

(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2019-509555

(P2019-509555A)

(43) 公表日 平成31年4月4日(2019.4.4)

(51) Int.Cl.

G06T 15/04

(2011.01)

F 1

GO 6 T 15/04

テーマコード（参考）

5B080

(P2019-509555A)

審查請求 未請求 予備審查請求 有 (全 28 頁)

(21) 出願番号	特願2018-543212 (P2018-543212)	(71) 出願人	595020643 クゥアルコム・インコーポレイテッド QUALCOMM INCORPORATED
(86) (22) 出願日	平成29年1月9日 (2017.1.9)		
(85) 翻訳文提出日	平成30年10月15日 (2018.10.15)		
(86) 國際出願番号	PCT/US2017/012734		
(87) 國際公開番号	W02017/142641		アメリカ合衆国、カリフォルニア州 92121-1714、サン・ディエゴ、モアハウス・ドライブ 5775
(87) 國際公開日	平成29年8月24日 (2017.8.24)		
(31) 優先権主張番号	15/047,472	(74) 代理人	100108855 弁理士 蔵田 昌俊
(32) 優先日	平成28年2月18日 (2016.2.18)	(74) 代理人	100109830 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

(54) 【発明の名称】テクスチャユニットにおけるフィードバックを伴うマルチステップテクスチャ処理

(57) 【要約】

シェーダプロセッサの複数の動作を行うためにテクスチャユニットを使用するための技法が説明される。シェーダプロセッサのいくつかの複数の動作は、条件が満たされるまで繰り返し実行され、各実行反復において、シェーダプロセッサはテクスチャユニットにアクセスする。条件が満たされるまでテクスチャユニットがそのような複数の動作を行うための技法が説明される。

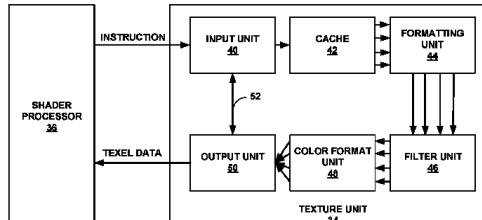


FIG. 3

【特許請求の範囲】**【請求項 1】**

データを処理する方法であって、前記方法は、

命令において定義された条件が満たされることに基づいて複数の動作を繰り返し実行することをテクスチャユニットに命令する前記命令を、前記テクスチャユニットで受信することと、

前記命令において定義された前記条件が満たされることまたは満たされないことにに基づいて、前記複数の動作を前記テクスチャユニットで繰り返し実行することと、

前記複数の動作の前記繰り返される実行から生じるデータを前記テクスチャユニットでおよびグラフィックス処理ユニット（GPU）に出力することと
を備える方法。
10

【請求項 2】

前記命令を受信することは前記GPUのシェーダプロセッサから前記命令を受信することを備え、ここにおいて、出力することは前記データを前記GPUの前記シェーダプロセッサに出力することを備える、請求項1に記載の方法。

【請求項 3】

前記複数の動作の前記繰り返される実行の第1の反復中にテクセル値を前記テクスチャユニットで読み取ることと、

前記テクセル値に基づく値を、前記命令において定義された変数と比較することによって、前記条件が満たされるのか満たされないのかを、前記テクスチャユニットで決定することと、
20

前記命令において定義された前記条件が満たされるのか満たされないのかの前記決定に基づいて、前記複数の動作の実行の第2の反復が必要とされるかどうかを、前記テクスチャユニットで決定することと

をさらに備え、

ここにおいて、前記複数の動作を繰り返し実行することは、前記複数の動作の実行の前記第2の反復が必要とされるという前記決定に基づいて、前記テクスチャユニットの出力をフィードバック信号として前記テクスチャユニットの入力に、前記テクスチャユニットで出力することを備え、

ここにおいて、データを出力することは、前記複数の動作の実行の前記第2の反復が必要とされないという前記決定に基づいて、前記複数の動作の前記繰り返される実行から生じる前記データを出力することを備える、
30

請求項1に記載の方法。

【請求項 4】

前記複数の動作を繰り返し実行することが、前記複数の動作を実行するために何も追加命令を受け取らずに、前記命令において定義された前記条件が満たされることまたは満たされないことにに基づいて、前記複数の動作を繰り返し実行することを備える、請求項1に記載の方法。

【請求項 5】

前記複数の動作はシェーダプログラムの複数の動作を備える、請求項1に記載の方法。
40

【請求項 6】

前記複数の動作はテクスチャ座標を変更するための複数の動作を備える、請求項1に記載の方法。

【請求項 7】

前記複数の動作は、視差遮蔽マッピング（POM）、スクリーン空間光線追跡（SSRT）、被写界深度（DOF）処理、ボリュームレンダリング、あるいは動的高さフィールドを用いた水または地形レンダリングのうちの1つまたは複数のための複数の動作を備える、請求項1に記載の方法。

【請求項 8】

前記複数の動作を繰り返し実行することは、前記条件が満たされるまで、または前記条
50

件が満たされる限り、前記複数の動作を繰り返し実行することを備える、請求項 1 に記載の方法。

【請求項 9】

前記複数の動作の前記繰り返される実行から生じる前記データを出力することは、前記複数の動作の前記繰り返される実行の全ての反復が完了した後にのみ、前記データをシェーダプロセッサに出力することを備える、請求項 1 に記載の方法。

【請求項 10】

データを処理するためのデバイスであって、前記デバイスは、
シェーダプロセッサを備えるグラフィックス処理ユニット（GPU）と、
テクスチャユニットとを備え、前記テクスチャユニットは、

10

命令において定義された条件が満たされることに基づいて複数の動作を繰り返し実行することを前記テクスチャユニットに命令する前記命令を、前記GPUの前記シェーダプロセッサから受信することと、

前記命令において定義された前記条件が満たされることまたは満たされないことにに基づいて、前記複数の動作を繰り返し実行することと、

前記複数の動作の前記繰り返される実行から生じるデータを、前記GPUに出力することと

を行うように構成された、デバイス。

【請求項 11】

前記テクスチャユニットは、前記GPUの前記シェーダプロセッサに、前記複数の動作の前記繰り返される実行から生じる前記データを出力するように構成された、請求項 1 0 に記載のデバイス。

20

【請求項 12】

前記テクスチャユニットは、
入力ユニットと、

前記複数の動作の前記繰り返される実行の第 1 の反復中にテクセル値を読み取るように構成された読み取りユニットと、

前記テクセル値に基づく値を、前記命令において定義された変数と比較することによって、前記条件が満たされるのか満たされないのかを決定することと、

前記命令において定義された前記条件が満たされるのか満たされないのかの前記決定に基づいて、前記複数の動作の実行の第 2 の反復が必要とされるかどうかを決定することと

30

を行うように構成された出力ユニットと
を備え、

ここにおいて、前記複数の動作を繰り返し実行するために、前記出力ユニットは、前記複数の動作の実行の前記第 2 の反復が必要とされるという前記決定に基づいて、前記テクスチャユニットの前記入力ユニットにフィードバック信号を出力するように構成され、

ここにおいて、データを出力するために、前記出力ユニットは、前記複数の動作の実行の前記第 2 の反復が必要とされないという前記決定に基づいて、前記複数の動作の前記繰り返される実行から生じる前記データを出力するように構成された、
請求項 1 0 に記載のデバイス。

40

【請求項 13】

前記テクスチャユニットは、前記複数の動作を実行するために何も追加命令を受け取らずに、前記命令において定義された前記条件が満たされることまたは満たされないことにに基づいて、前記複数の動作を繰り返し実行するように構成された、請求項 1 0 に記載のデバイス。

【請求項 14】

前記複数の動作がシェーダプログラムの複数の動作を備える、請求項 1 0 に記載のデバイス。

【請求項 15】

50

前記複数の動作がテクスチャ座標を変更するための複数の動作を備える、請求項 10 に記載のデバイス。

【請求項 16】

前記複数の動作は、視差遮蔽マッピング (POM)、スクリーン空間光線追跡 (SSRT)、被写界深度 (DOF) 処理、ボリュームレンダリング、あるいは動的高さフィールドを用いた水または地形レンダリングのうちの 1 つまたは複数のための複数の動作を備える、請求項 10 に記載のデバイス。

【請求項 17】

前記テクスチャユニットは、前記条件が満たされるまで、または前記条件が満たされる限り、前記動作を繰り返し実行するように構成された、請求項 10 に記載のデバイス。

10

【請求項 18】

前記テクスチャユニットは、前記複数の動作の前記繰り返される実行の全ての反復が完了した後にのみ、前記複数の動作の前記繰り返される実行から生じる前記データを前記 GPU の前記シェーダプロセッサに出力するように構成された、請求項 10 に記載のデバイス。

【請求項 19】

前記デバイスは、
集積回路、
マイクロプロセッサ、または
ワイヤレス通信デバイス
のうちの 1 つを備える、請求項 10 に記載のデバイス。

20

【請求項 20】

前記 GPU は前記テクスチャユニットを備える、請求項 10 に記載のデバイス。

【請求項 21】

データを処理するためのデバイスであって、前記デバイスは、
命令において定義された条件が満たされることに基づいて複数の動作を繰り返し実行することをテクスチャユニットに命令する前記命令を受信するための手段と、
前記命令において定義された前記条件が満たされることまたは満たされないことにに基づいて、前記複数の動作を繰り返し実行するための手段と、

前記複数の動作の前記繰り返される実行から生じるデータを、グラフィックス処理ユニット (GPU) に出力するための手段と
を備える、デバイス。

30

【請求項 22】

前記複数の動作の前記繰り返される実行の第 1 の反復中にテクセル値を読み取るための手段と、

前記テクセル値に基づく値を、前記命令において定義された変数と比較することによって、前記条件が満たされるのか満たされないのかを決定するための手段と、

前記命令において定義された前記条件が満たされるのか満たされないのかの前記決定に基づいて、前記複数の動作の実行の第 2 の反復が必要とされるかどうかを決定するための手段と

40

をさらに備え、

ここにおいて、前記複数の動作を繰り返し実行するための前記手段は、前記複数の動作の実行の前記第 2 の反復が必要とされるという前記決定に基づいて、前記テクスチャユニットの出力をフィードバック信号として前記テクスチャユニットの入力に出力するための手段を備え、

ここにおいて、データを出力するための前記手段は、前記複数の動作の実行の前記第 2 の反復が必要とされないという前記決定に基づいて、前記複数の動作の前記繰り返される実行から生じる前記データを出力するための手段を備える、
請求項 21 に記載のデバイス。

【請求項 23】

50

前記複数の動作を繰り返し実行するための前記手段は、前記複数の動作を実行するために何も追加命令を受け取らずに、前記命令において定義された前記条件が満たされることまたは満たされないことに基づいて、前記複数の動作を繰り返し実行するための手段を備える、請求項21に記載のデバイス。

【請求項24】

前記複数の動作の前記繰り返される実行から生じる前記データを出力するための前記手段は、前記複数の動作の前記繰り返される実行の全ての反復が完了した後にのみ、前記データをシェーダプロセッサに出力するための手段を備える、請求項21に記載のデバイス。
。

【請求項25】

非一時的コンピュータ可読記憶媒体であって、
命令において定義された条件が満たされることに基づいて複数の動作を繰り返し実行することをテクスチャユニットに命令する前記命令を受信することと、
前記命令において定義された前記条件が満たされることまたは満たされないことに基づいて、前記複数の動作を繰り返し実行することと、
前記複数の動作の前記繰り返される実行から生じるデータを、グラフィックス処理ユニット(GPU)に出力することと、
実行されたとき、データを処理するためのデバイスの1つまたは複数のプロセッサに行わせる、複数の命令を記憶している、非一時的コンピュータ可読記憶媒体。

【請求項26】

前記複数の動作の前記繰り返される実行の第1の反復中にテクセル値を読み取ることと、
前記テクセル値に基づく値を、前記命令において定義された変数と比較することによって、前記条件が満たされるのか満たされないのかを決定することと、
前記命令において定義された前記条件が満たされるのか満たされないのかの前記決定に基づいて、前記複数の動作の実行の第2の反復が必要とされるかどうかを決定することとを前記1つまたは複数のプロセッサに行わせる複数の命令をさらに備え、
ここにおいて、前記1つまたは複数のプロセッサに前記複数の動作を繰り返し実行させる前記複数の命令は、前記1つまたは複数のプロセッサに、前記複数の動作の実行の前記第2の反復が必要とされるという前記決定に基づいて、前記テクスチャユニットの出力をフィードバック信号として前記テクスチャユニットの入力に出力させる複数の命令を備え、
ここにおいて、前記1つまたは複数のプロセッサにデータを出力させる前記複数の命令は、前記1つまたは複数のプロセッサに、前記複数の動作の実行の前記第2の反復が必要とされないという前記決定に基づいて、前記複数の動作の前記繰り返される実行から生じる前記データを出力させる複数の命令を備える、請求項25に記載の非一時的コンピュータ可読記憶媒体。

【請求項27】

前記1つまたは複数のプロセッサに前記複数の動作を繰り返し実行させる前記複数の命令は、前記1つまたは複数のプロセッサに、前記複数の動作 . 24 を実行するために何も追加命令を受け取らずに、前記命令において定義された前記条件が満たされることまたは満たされないことに基づいて、前記複数の動作を繰り返し実行させる命令を備える、請求項25に記載の非一時的コンピュータ可読記憶媒体。

【請求項28】

前記1つまたは複数のプロセッサに前記複数の動作の前記繰り返される実行から生じる前記データを出力させる前記複数の命令は、前記1つまたは複数のプロセッサに、前記複数の動作の前記繰り返される実行の全ての反復が完了した後にのみ、前記データをシェーダプロセッサに出力させる複数の命令を備える、請求項25に記載の非一時的コンピュータ可読記憶媒体。

【発明の詳細な説明】

10

20

30

40

50

【技術分野】**【0001】**

[0001] 本開示は、グラフィックス処理システムに関し、より詳細には、テクスチャユニットを利用するグラフィックス処理システムに関する。

【背景技術】**【0002】**

[0002] コンピューティングデバイスは、しばしば、表示のためのグラフィックスデータのレンダリングを加速するために、グラフィックス処理ユニット（GPU）を利用する。そのようなコンピューティングデバイスは、例えば、コンピュータワークステーション、いわゆるスマートフォンなどのモバイルフォン、組込みシステム、パーソナルコンピュータ、タブレットコンピュータ、およびビデオゲームコンソールを含み得る。GPUは、一般に、グラフィックス処理コマンドを実行するために一緒に動作する複数の処理段を含むグラフィックス処理パイプラインを実行する。ホスト中央処理ユニット（CPU）は、1つまたは複数のグラフィックス処理コマンドをGPUに発行することでGPUの動作を制御し得る。10

【発明の概要】**【0003】**

[0003] 本開示は、テクスチャユニットへの呼出しを制限するようにグラフィックス処理ユニット（GPU）のシェーダプロセッサの複数の動作を実施(implement)するためにテクスチャユニットを使用することを意図する。シェーダプロセッサが行うべきテクスチャユニットへの複数の呼出しを含む複数の動作は、代わりに、テクスチャユニットによって行われる。シェーダプロセッサが行うべきこれら動作の各々は、代わりに、テクスチャユニット内のハードウェア構成要素によって行われ得る。このようにして、GPUは、シェーダプロセッサが行うべき複数の動作を行うためにテクスチャユニットのハードウェアを活用し、テクスチャユニットへの呼出しを制限する。20

【0004】

[0004] 一例において、本開示は、データを処理する例示的な方法であって、命令において定義された条件が満たされることに基づいて複数の動作を繰り返し実行することをテクスチャユニットに命令する1つの命令をテクスチャユニットで受け取ることと、この命令において定義された条件が満たされることまたは満たされないことにに基づいてこれら動作をテクスチャユニットで繰り返し実行することと、これら動作の繰り返される実行から生じるデータを、グラフィックス処理ユニット（GPU）にテクスチャユニットで出力することとを備える方法について説明する。30

【0005】

[0005] 一例において、本開示は、データを処理するための例示的なデバイスであって、シェーダプロセッサを備えるグラフィックス処理ユニット（GPU）と、テクスチャユニットとを備え、このテクスチャユニットは、命令において定義された条件が満たされることに基づいて動作を繰り返し実行することをテクスチャユニットに命令する命令を、GPUのシェーダプロセッサから受け取ることと、命令において定義された条件が満たされることまたは満たされないことにに基づいて動作を繰り返し実行することと、動作の繰り返される実行から生じるデータをGPUに出力することを行なうように構成される、デバイスについて説明する。40

【0006】

[0006] 一例において、本開示は、データを処理するための例示的なデバイスであって、命令において定義された条件が満たされることに基づいて動作を繰り返し実行することをテクスチャユニットに命令する命令を受け取るための手段と、命令において定義された条件が満たされることまたは満たされないことにに基づいて動作を繰り返し実行するための手段と、動作の繰り返される実行から生じるデータをグラフィックス処理ユニット（GPU）に出力するための手段とを備えるデバイスについて説明する。

【0007】

10

20

30

40

50

[0007] 一例において、本開示は、例示的な非一時的コンピュータ可読記憶媒体であって、実行されたとき、データを処理するためのデバイスの1つまたは複数のプロセッサに、命令において定義された条件が満たされることに基づいて動作を繰り返し実行することをテクスチャユニットに命令する命令を受け取ることと、命令において定義された条件が満たされることまたは満たされないことにに基づいて動作を繰り返し実行することと、動作の繰り返される実行から生じるデータをグラフィックス処理ユニット(GPU)に出力することとを行わせる複数の命令を記憶している非一時的コンピュータ可読記憶媒体について説明する。

【0008】

[0008] 本開示の1つまたは複数の例の詳細が添付の図面および以下の説明に記載される。本開示の他の特徴、目的、および利点は、本説明および図面、並びに特許請求の範囲から明らかになるはずである。 10

【図面の簡単な説明】

【0009】

【図1】本開示の技法を実施するために使用され得る例示的なコンピューティングデバイスを示すブロック図。

【図2】図1のコンピューティングデバイスのCPU、GPUおよびメモリをさらに詳細に示すブロック図。

【図3】図2のテクスチャユニットの一例をさらに詳細に示すブロック図。

【図4】本開示で説明される1つまたは複数の例示的な技法による、データを処理する例示的な方法を示すフローチャート。 20

【発明を実施するための形態】

【0010】

[0013] 本開示は複数の動作を行うためにテクスチャユニットを活用することを意図するもので、これら動作はそうでなければグラフィックス処理ユニット(GPU)のシェーダプロセッサがこれら動作を行うために複数の呼出しをテクスチャユニットに発行することを要求することになる。様々なグラフィックス処理アルゴリズムに関し、シェーダプロセッサは、テクスチャユニットがテクスチャデータを取り出し、テクスチャデータを処理し、処理されたテクスチャデータ(例えば、テクセル)をシェーダプロセッサに出力することの複数の要求を出力する。 30

【0011】

[0014] テクスチャユニットにアクセスすることをシェーダプロセッサに繰り返し行わせる様々なグラフィックス処理アルゴリズムにおける1つの共通要因は、シェーダプロセッサ上で実行する複数の命令の構造である。例えば、これら命令の構造は、一般に終了条件をもつループと、テクスチャ座標を進める/変更するためのロジックと、これら命令において定義された動作の結果を計算するためのロジックとを含む。

【0012】

[0015] 本開示で説明される技法において、テクスチャユニットへの繰り返される呼出しを含む複数の命令をシェーダプロセッサに実行させてなく、これら命令はテクスチャユニットのハードウェア構成要素によって行われるようにマッピングされ得る。例えば、テクスチャユニットは、テクスチャユニットの出力が入力を受け取るテクスチャユニットの構成要素にフィードバックするフィードバック経路を含み得る。このフィードバックで、テクスチャユニットは、シェーダプロセッサのような別のユニットからテクスチャユニットへの繰り返される呼出しを必要とせずに、ループの反復を実施するように構成され得る。こうして、シェーダプロセッサは、データのセット(例えば、シェーダプロセッサがそれに対して動作を行おうとしたデータ)をテクスチャユニットに出力することをシェーダプロセッサにさせる1つの命令を実行し得、テクスチャユニットは、次いで、ループにおける複数の命令の反復を内部フィードバック経路を使用して行い、最終データ(テクセル)を伴う結果をシェーダプロセッサに1回出力する。従って、シェーダプロセッサは、中間データを伴って複数回出力するのでなく、テクスチャユニットに1回出力すること 40

10

20

30

40

50

が必要であり、テクスチャユニットに複数回アクセスすることをシェーダプロセッサにさせるのでなく（例えば、中間出力をテクスチャユニットから複数回受け取って、複数の呼出しでテクスチャユニットを起動するのでなく）、データをテクスチャユニットから1回受け取り得る。

【0013】

[0016] いくつかの例において、GPUとテクスチャユニットとは、同じ集積回路内に存在し得るかまたは異なる集積回路内に存在し得る。テクスチャユニットは、命令において定義された条件（例えば、終了条件）の生起に基づいて動作（例えば、ループされた複数の命令）を繰り返し実行することをテクスチャユニットに命令する命令をGPUのシェーダプロセッサから受け取るように構成され得る。それに応じて、テクスチャユニットは、命令において定義された条件が満たされるかまたは満たされなくなるまで、動作を繰り返し実行し、動作の繰り返される実行から生じるデータをGPUに出力し得る。
10

【0014】

[0017] 図1は、本開示の技法を実施するために使用され得る例示的なコンピューティングデバイス2を示すブロック図である。コンピューティングデバイス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、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、ワイヤレス通信デバイス（例えば、携帯電話、セルラー電話、衛星電話、および／または携帯電話ハンドセットなど）、固定電話、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、パーソナル音楽プレーヤ、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、あるいはグラフィカルデータを処理および／または表示する任意の他のタイプのデバイスを備え得る。
20

【0015】

[0018] 図1の例に示されるように、コンピューティングデバイス2は、ユーザ入力インターフェース4と、CPU6と、メモリコントローラ8と、システムメモリ10と、グラフィックス処理ユニット（GPU）12と、ローカルメモリ14と、ディスプレイインターフェース16と、ディスプレイ18と、バス20とを含む。ユーザ入力インターフェース4、CPU6、メモリコントローラ8、GPU12およびディスプレイインターフェース16は、バス20を使用して互いに通信し得る。バス20は、第3世代バス（例えば、HyperTransportバスまたはInfiniBandバス）、第2世代バス（例えば、アドバンストグラフィックスポートバス、周辺構成要素相互接続（PCI）エクスプレスバス、またはアドバンストエクステンシブルインターフェース（AXI：Advanced eXtensible Interface）バス）、あるいは別のタイプのバスまたはデバイスの相互接続などの様々なバス構造のいずれかであり得る。図1に示される異なる構成要素間のバスおよび通信インターフェースの特定の構成は例にすぎず、コンピューティングデバイスの他の構成および／あるいは同じまたは異なる構成要素をもつ他のグラフィックス処理システムが、本開示の技法を実施するために使用され得ることに留意されたい。
30

【0016】

[0019] CPU6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プロセッサまたは専用プロセッサを備え得る。ユーザは、CPU6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ためにコンピューティングデバイス2に入力を与え得る。CPU6上で実行す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ー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。ユーザは、ユーザ入力インターフェース4を介してコンピューティングデバイス2に結合されたキーボード、マウス、マイクロフォン、タッチパッドまたは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、コンピューティングデバイス2に入力を与え得る。
40
50

【0017】

[0020] C P U 6 上で実行するソフトウェアアプリケーションは、ディスプレイ 1 8 へのグラフィックスデータのレンダリングを行うことを C P U 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。いくつかの例において、複数のソフトウェア命令は、例えば、オープングラフィックスライブラリ (O p e n G L (登録商標) : Open Graphics Library) アプリケーションプログラミングインターフェース (A P I) 、オープングラフィックスライブラリ組込みシステムズ (O p e n G L E S : Open Graphics Library Embedded Systems) A P I 、 D i r e c t 3 D A P I 、 X 3 D A P I 、 R e n d e r M a n A P I 、 W e b G L A P I 、あるいは任意の他の公開またはプロプライエタリ(proprietary)標準グラフィックスA P IなどのグラフィックスA P Iに準拠し得る。複数のグラフィックスレンダリング命令を処理するために、C P U 6 は、G P U 1 2 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部を行わせるために、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G P U 1 2 に発行し得る。いくつかの例において、レンダリングされるべきグラフィックスデータは、グラフィックスプリミティブのリスト、例えば、点、線、三角形、四角形、三角形ストリップなどを含み得る。

10

【0018】

[0021] メモリコントローラ 8 は、システムメモリ 1 0 に入るおよびそれから出るデータの転送を容易にする(facilitate)。例えば、メモリコントローラ 8 はメモリ読み取りおよび書き込みコマンドを受け取り、コンピューティングデバイス 2 内の構成要素にメモリサービスを提供するためにメモリ 1 0 に関してそうしたコマンドをサービスし得る。メモリコントローラ 8 はシステムメモリ 1 0 に通信可能に結合される。メモリコントローラ 8 は、C P U 6 およびシステムメモリ 1 0 の両方と別個である処理モジュールとして図 1 の例示的なコンピューティングデバイス 2 内に示されるが、他の例では、メモリコントローラ 8 の機能の一部または全部がC P U 6 およびシステムメモリ 1 0 の一方または両方上で実施され得る。

20

【0019】

[0022] システムメモリ 1 0 は、C P U 6 が実行するためにアクセス可能である複数のプログラムモジュールおよび / または複数の命令、並びに / あるいはC P U 6 上で実行しているプログラムによる使用のためのデータを記憶し得る。例えば、システムメモリ 1 0 は、ユーザアプリケーションと、アプリケーションに関連するグラフィックスデータとを記憶し得る。システムメモリ 1 0 は、コンピューティングデバイス 2 の他の構成要素による使用のための情報、および / または他の構成要素によって生成される情報をさらに記憶し得る。例えば、システムメモリ 1 0 は、G P U 1 2 のためのデバイスマモリとして働き得、G P U 1 2 によってそれに対して演算されるべきデータ、並びにG P U 1 2 によって行われる演算から生じるデータを記憶し得る。例えば、システムメモリ 1 0 は、テクスチャバッファ、深度バッファ、ステンシルバッファ、頂点バッファ、フレームバッファなどの任意の組合せを記憶し得る。さらに、システムメモリ 1 0 は、G P U 1 2 による処理のためのコマンドストリームを記憶し得る。システムメモリ 1 0 は、例えば、ランダムアクセスメモリ (R A M) 、 static RAM (S R A M) 、ダイナミック RAM (D R A M) 、読み取り専用メモリ (R O M) 、消去可能プログラマブル ROM (E P R O M) 、電気的消去可能プログラマブル ROM (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、1 つまたは複数の揮発性または不揮発性メモリまたはストレージデバイスを含み得る。

30

【0020】

[0023] G P U 1 2 は、1 つまたは複数のグラフィックスプリミティブをディスプレイ 1 8 にレンダリングするためにグラフィックス演算を行うように構成され得る。従って、C P U 6 上で実行しているソフトウェアアプリケーションのうちの 1 つがグラフィックス処理を必要とするとき、C P U 6 は、ディスプレイ 1 8 にレンダリングするための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G P U 1 2 に与え得る。グラフィックス

40

50

コマンドは、例えば、描画呼出しなどの描画コマンド、GPU状態プログラミングコマンド、メモリ転送コマンド、汎用コンピューティングコマンド、カーネル実行コマンドなどを含み得る。いくつかの例において、CPU6は、GPU12によってアクセスされ得る、コマンドおよびグラフィックスデータをメモリ10に書き込むことによって、コマンドおよびグラフィックスデータをGPU12に与え得る。いくつかの例において、GPU12は、CPU6上で実行しているアプリケーションのための汎用コンピューティングを行うようにさらに構成され得る。

【0021】

[0024] GPU12は、いくつかの事例において、ベクトル演算の、CPU6よりも効率的な処理を行う高度並列構造で構築され得る。例えば、GPU12は、複数の頂点またはピクセル上で並列様式で演算するように構成された複数の処理要素を含み得る。GPU12の高度並列特質は、いくつかの事例において、GPU12が、CPU6を使用して直接ディスプレイ18にシーンを描画するよりも速く、グラフィックス画像（例えば、GUI、並びに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18上に描画することを可能にし得る。さらに、GPU12の高度並列特質は、GPU12が、CPU6よりも速く、汎用コンピューティングアプリケーションのためのいくつかのタイプのベクトルおよび行列演算を処理することを可能にし得る。

10

【0022】

[0025] GPU12は、いくつかの事例において、コンピューティングデバイス2のマザーボードに組み込まれ得る。他の事例において、GPU12は、コンピューティングデバイス2のマザーボードにおけるポートに設置されたグラフィックスカード上に存在し得るか、または場合によっては、コンピューティングデバイス2と相互動作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。さらなる事例において、GPU12は、システムオンチップ（SOC）を形成するCPU6と同じマイクロチップ上に配置され得る。GPU12とCPU6とは、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（ASIC）、フィールドプログラマブルゲートアレイ（FPGA）、デジタル信号プロセッサ（DSP）、あるいは他の等価な集積またはディスクリート論理回路など、1つまたは複数のプロセッサを含み得る。

20

【0023】

[0026] GPU12はローカルメモリ14に直接結合され得る。従って、GPU12は、必ずしもバス20を使用せずに、ローカルメモリ14からデータを読み取り、ローカルメモリ14にデータを書き込み得る。言い換れば、GPU12は、オフチップメモリの代わりに、ローカルストレージを使用してデータをローカルに処理し得る。これは、GPU12が、重いバストラフィックを経験し得る、バス20を介したデータの読み取りおよび書き込みを行う必要をなくすことによって、GPU12がより効率的な様式で動作することを可能にする。しかしながら、いくつかの事例において、GPU12は、別個のキャッシュを含まないことがあるが、代わりに、バス20を介してシステムメモリ10を利用し得る。ローカルメモリ14は、例えば、ランダムアクセスメモリ（RAM）、静态RAM（SRAM）、ダイナミックRAM（DRAM）、消去可能プログラマブルROM（EPROM）、電気的消去可能プログラマブルROM（EEPROM）、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、1つまたは複数の揮発性または不揮発性メモリあるいはストレージデバイスを含み得る。

30

40

【0024】

[0027] CPU6および/またはGPU12は、レンダリングされた画像データを、システムメモリ10内で割り振られたフレームバッファに記憶し得る。ディスプレイインターフェース16は、フレームバッファからデータを取り出し、レンダリングされた画像データによって表される画像を表示するようにディスプレイ18を構成し得る。いくつかの例において、ディスプレイインターフェース16は、フレームバッファから取り出されたデジタル値を、ディスプレイ18で消費できるアナログ信号に変換するように構成されたデジタルアナログ変換器（DAC）を含み得る。他の例において、ディスプレイинтер-

50

フェース 16 は、処理のために、デジタル値をディスプレイ 18 に直接受け渡し得る。ディスプレイ 18 は、モニタ、テレビジョン、投影デバイス、液晶ディスプレイ（LCD）、プラズマディスプレイパネル、発光ダイオード（LED）アレイ、陰極線管（CRT）ディスプレイ、電子ペーパー、表面伝導電子放出ディスプレイ（SED）、レーザーテレビジョンディスプレイ、ナノ結晶ディスプレイまたは別のタイプのディスプレイユニットを含み得る。ディスプレイ 18 は、コンピューティングデバイス 2 内に組み込まれ得る。例えば、ディスプレイ 18 は、携帯電話ハンドセットまたはタブレットコンピュータのスクリーンであり得る。代替的に、ディスプレイ 18 は、ワイヤードまたはワイヤレス通信リンクを介してコンピューティングデバイス 2 に結合されたスタンドアロンデバイスであり得る。例えば、ディスプレイ 18 は、ケーブルまたはワイヤレスリンクを介してパーソナルコンピュータに接続されたコンピュータモニタまたはフラットパネルディスプレイであり得る。

10

【0025】

[0028] 説明されたように、CPU 6 は、グラフィックス処理を GPU 12 にオフロードし得る。GPU 12 は、次に、グラフィックスデータをレンダリングするために、様々なグラフィックス処理アルゴリズムを行い得る。グラフィックス処理アルゴリズムの例は、視差遮蔽マッピング（POM：parallax occlusion mapping）、スクリーン空間光線追跡（SSRT：screen space ray tracing）、被写界深度（DoF：depth of field）処理、ボリュームレンダリング、あるいは動的高さフィールドを用いた水または地形レンダリングを含む。追加のグラフィックス処理アルゴリズムも存在し、上記は、いくつかの例として与えられるにすぎない。

20

【0026】

[0029] 例示的なグラフィックス処理アルゴリズムにおけるグラフィックス処理の一部は、テクスチャリングを含む。テクスチャリングは、テクスチャバッファからビットマップを取り出し、ビットマップをグラフィカルオブジェクトの上に重ねる、テクスチャユニットを伴う。いくつかの例において、GPU 12 はテクスチャユニットを含むが、テクスチャユニットは GPU 12 の外部にあり得る。いくつかの例において、テクスチャユニット、GPU 12、および CPU 6 は全て、同じ集積回路（IC）またはマイクロコントローラーの一部であり得る。本開示において、テクスチャユニットは、GPU 12 の内部にあるものとして説明される。

30

【0027】

[0030] グラフィックス処理を行うために、GPU 12 のシェーダプロセッサはシェーダプログラムの複数の動作を実行し得る。シェーダプログラムのこれら動作の実行の一部は、テクスチャユニットへの繰り返されるアクセスを含み得る。例えば、シェーダプロセッサによって実施されるグラフィックス処理アルゴリズムは、以下の構造、すなわち反復カウントに対して上限をもつループ、ループにおける終了条件、テクスチャ座標を進める／変更するための単純ロジック、および動作の結果を計算するための単純ロジック、を形成する複数の動作を含み得る。

【0028】

[0031] これら動作を行う際に、ループの各反復について、シェーダプロセッサは、データを取り出し、そのデータに対する処理を行い、そのデータをシェーダプロセッサに戻すように出力するためにテクスチャユニットへの要求を出力し得る。これは、テクスチャユニットへの複数の要求を生じ、これは、テクスチャユニットが GPU 12 の外部にある例において、電力と、クロックサイクルと、GPU 12 との接続線の帯域幅またはバス 20 の帯域幅とを消費する。

40

【0029】

[0032] 本開示で説明される技法では、GPU 12 のシェーダプロセッサによって実行されるべきである複数の動作が、代わりに、テクスチャユニットによって実行される。例えば、GPU 12 のシェーダプロセッサが実行すべきであったこれら動作は、条件が満たされるまで繰り返し実行されるべきであるこれら動作（例えば、反復カウントに対する上

50

限をもつ複数の動作のループ)を含む。テクスチャユニットは、代わりに、シェーダプロセッサによる繰り返される要求の必要なしに、条件が満たされるかまたは満たされなくなるまで、これらの命令を繰り返し実行し得る。例えば、条件は、上限が到達されるまで繰り返す(例えば、 $A < B$ である限り繰り返す)ことであり得る。この場合、テクスチャユニットは、条件が満たされる限り繰り返し実行する。条件は、条件がもはや満たされなくなるまで繰り返す(例えば、 $A = B$ まで繰り返す)ことであり得る。この場合、テクスチャユニットは、条件が満たされない限り繰り返し実行する。

【0030】

[0033] 例えば、テクスチャユニット(例えば、GPU12内またはGPU12外にあるもの)は、命令において定義された条件(例えば、反復カウントの上限に到達することなど)の生起に基づいて複数の動作を繰り返し実行することをテクスチャユニットに命令する命令を受け取り得る。テクスチャユニットは、命令において定義された条件が満たされるかまたは満たされなくなるまで、これら動作を繰り返し実行し得、これら動作の繰り返される実行から生じるデータを出力し得る。

10

【0031】

[0034] 一例として、テクスチャユニットは、これら動作の実行の第1の反復中に(例えば、テクスチャバッファからの)テクセル値を読み取り得る。テクスチャユニットは、テクセル値の、命令において定義された変数との比較に基づいて、条件が満たされるのか満たされないのかを決定し、命令において定義された条件が満たされるのか満たされないのかの決定に基づいて、これら動作の実行の第2の反復が必要とされるかどうかを決定し得る。

20

【0032】

[0035] これら動作の繰り返される実行は、動作の実行の第2の反復が必要とされるという決定に基づいて、テクスチャユニットの出力をフィードバック信号としてテクスチャユニットの入力に出力するテクスチャユニットを含む。テクスチャユニットは、これら動作の実行の第2の反復が必要とされないという決定に基づいて、これら動作の繰り返される実行から生じるデータをGPUに出力し得る。

20

【0033】

[0036] このように、シェーダプロセッサがテクスチャユニットへの1つの命令を発行して、テクスチャユニットにこれら動作を繰り返し実行させることができることから、シェーダプロセッサの作業量は、シェーダプロセッサがループにおいてこれら動作を実行しなければならない例と比較して低減され得る。これはまた、ループを形成したこれら動作の全てが、テクスチャユニットへの単一命令として表され得るので、より少ないシェーダコードが、GPU12の命令キャッシュに記憶されることになり得る。

30

【0034】

[0037] 電力使用量の減少および処理効率の増加もあり得る。シェーダプロセッサがより少數の動作を実行するので、シェーダプロセッサはより少ない電力を消費し得る。テクスチャユニットの処理ハードウェアユニット(例えば、算術論理ユニット(ALU:arithmetic logic unit))は、シェーダプロセッサと比較してより電力効率的であり得、従って、これら動作の実行をテクスチャユニットにシフトすることによって、電力の全体的な低減があり得る。テクスチャユニットの処理ハードウェアユニットはまた、シェーダプロセッサよりも高いスループットを与え、その結果、テクスチャユニットがこれら動作を繰り返し実行する場合よりもこれら動作のより速い処理をもたらし得る(例えば、テクスチャユニットは、シェーダプロセッサからの複数の命令を待つアイドルでなく、シェーダプロセッサに繰り返し出力するクロックサイクルを浪費する必要がない)。

40

【0035】

[0038] シェーダプロセッサが、ループの条件が満たされるまでループのこれら動作を繰り返し実行する例と比較して、シェーダプロセッサの汎用レジスタ(GPR:general purpose register)のサイズも低減され得る。GPRは、シェーダプロセッサが1つの動作の実行から生じるデータを一時的に記憶するために使用するレジスタである。シェーダ

50

プロセッサがこれら動作を繰り返し実行すべきである場合、シェーダプロセッサは、各動作について得られたデータをG P Rに記憶し、比較的大きいG P Rが、実行の各反復から生じるデータを記憶することを必要とすることになる。本開示で説明される例示的な技法があれば、テクスチャユニットは、実行の反復から生じる何らかの中間データを記憶し、シェーダプロセッサのG P Rが他の目的のために使用されること、またはG P Rのサイズが低減されることを可能にする。

【0036】

[0039] 図2は、図1のコンピューティングデバイス2のC P U 6、G P U 1 2、およびメモリ1 0をさらに詳細に示すブロック図である。図2に示されるように、C P U 6はG P U 1 2およびメモリ1 0と通信可能に結合され、G P U 1 2はC P U 6およびメモリ1 0と通信可能に結合される。G P U 1 2は、いくつかの例において、マザーボード上でC P U 6と一体化され得る。追加の例において、G P U 1 2は、C P U 6を含むマザーボードのポートに設置されたグラフィックスカード上に実装され(implemented)得る。さらなる例において、G P U 1 2は、C P U 6と相互動作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。追加の例において、G P U 1 2は、システムオンチップ(S o C)を形成するC P U 6と同じ集積回路またはマイクロプロセッサ上に配置され得る。C P U 6は、ソフトウェアアプリケーション(A p p)2 4と、グラフィックスA P I 2 6と、G P U ドライバ2 8と、オペレーティングシステム3 0とを実行するように構成される。
10

【0037】

[0040] G P U 1 2は、コントローラ3 2と、テクスチャユニット3 4と、シェーダプロセッサ3 6と、1つまたは複数の固定機能ユニット3 8と、ローカルメモリ1 4とを含む。図2において、ローカルメモリ1 4およびテクスチャユニット3 4は、G P U 1 2内にあるように示されるものの、ローカルメモリ1 4およびテクスチャユニット3 4はG P U 1 2外にあってもよい。
20

【0038】

[0041] ソフトウェアアプリケーション2 4は、各々、グラフィックコンテンツが表示されるようにさせる1つまたは複数の命令、または非グラフィックタスク(例えば、汎用コンピューティングタスク)がG P U 1 2上で行われるようにさせる1つまたは複数の命令のうちの少なくとも1つを含み得る。ソフトウェアアプリケーション2 4は、複数の命令をグラフィックスA P I 2 6に発行し得る。グラフィックスA P I 2 6は、ソフトウェアアプリケーション2 4から受け取ったこれら命令を、G P U ドライバ2 8で消費できるフォーマットに変換するランタイムサービスであり得る。
30

【0039】

[0042] G P U ドライバ2 8は、グラフィックスA P I 2 6を介して、ソフトウェアアプリケーション2 4からこれら命令を受け取り、これら命令をサービスするためにG P U 1 2の動作を制御する。例えば、G P U ドライバ2 8は、1つまたは複数のコマンドストリームを作り、コマンドストリームをメモリ1 0に入れ、コマンドストリームを実行することをG P U 1 2に命令し得る。G P U ドライバ2 8は、コマンドストリームをメモリ1 0に入れ、オペレーティングシステム3 0を介して、例えば、1つまたは複数のシステム呼出しを介してG P U 1 2と通信し得る。
40

【0040】

[0043] コントローラ3 2は、G P U 1 2のハードウェアであり得るか、G P U 1 2上で実行するソフトウェアまたはファームウェアであり得るか、あるいはその両方の組合せである。コントローラ3 2は、G P U 1 2の様々な構成要素の複数の動作を制御し得る。例えば、コントローラ3 2は、複数の命令およびデータが構成要素に与えられたときに制御し、複数の命令およびデータの受け取りを制御し、G P U 1 2からのデータの出力を制御し得る。

【0041】

[0044] シェーダプロセッサ3 6および固定機能ユニット3 8は一緒に、G P U 1 2がグラフィックス処理をそれを介して行うグラフィックス処理パイプラインを形成するグラ
50

フィックス処理段を与える。シェーダプロセッサ36は、プログラマブルフレキシビリティを与えるように構成され得る。例えば、シェーダプロセッサ36は、CPU6を介してGPU12上にダウンロードされた1つまたは複数のシェーダプログラムを実行するよう構成され得る。シェーダプログラムは、いくつかの例において、例えば、OpenGLシェーディング言語(GLSL)、高レベルシェーディング言語(HLSL)、グラフィックスのためのC(Cg)シェーディング言語のような高レベルシェーディング言語で書き込まれたプログラムのコンパイルされたバージョンであり得る。

【0042】

[0045] いくつかの例において、シェーダプロセッサ36は、例えば、单一命令複数データ(SIMD)パイプラインとして、並列に動作するよう構成された複数の処理ユニットを含む。シェーダプロセッサ36は、複数のシェーダプログラム命令を記憶するプログラムメモリと、処理されるべきであるデータおよび得られたデータを記憶する汎用レジスタ(GPR)と、実行状態レジスタ、例えば、実行されているプログラムメモリにおける現在の命令またはフェッチされるべき次の命令を示すプログラムカウンタレジスタとを有し得る。シェーダプロセッサ36上で実行するシェーダプログラムの例は、例えば、頂点シェーダ、ピクセルシェーダ、ジオメトリシェーダ、ハルシェーダ、ドメインシェーダ、コンピュートシェーダ、および/またはユニファイドシェーダを含む。

10

【0043】

[0046] 1つまたは複数の固定機能ユニット38は、いくつかのファンクションを行うために配線接続されたハードウェアを含み得る。固定機能ハードウェアは、1つまたは複数の制御信号を介して、例えば、異なるファンクションを行うように構成可能であり得るが、1つまたは複数の固定機能ユニット38の固定機能ハードウェアは、一般に、ユーザによってコンパイルされたプログラムを受け取ることが可能であるプログラムメモリを含まない。いくつかの例において、1つまたは複数の固定機能ユニット38は、例えば、深度テスト、シザーテスト、アルファブレンディングのようなラスタ動作を行う、例えば、処理ユニットを含む。

20

【0044】

[0047] GPU12はまた、GPU12のうちの1つのハードウェアユニットであってテクスチャリングアルゴリズムにおいて使用されるテクスチャユニット34を含む。テクスチャリングは、システムメモリ10の一部であり得るテクスチャバッファからビットマップを取り出すことと、そのビットマップを処理することと、この処理されたビットマップをグラフィカルオブジェクトの上に置くことを含み得る。ビットマップは、シェーダプロセッサ36または1つまたは複数の固定機能ユニット38が、グラフィカルオブジェクトの上に置くことができるような、テクスチャユニット34が処理する2次元画像と見なされ得る。ビットマップのピクセルはテクセルと呼ばれ得、テクスチャユニット34がシェーダプロセッサ36への出力のために生成するデータはテクセルデータと呼ばれ得る。

30

【0045】

[0048] 単純な例として、ビットマップは、世界地図の平坦化された2次元画像であり得る。テクスチャユニット34は、世界地図のこの2次元画像を処理し得、(例えば、シェーダプロセッサ36および/または固定機能ユニット38を介した)GPU12は、この画像をグラフィカルな地球を形成する球状グラフィカルオブジェクトの上に置き得る。

40

【0046】

[0049] グラフィカルな地球を形成するためにテクスチャリングを使用することは一例であるが、テクスチャリングの様々な他の例があり得る。テクスチャリングアルゴリズムのいくつかの例は、視差遮蔽マッピング(POM)、スクリーン空間光線追跡(SSRT)、被写界深度(DoF)、ボリュームレンダリング、および動的高さフィールドを用いた水/地形レンダリングを含む。本開示で説明される例は、これらのテクスチャリングアルゴリズム、これらのテクスチャリングアルゴリズムのサブセット、これらの例の他のテクスチャリングアルゴリズム、または上記の任意の組合せに適用可能である。

50

【0047】

[0050] テクスチャリングを行うために、アプリケーション24は、グラフィックスAPI26への命令を発行し、次に、GPUドライバ28への命令を発行し得る。GPUドライバ28は、データの処理を行うことをテクスチャユニット34に命令するテクスチャユニット34への呼出しを含む複数の動作を実行するために複数の命令をシェーダプロセッサ36に発行し得る。いくつかの例において、テクスチャリングを行うために、シェーダプロセッサ36は、満たされるべき条件（例えば、ループの反復カウントに対する限界が満たされるまでループが続く）を有するループされたファンクション（looped function）（例えば、「while」ループまたは「for」ループなど）を実行し得る。各反復中に、シェーダプロセッサ36は、テクスチャユニット34への呼出しを出力し、テクスチャユニット34からデータを受け取り得る。

10

【0048】

[0051] 一例として、ループされたファンクションの構造は、以下の通りであり得る。

【0049】

【数1】

```
initialize
while(条件)
{
    texOffsets
    loopBody
    サンプルテクスチャ
}
```

20

【0050】

[0052] ループされたファンクションをさらに示すために、以下は、シェーダプロセッサ36がPOMレンダリングのために実行する複数の動作の一例である。

【0051】

【数2】

```
float height = read_imagef(heightMap, tex).x;
float prevHeight=高さ,
while(currentLayerHeight>高さ)
{
    texOffset += dTex;
    prevHeight=高さ,
    高さ=read_imagef(heightMap, tex+texOffset).x
    currentLayerHeight -= layerHeight;
}
```

30

40

【0052】

[0053] 上記の例において、シェーダプロセッサ36が実行するシェーダプログラムは、「read_imagef」ファンクションを介して1つの動作を実行することをシェーダプロセッサ36にさせる。「read_imagef」ファンクションは、ロケーションtex_textureOffsetにおいてheightmapテクスチャをサンプリングするために使用され、tex_offset+=dTexはテクスチャ座標を変更する。わかるように、シェーダプロセッサ36は、currentLayerHeightが高さに等しくなるという条件が満たされるまで、「read_imagef」ファンクションを繰り返し実行する。各実行中に、テクセル値が読み取られ（例えば、heightma

50

p 値)、変数(例えば、高さ)に割り当てられる。条件が満たされるかどうかは、テクセル値の、変数との比較(例えば、currentLayerHeightが高さよりも大きいかどうかを決定するために、高さは、currentLayerHeightと比較される)に基づく。

【0053】

[0054] POMレンダリングのための複数の動作を行うために、シェーダプロセッサ36は、テクスチャユニット34への要求を出力し(例えば、read_imagefの動作を実行することをテクスチャユニット34に命令する)、次に、GPRに記憶するためのテクセルデータを受け取り得る。シェーダプロセッサ36は、上記のコードにおいて複数の追加動作を行い、条件が満たされるかどうかを決定し得る。条件がまだ満たされている場合、シェーダプロセッサ36は、テクスチャユニット34への要求を繰り返し、次に、GPRに記憶するためのテクセルデータを受け取り、条件が満たされるかどうかに基づいてこれらのステップを繰り返し続け得る。10

【0054】

[0055] いくつかの例において、シェーダプロセッサ36は、条件が満たされるまで複数の動作にわたってループ(例えば、上限が到達されるまでループ)し得るか、または条件が満たされる限り(例えば、第1の値が第2の値未満である限り)これら動作にわたってループし得る。これらの例において、シェーダプロセッサ36は、条件が、満たされること(例えば、条件が満たされる限り)、または満たされなくなること(例えば、条件が満たされるまで)に基づいて、これら動作にわたってループし得る。20

【0055】

[0056] 条件が満たされることは、「whileループ」の一部であり得、条件が満たされないことは、「doループ」の一部であり得る。例えば、条件は、A < Bである間、複数の動作の1つのセットを行う、であり得る。この場合、テクスチャユニット34は、条件が満たされることに基づいて、これら動作を繰り返し実行し得る(例えば、AがB未満である場合、テクスチャユニット34はこれら動作のさらなる反復を実行する)。別の例として、条件は、A Bまで繰り返すことであり得る。この場合、テクスチャユニット34は、条件が満たされないことにに基づいて、これら動作を繰り返し実行し得る(例えば、AがBに等しくないかまたはそれよりも大きくない場合、テクスチャユニット34はこれら動作のさらなる反復を実行する)。本開示で説明される技法は、両方の場合(例えば、どのようにループが定義されるかというファンクションである、条件が満たされることに基づいて繰り返し実行することと、条件が満たされないことにに基づいて繰り返し実行することと)に適用可能である。容易さのために、本説明は、テクスチャユニット34が、条件が満たされることに基づいて繰り返し実行する場合を指し得るが、そのような説明は、テクスチャユニット34が、条件が満たされないことにに基づいて繰り返し実行する場合に技法が適用可能でないことを意味すると解釈されるべきでない。30

【0056】

[0057] シェーダプロセッサ36による、テクスチャユニット34への繰り返される呼出しは、シェーダプロセッサ36の作業量を増加させ、ループが実行されている間、他の目的のために利用不可能である比較的大きいGPRをシェーダプロセッサ36が含むことを必要とし、並びにループの複数の動作の全てを記憶するためにローカルメモリ14においてより大きい命令キャッシュを使用し得る。本開示で説明される技法では、シェーダプロセッサ36が、テクスチャユニット34へのアクセスを伴うこれら動作を繰り返し実行するだけでなく、テクスチャユニット34がシェーダプロセッサ36からのアクセスに応答してこれら動作を繰り返し実行し、シェーダプロセッサ36がテクスチャユニット34に繰り返しアクセスする必要がないように構成され得る。40

【0057】

[0058] 例えば、テクスチャユニット34は、シェーダプロセッサ36による単一のアクセスに応答して、複数の動作を繰り返し実行するように構成され得る。そのような複数の動作のうちの少なくともいくつかは、従来、シェーダプロセッサ36による複数の多重

10

20

30

40

50

アクセスの各々に応答して行われること、例えば、1つのアクセスに応答した1つの動作になる。対照的に、本開示の様々な例によれば、テクスチャユニット34は、シェーダプロセッサ36による、テクスチャユニット34への所与のアクセスに応答して、複数の動作(multiple operations)を実行し得る。

【0058】

[0059] シェーダプロセッサ36によるテクスチャユニット34へのアクセスの数を低減して複数の動作の1つのセットを遂行することによって、シェーダプロセッサ36の作業量が低減され得る。さらに、いくつかの場合に、テクスチャユニット34は、条件テスト(例えば、条件検査)、ループにおける数学演算、および他のそのようなファンクションをより高いスループットで行い、シェーダプロセッサ36よりも少ない電力を利用することが可能であり得る。10

【0059】

[0060] 例えば、POMレンダリングの例示的なシェーダコード(例えば、シェーダプログラムの複数の動作)は、シェーダプロセッサ36がテクスチャユニット34に出力する1つの単一命令(例えば、シェーダプロセッサ36がテクスチャユニット34にアクセスするという1つのインスタンス)であり得る。一例として、シェーダプロセッサ36がテクスチャユニット34に出力し得る命令は、以下の通りであり得る。

【0060】

【数3】

float4結果=textureLoop(heightMap, tex, dTex, layerHeight, 条件…)

【0061】

[0061] シェーダプロセッサ36がテクスチャユニット34に出力する命令中に、シェーダプロセッサ36は、テクスチャユニット34が行うべきである複数の動作の変数(例えば、heightMap、tex、dTex、layerHeight)、並びに条件の定義を含める。次に、テクスチャユニット34は、シェーダプロセッサ36から受け取られた命令において定義された条件が満たされることに基づいて、これら動作を繰り返し実行し得る。

【0062】

[0062] テクスチャユニット34がこれら動作を繰り返し実行する場合、テクスチャユニット34は、単一のファンクション呼出しが、複数の動作の特定セットを行うことをテクスチャユニット34に命令するものであると認識するように構成される必要があり得る。例えば、テクスチャユニット34の設計中に、テクスチャユニット34は、テクスチャユニット34が特定の名前を有するファンクションを受け取るか、あるいは複数の変数の特定セットまたは複数の変数の順序を有するファンクションを受け取った場合、テクスチャユニット34は複数の動作の特定セットを繰り返し実行すべきであるように、設計され得る。一例として、テクスチャユニット34がtextureLoopファンクションを含む命令を受け取った場合、テクスチャユニット34は、テクスチャユニット34が、シェーダプロセッサ36によって実行されるものとして上記で説明されたもののような複数の動作を繰り返し実行すべきであると決定し得る。テクスチャユニット34が異なるファンクション(例えば、SSRTのためのファンクション)を含む命令を受け取った場合、テクスチャユニット34は、通常ならば、シェーダプロセッサ36によって実行されるであろう複数の動作を、テクスチャユニット34が繰り返し実行すべきであると決定し得る。30

。

【0063】

[0063] テクスチャユニット34は、異なるタイプのテクスチャリングのための複数の動作を繰り返し実行するように予め構成され得、より多くのテクスチャリングアルゴリズムが開発されるにつれて、テクスチャユニット34は、これらのテクスチャリングアルゴリズムのための複数の動作も繰り返し実行するように構成され得る。より一般的には、例はテクスチャリングアルゴリズムのためのものであるとして説明されるが、本開示で説明される技法はそのように限定されない。例えば、本開示で説明される技法は、複数のル40

10

20

30

40

50

普動作がテクスチャリング目的のために使用されていない場合でも、テクスチャユニット34へのアクセスを必要とするこれらループ動作が使用される、他の場合に拡張され得る。このようにして、テクスチャユニット34は、プログラム可能なテクスチャ処理ユニットにより厳密に匹敵し得る様式で構成され得る。

【0064】

[0064] G P U 1 2 のための開発者ガイドは、テクスチャユニット34が、どのループされた複数の動作を行うように構成されたかを示す情報と、これら動作をテクスチャユニット34に行わせるためのファンクション呼出しのための命令とを含み得る。テクスチャ座標を変更するために使用されるシェーダプログラムの開発中に、開発者は、ループされた複数の動作でなく、ファンクション呼出しのための命令をシェーダプログラムのコードに含め得る。10

【0065】

[0065] 別の例として、開発者が、ループされたこれら動作を実行するためにテクスチャユニット34の能力を活用することに依拠するだけでなく、シェーダプログラムをコンパイルする、C P U 6 上で実行するコンパイラが、ループされたこれら動作を、テクスチャユニット34への特定のファンクション呼出しを含む単一命令にコンパイルし得る。代替的にまたは追加として、G P U ドライバ28またはG P U ドライバ28のためのラッパーは、シェーダプログラムのコードの高水準言語を読み取り、テクスチャユニット34が実行するように構成された特定のループされた複数の動作を含むコード中の場所を決定するように構成され得る。G P U ドライバ28またはG P U ドライバ28のためのラッパーは、ループされたこれら動作をテクスチャユニット34に実行させるための特定のファンクション呼出しをもつ単一命令を含めるためにシェーダプログラムのコードを変更し得る。20

【0066】

[0066] 従って、テクスチャユニット34は、命令において定義された条件が満たされること（または満たされないこと）に基づいて複数の動作を繰り返し実行することをテクスチャユニット34に命令する、シェーダプロセッサ36によって出力された命令を受け取るように構成され得る。これら動作は、シェーダプログラムの複数の動作であり、テクスチャ座標を変更するための複数の動作を含み得る。

【0067】

[0067] テクスチャユニット34は、この命令において定義された条件が満たされることまたは満たされないことに基づいて（例えば、条件が満たされる限り、または条件が満たされるまで）、複数の動作を繰り返し実行し、これら動作を実行するために何も追加命令をシェーダプロセッサ36から受け取らずに繰り返し実行し得る。このようにして、シェーダプロセッサ36の作業量、並びにシェーダプロセッサ36とテクスチャユニット34との間の対話の頻度が低減され得る。30

【0068】

[0068] テクスチャユニット34は、これら動作の繰り返される実行から生じるデータを出力し得る。例えば、テクスチャユニット34は、これら動作の繰り返される実行の全ての反復が完了した後にのみ、データをシェーダプロセッサ36に出力し得る。言い換えれば、テクスチャユニット34は、ループが完了するまで、繰り返される実行から生じるデータを出力しないことがある。従って、テクスチャユニット34がシェーダプロセッサ36に出力する必要がある回数も制限され得る。しかしながら、いくつかの例において、テクスチャユニット34は、周期的にまたは1つの反復の終わりにおいて、実行から生じるデータをシェーダプロセッサ36に出力し得る。従って、繰り返される実行から生じるデータの出力の例としては、全ての反復が完了した後の最終データ、または繰り返される実行中の周期的なものを含む。40

【0069】

[0069] 図3は、図2のテクスチャユニットの一例をさらに詳細に示すブロック図である。図3において、シェーダプロセッサ36は、命令において定義された条件が満たされることまたは満たされないことに基づいて複数の動作を繰り返し実行することをテクスチ50

ヤユニット 34 に命令する 1 つの命令を、テクスチャユニット 34 に出力し得る。例えば、シェーダープロセッサ 36 がテクスチャユニット 34 に出力する命令は、テクスチャユニット 34 がそれについて動作する複数の変数と、ループされたこれら動作がいつ完了するかを定義する条件とを含む。

【0070】

[0070] 図示のように、テクスチャユニット 34 は、入力ユニット 40 と、(テクスチャユニット 34 のローカルキャッシュまたはローカルメモリ 14 の一部であり得る) キャッシュ 42 と、フォーマッティングユニット 44 と、フィルタユニット 46 と、カラーフォーマットユニット 48 と、出力ユニット 50 とを含む。例示的なテクスチャユニット 34 において、出力ユニット 50 と入力ユニット 40 とは、フィードバック信号 52 を介して互いに接続される。より詳細に説明されるように、フィードバック信号 52 は、条件が満たされたのか満たされないのかを決定するための機構を与える。

10

【0071】

[0071] 図 3 に示されるテクスチャユニット 34 のユニットは、理解を容易にするために示される。異なるタイプのテクスチャユニット 34 は、図示されるものに対して多かったり、少なかつたり、または異なったりするユニットを含み得、構成要素間の相互接続は必ずしも図示のようである必要があるとは限らない。本開示で説明される技法は、テクスチャユニット 34 のそのような例にも適用可能である。

【0072】

[0072] (例えば、テクスチャユニット 34 が、シェーダープロセッサ 36 によって一般に行われる複数の動作を行うために再利用されない) 通常動作において、入力ユニット 40 は、アドレス指定目的のために使用され得る。入力ユニット 40 は、(u, v) 座標をメモリアドレスに変換し得る。キャッシュ 42 は、入力ユニット 40 によってアドレス指定された情報を記憶し得る。フォーマッティングユニット 44 は、テクスチャリングアルゴリズムによって定義されるように、ビットマップに対して様々なフォーマッティングを行い得る。フィルタユニット 46 は、双線形フィルタ処理 / 補間を行い得る。カラーフォーマットユニット 48 は、ビットマップの色をフォーマットし得る。出力ユニット 50 は、カラーフォーマットユニット 48 からの出力を受け取り、テクセルデータを出力するためのシェーダープロセッサ 36 に対する出力インターフェースである。

20

【0073】

[0073] 但し、本開示で説明される例示的な技法において、これら様々なユニットは、シェーダープログラムの複数の動作を繰り返し実行に再利用され得る。例えば、上記で説明されたように、ループされたこれら動作の構造は以下の通りであり得る。

30

【0074】

【数 4】

```

initialize
while(条件)
{
    texOffsets
    loopBody
    サンプルテクスチャ
}

```

40

【0075】

[0074] いくつかの例において、入力ユニット 40 は、初期化および textureOffset 動作を行うように構成され得る。フィルタユニット 46 は、条件の動作を行うように構成され得る。カラーフォーマットユニット 48 は、loopBody の動作を行うように構成され得る。出力ユニット 50 は、条件が満たされるかどうかを決定するように構成され得る。

50

【0076】

[0075] 例えば、出力ユニット50は、命令において定義された条件が満たされるのか満たされないのかに基づいて、複数の動作の実行の反復が必要とされるかどうかを決定するように構成され得る。whileループの例において、出力ユニット50は、満たされるべき条件が依然として真であるかどうかを決定し得る。満たされるべき条件が依然として真である場合、出力ユニット50は、これら動作の実行の反復（すなわち、ループにわたるさらなるパス）が必要とされると決定し得る。この場合、これら動作を繰り返し実行するために、出力ユニット50は、これら動作の実行の反復が必要とされるという決定に基づいて、テクスチャユニット34から、入力ユニット40にフィードバック信号52を出力（例えば、ループの1つの反復から生じるデータを出力）するように構成される。満たされるべき条件が偽である場合、出力ユニット50は、これら動作の実行の反復がもはや必要とされない（すなわち、ループが完了した）と決定し得る。この場合、出力ユニット50は、これら動作の実行の反復が必要とされないという決定に基づいて、これら動作の繰り返される実行から生じるデータを出力する。いくつかの例において、入力ユニット40は、フィードバック信号52に、シェーダプロセッサ36からのどの出力よりも高い優先度を与えるように構成され得る。

10

【0077】

[0076] いくつかの例において、ループの実行の各反復中に、テクスチャユニット34はテクセル値を読み取り得る。例えば、入力ユニット40またはフォーマッティングユニット44は、（例えば、ローカルメモリ14または可能なシステムメモリ10内にある）テクスチャバッファからの、またはキャッシュ42からのテクセル値を読み取るように構成され得、各反復中にテクセル値を読み取り得る。出力ユニット50は、反復のさらなるラウンドが必要とされるかどうかを決定するために、この読み取られたテクセル値（または読み取られたテクセル値の処理されたバージョン）を、命令において定義された変数と比較し得る。これらの例において、読み取られたテクセル値は、命令において定義された変数との比較に基づいて、実行のより多くの反復が必要とされるかどうかを制御する。

20

【0078】

[0077] 例えば、読み取りユニット（例えば、入力ユニット40および/またはフォーマッティングユニット44、あるいは場合によってはテクスチャユニット34の何らかの他のユニット）は、複数の動作の実行の第1の反復中にテクセル値を読み取り得る。出力ユニット50は、テクセル値に基づく値（例えば、テクセル値自体またはテクセル値を処理することから決定された値）と、命令において定義された変数との比較に基づいて、条件が満たされるかどうかを決定し得る。出力ユニット50は、命令において定義された条件が満たされるかどうかの決定に基づいて、これら動作の実行の第2の反復が必要とされるかどうかを決定し得る。

30

【0079】

[0078] 一例において、これら動作を繰り返し実行するために、出力ユニット50は、これら動作の実行の第2の反復が必要とされるという決定に基づいて、入力ユニット40にフィードバック信号を出力し得る。別の例において、出力ユニット50は、これら動作の実行の第2の反復が必要とされないという決定に基づいて、これら動作の繰り返される実行から生じるデータを出力し得る。

40

【0080】

[0079] 一例として、上記のPOMアルゴリズムの場合、whileループの条件はwhile (currentLayerHeight > 高さ)であり、loopBodyは高さ = read_imagef (heightMap.tex + texOffset)であった。この例では、読み取りユニット（例えば、2つの非限定的な例として、入力ユニット40またはフォーマッティングユニット44）が、これら動作の第1の実行中にテクセル値（例えば、heightMap.tex + texOffsetにおいて記憶された値）を読み取り得る。

【0081】

50

[0080] `while` ループのこれら動作の実行の 1 つの反復の後に、出力ユニット 50 は、`currentLayerHeight` が依然として高さよりも大きいかどうかを決定し得る。例えば、出力ユニット 50 は、テクセル値に基づく値（この場合、高さの値、テクセル値自体）と、命令において定義された変数（例えば、`currentLayerHeight`）との比較に基づいて、条件が満たされるかどうかを決定し得る。出力ユニット 50 は、命令において定義された条件が満たされるかどうかの決定に基づいて、これら動作の実行の第 2 の反復が必要とされるかどうかを決定し得る。

【0082】

[0081] 例えば、真である（例えば、`currentLayerHeight` が依然として高さよりも大きい）場合、出力ユニット 50 は、前に計算されたような高さの値を、フィードバック信号 52 として入力ユニット 40 に出力し得、従って、テクスチャユニット 34 のユニットはこれら動作の反復を実行する。従って、これら動作を繰り返し実行するため、出力ユニット 50 は、これら動作の実行の第 2 の反復が必要とされるという決定に基づいて、入力ユニット 40 にフィードバック信号 52 を出力するように構成され得る。この処理は、条件がもはや真でなくなるまで繰り返す。

10

【0083】

[0082] 偽である（例えば、`currentLayerHeight` がもはや高さよりも大きくない）場合、出力ユニット 50 は、高さの最終値を、これら動作の繰り返される実行を介して決定されたものとして、シェーダプロセッサ 36 に出力し得る。従って、出力ユニット 50 は、これら動作の実行の第 2 の反復が必要とされないという決定に基づいて、これら動作の繰り返される実行から生じるデータを出力し得る。

20

【0084】

[0083] この例において、第 1 の反復および第 2 の反復は、理解を支援する様式として使用される。複数の反復があり得、各反復について、出力ユニット 50 は、読み取られたテクセル値に基づく値（例えば、テクセル値自体または処理されたテクセル値）と、命令において定義された変数との間の比較に基づいて、入力ユニット 40 にフィードバック信号 52 を出力し得る。さらなる反復が必要とされる場合、出力ユニット 50 は、入力ユニット 40 にフィードバック信号 52 を出力し得、さらなる反復が必要とされない場合、出力ユニット 50 は GPU 12 に出力し得る。

30

【0085】

[0084] 上記は、POM を使用する一例を与えた。以下は、POM を使用することの別の例を含む、いくつかの追加の例示的な使用を与える。さらに、POM の上記の例は、テクスチャユニット 34 のどのユニットがどの複数の動作を行うかについての一例を与えた。しかしながら、本開示で説明される技法はそのように限定されず、テクスチャユニット 34 のユニットは、上記で与えられた POM の例とは異なる複数の動作を行い得る。いくつかの例において、これら動作は、テクスチャ座標（例えば、`texOffset`）を変更するための複数の動作を含む。また、本開示で説明される技法は、テクスチャリングに限定されると見なされるべきでなく、光線追跡および他の例など、他の目的のために使用され得る。いくつかの例において、出力ユニット 50 は、これら動作の実行の全ての反復の完了後のみでなく、周期的にシェーダプロセッサ 36 にデータを出力し得る。

40

【0086】

[0085] POM の別の例として、ループ構造において、初期化動作は `currHeight = 1` であり、条件は (`currHeight > 高さ`、および `currHeight > 0`) であり、`texOffset` 動作は `texCoord += dTex` であり、`loopBody` は `currHeight -= layerHeight` である。この例において、入力ユニット 40 は、`currHeight = 1` を設定し、`texCoord += dTex` の動作を行い得る。フォーマッティングユニット 44 は、`currHeight > 高さ` の条件が満たされるかどうかに基づいて、真値または偽値を割り当て得る。フィルタユニット 46 は、`currHeight -= layerHeight` の動作を行い得る。

50

【0087】

[0086] 出力ユニット 50 は、命令において定義された条件が満たされるかどうかに基づいて（例えば、フォーマッティングユニット 44 によって行われた真または偽の決定に基づいて）、これら動作の実行の反復が必要とされるかどうかを決定し得る。この例において、これら動作を繰り返し実行するために、出力ユニット 50 は、これら動作の実行の反復が必要とされるという決定に基づいて、テクスチャユニット 34 から、入力ユニット 40 に（例えば、これら動作の 1 つの反復からのデータ）フィードバック信号 52 を出力する。場合によっては、データを出力するために、出力ユニット 50 は、これら動作の実行の反復が必要とされないという決定に基づいて、これら動作の繰り返される実行から生じるデータを出力する。各場合において、ループのさらなる反復が必要とされるかどうかは、テクセル値（またはテクセル値から決定された値）と命令において定義された変数との比較に基づき得る。

10

【0088】

[0087] スクリーン空間光線追跡 (SSRT) の一例として、初期化動作がないことがある。条件は ($P.x * stepDir \text{ endP}$ および $stepCount > maxSteps$) である。 $texOffset$ 動作は、 $(P, Q.z, k) += (dP, dQ.z, dK)$ である。 $loopBody$ 動作は、 $rayZmax = (dQ.z * 0.5 + Q.z) / (dK * 0.5 + k)$ である。上記と同様に、この例において、入力ユニット 40 は、 $(P, Q.z, k) += (dP, dQ.z, dK)$ の動作を行い得る。フォーマッティングユニット 44 は、 $P.x * stepDir \text{ endP}$ および $stepCount > maxSteps$ の条件が満たされるかどうかに基づいて、真値または偽値を割り当て得る。フィルタユニット 46 は、 $rayZmax = (dQ.z * 0.5 + Q.z) / (dK * 0.5 + k)$ の動作を行い得る。

20

【0089】

[0088] 出力ユニット 50 は、命令において定義された条件が満たされるかどうかに基づいて、複数の動作の実行の反復が必要とされるかどうかを決定し得る。真である場合、出力ユニット 50 は、さらなる実行反復のために、入力ユニット 40 にフィードバック信号 52 を出力する。偽である場合、出力ユニット 50 は、シェーダプロセッサ 36 に、テクスチャユニット 34 による繰り返される実行から生じる最終データを出力する。

【0090】

[0089] 本開示で説明される例示的な技法は、ツリートラバーサルアルゴリズムにも適用可能であり得る。例えば、光線追跡のためのバウンディングボリューム階層 (BVH : bounding volume hierarchy) ツリートラバーサルにおいて、テクスチャユニット 34 は、レイ - ボックス交差テストのために、およびループにおける複数の動作の多くの実行反復を使用してツリーをトラバースするように構成され得る。クアッドツリートラバーサル（例えば、スクリーン空間光線追跡、被写界深度、ボリュームレンダリング、ビュー合成など）の場合、開発者は、深度バッファの上にクアッドツリーを構築し得る。ツリーをトラバースするために、テクスチャユニット 34 は、レイ - ボックス交差テストおよびレイ - 平面交差テストに加えて複数のループ動作を実行するように構成され得る。レイ - 平面交差テストは、簡略化されたレイ - ボックス交差テストである。

30

【0091】

[0090] 図 4 は、本開示で説明される 1 つまたは複数の例示的な技法による、データを処理する例示的な方法を示すフローチャートである。テクスチャユニット 34 は、命令において定義された条件が満たされることに基づいて複数の動作を繰り返し実行することをテクスチャユニット 34 に命令する、GPU 12 のシェーダプロセッサ 36 からの命令を受け取る（54）。これら動作は、シェーダプログラムの複数の動作と、テクスチャ座標を変更するための複数の動作とであり得る。これら動作の例は、POM、SSRT、DOF 处理、ボリュームレンダリング、あるいは動的高さフィールドを用いた水または地形レンダリングを含む。

40

【0092】

[0091] テクスチャユニット 34 は、命令において定義された条件が満たされることま

50

たは満たされないことに基づいて、これら動作を繰り返し実行する（56）。例えば、テクスチャユニット34は、条件が満たされるまで複数の動作を繰り返し実行する（例えば、条件が満たされない場合、繰り返し実行する）、または条件が満たされる限り複数の動作を繰り返し実行する（例えば、条件が満たされる場合、繰り返し実行する）。また、テクスチャユニット34は、複数の動作を実行するために何も追加命令を受け取らずに、命令において定義された条件が満たされることまたは満たされることに基づいて、複数の動作を実行する。

【0093】

[0092] テクスチャユニット34は、シェーダプロセッサ36にこれら動作の繰り返される実行から生じるデータを出力する（58）。一例において、テクスチャユニット34は、これら動作の繰り返される実行の全ての反復が完了した後にのみ、データをシェーダプロセッサ36に出力する。

10

【0094】

[0093] いくつかの例において、テクスチャユニット34の出力ユニット50は、命令において定義された条件が満たされるのか満たされないのかに基づいて、これら動作の実行の反復が必要とされるかどうかを決定するように構成され得る。出力ユニット50は、これら動作の実行の反復が必要とされるという決定に基づいて、テクスチャユニット34から、入力ユニット40にフィードバック信号52を出力するように構成され得る。場合によっては、出力ユニット50は、これら動作の実行の反復が必要とされないという決定に基づいて、これら動作の繰り返される実行から生じるデータを出力するように構成され得る。いくつかの例において、さらなる反復を実行すべきかどうかを決定する際に、出力ユニット50は、読み取られたテクセル値またはテクセル値に基づく値を、命令において定義された変数と比較し得る。

20

【0095】

[0094] 本開示で説明された技法は、少なくとも部分的に、ハードウェア、ソフトウェア、ファームウェアまたはそれらの任意の組合せで実施され得る。例えば、説明された技法の様々な態様は、1つまたは複数のマイクロプロセッサ、デジタル信号プロセッサ（D S P）、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）、あるいは他の等価な集積回路またはディスクリート論理回路、並びにそのような構成要素の任意の組合せを含む、1つまたは複数のプロセッサ内で実施され得る。「プロセッサ」または「処理回路」という用語は、概して、単独のまたは他の論理回路と組み合わせられた上記の論理回路のいずれか、あるいは処理を行う個別ハードウェアのような任意の他の等価な回路を指し得る。

30

【0096】

[0095] そのようなハードウェア、ソフトウェア、およびファームウェアは、本開示で説明された複数の様々な動作および機能をサポートするために、同じデバイス内にまたは別個のデバイス内で実施され得る。さらに、説明されたユニット、モジュールまたは構成要素のいずれも、一緒に、または個別であるが相互動作可能な論理デバイスとして別々に実施され得る。モジュールまたはユニットとしての異なる特徴の記述は、異なる機能的態様を強調するものであり、そのようなモジュールまたはユニットが、別個のハードウェアまたはソフトウェア構成要素によって実現されなければならないことを必ずしも暗示するとは限らない。むしろ、1つまたは複数のモジュールまたはユニットに関連する機能が、別個のハードウェア、ファームウェア、および／またはソフトウェア構成要素によって行われるか、あるいは共通または別個のハードウェアまたはソフトウェア構成要素内に組み込まれ得る。

40

【0097】

[0096] 本開示で説明された技法はまた、複数の命令を記憶するコンピュータ可読記憶媒体など、コンピュータ可読媒体内に記憶、具現または符号化され得る。コンピュータ可読媒体内に埋め込まれたまたは符号化された複数の命令は、例えば、これら命令が1つまたは複数のプロセッサによって実行されたとき、1つまたは複数のプロセッサに本明細書

50

で説明された技法を行わせ得る。コンピュータ可読記憶媒体は、ランダムアクセスメモリ（RAM）、読み取り専用メモリ（ROM）、プログラマブル読み取り専用メモリ（PROM）、消去可能プログラマブル読み取り専用メモリ（EPROM）、電子的に消去可能なプログラマブル読み取り専用メモリ（EEPROM）、フラッシュメモリ、ハードディスク、CD-ROM、フロッピー（登録商標）ディスク、カセット、磁気媒体、光媒体、または有形である他のコンピュータ可読記憶媒体を含み得る。

【0098】

[0097] 様々な態様および例が説明された。しかしながら、以下の請求項の範囲から逸脱することなく、本開示の構造または技法に変更が行われ得る。

【図1】

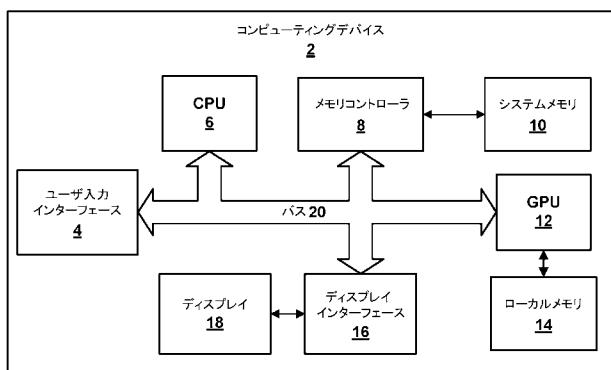


FIG. 1

【図2】

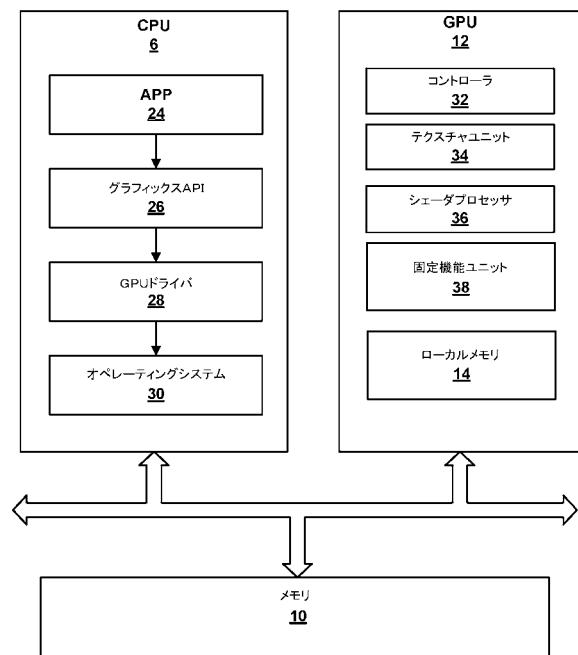
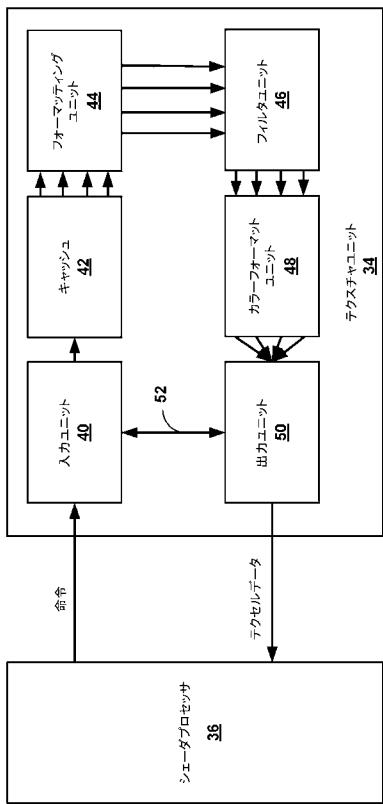


FIG. 2

【図3】



【図4】

FIG. 3

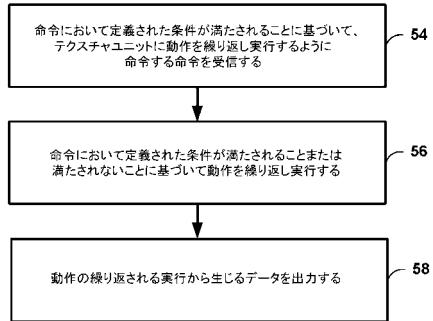


FIG. 4

【国際調査報告】

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2017/012734

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06T15/00
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	US 2015/379676 A1 (GIRADO JAVIER IGNACIO [US] ET AL) 31 December 2015 (2015-12-31) paragraph [0065] -----	1-28
A	US 7 528 843 B1 (KILGARD MARK J [US] ET AL) 5 May 2009 (2009-05-05) abstract -----	1-28
A	US 6 980 209 B1 (DONHAM CHRISTOPHER D S [US] ET AL) 27 December 2005 (2005-12-27) abstract -----	1-28



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

Date of mailing of the international search report

10 March 2017

20/03/2017

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

dos Santos, Luís

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/US2017/012734

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2015379676 A1	31-12-2015	US 2015379676 A1 WO 2015200685 A1	31-12-2015 30-12-2015
US 7528843 B1	05-05-2009	NONE	
US 6980209 B1	27-12-2005	NONE	

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,R0,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,DJ,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,KH,KN,KP,KR,KW,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

(74)代理人 100184332

弁理士 中丸 慶洋

(72)発明者 セイラン、ウサメ

アメリカ合衆国、カリフォルニア州 92121-1714、サン・ディエゴ、モアハウス・ドライブ 5775

(72)発明者 ゴエル、ビネット

アメリカ合衆国、カリフォルニア州 92121-1714、サン・ディエゴ、モアハウス・ドライブ 5775

(72)発明者 オーベルト、ユライ

アメリカ合衆国、フロリダ州 32817、オーランド、ファブリー・サークル 2408

(72)発明者 リ、リヤン

アメリカ合衆国、カリフォルニア州 92121-1714、サン・ディエゴ、モアハウス・ドライブ 5775

F ターム(参考) 5B080 CA01 CA03 CA08 GA22