を含むと決定したことに応
答して、前記複数の帯域幅圧縮グラフィックスデータに関連付けられた前記データを、前
記複数の帯域幅圧縮グラフィックスデータの前記それぞれの1つ以上を含む、前記複数
のブロックのうちの前記1つ以上のブロックの前記未使用空間に記憶するための手段と
をさらに備える、C 2 3に記載の装置。

【図 1】

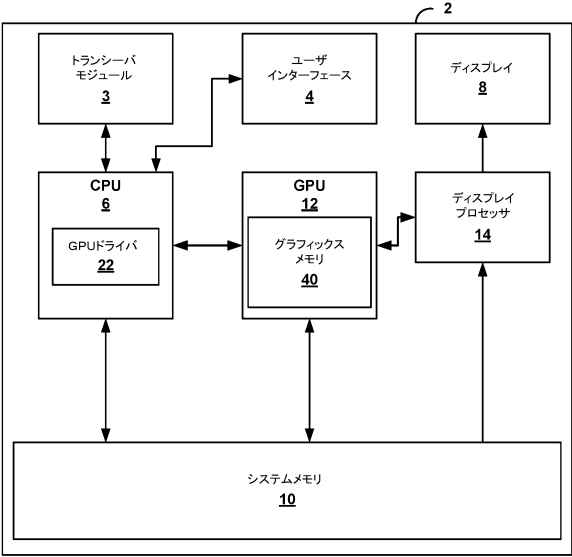


FIG. 1

【図 2】

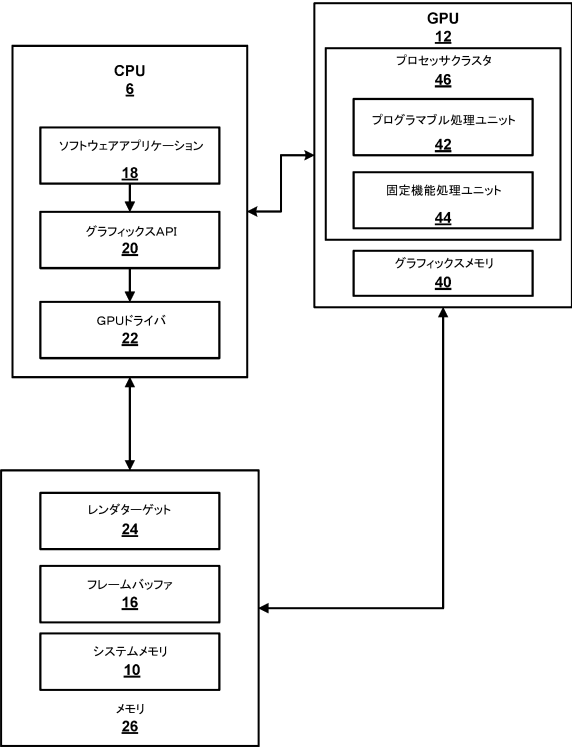


FIG. 2

【図 3 A】

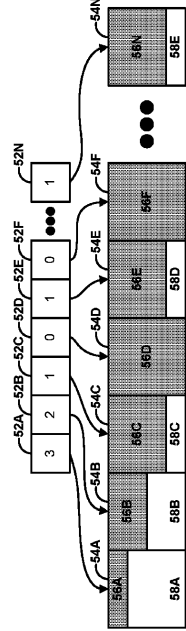


FIG. 3A

【図 3 B】

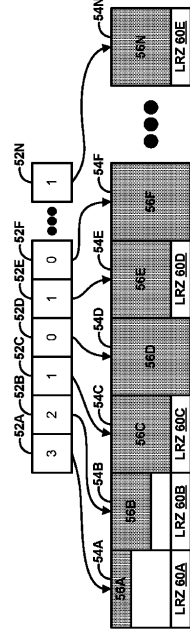


FIG. 3B

【図 4】

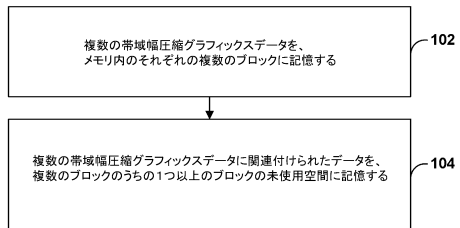


FIG. 4

フロントページの続き

(74)代理人 100184332

弁理士 中丸 慶洋

(72)発明者 グルバー、アンドリュー・イバン

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 ヒル、レクスフォード・アラン

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 カンデルウォール、シャンプー

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

審査官 酒井 恭信

(56)参考文献 特開 2 0 0 0 - 2 3 2 5 8 7 (J P , A)

米国特許第 0 8 3 7 8 8 5 9 (U S , B 1)

米国特許出願公開第 2 0 1 3 / 0 3 2 2 7 5 6 (U S , A 1)

AKENINE-MOLLER, Tomas et al., Graphics Processing Units for Handhelds, Proceedings of the IEEE, 2 0 0 8 年 5 月, Vol.96, No.5, pp.779 - 789

(58)調査した分野(Int.Cl., D B 名)

G 0 6 F 1 2 / 0 0 - 1 2 / 0 4

G 0 6 T 1 / 2 0