

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第6947352号
(P6947352)

(45) 発行日 令和3年10月13日(2021.10.13)

(24) 登録日 令和3年9月21日(2021.9.21)

(51) Int.Cl. F I
G06T 15/00 (2011.01) G O 6 T 15/00 5 0 1
G06T 19/00 (2011.01) G O 6 T 19/00 3 0 0 B

請求項の数 26 外国語出願 (全 31 頁)

(21) 出願番号	特願2017-169634 (P2017-169634)	(73) 特許権者	500395107 アーム・リミテッド
(22) 出願日	平成29年9月4日(2017.9.4)		グレート・ブリテン及び北部アイルランド
(65) 公開番号	特開2018-49603 (P2018-49603A)		連合王国、イングランド、シー・ビー1
(43) 公開日	平成30年3月29日(2018.3.29)		、9エヌ・ジェイ、ケンブリッジ、フルボ
審査請求日	令和2年8月19日(2020.8.19)		ーン・ロード 1 1 0
(31) 優先権主張番号	1615012.0	(74) 代理人	100107456 弁理士 池田 成人
(32) 優先日	平成28年9月5日(2016.9.5)	(74) 代理人	100162352 弁理士 酒巻 順一郎
(33) 優先権主張国・地域又は機関	英国 (GB)	(74) 代理人	100123995 弁理士 野田 雅一

最終頁に続く

(54) 【発明の名称】 グラフィックス処理システム、グラフィックスプロセッサ及び該システム又はプロセッサを動作させる方法

(57) 【特許請求の範囲】

【請求項1】

グラフィックス処理システムを動作させる方法であって、前記システムが、少なくとも初期処理ステージとさらなる処理ステージとを含むグラフィックス処理パイプラインを含み、前記方法が、

前記初期処理ステージにおいて、第1の解像度でシーンのデータを処理するステップと、

前記初期処理ステージにおいて、第2の解像度で前記シーンのデータを処理するステップと、

前記初期処理ステージにおいて前記第1の解像度及び前記第2の解像度で前記シーンのデータを処理した後、バーチャルリアリティユーザデバイスのユーザの少なくとも片目の現在の位置付けに関する注視追跡データを受信するステップと、

受信された前記注視追跡データから、前記シーンの少なくとも1つのサブ領域を特定するステップと、

前記さらなる処理ステージにおいて、前記第1の解像度で前記シーンのデータを処理するステップと、

前記さらなる処理ステージにおいて、前記シーンの特定された前記少なくとも1つのサブ領域に対応するデータのみを前記第2の解像度で処理するステップと、

前記さらなる処理ステージにおいて前記第1の解像度で処理された前記シーンの前記データと、前記さらなる処理ステージにおいて前記第2の解像度で処理された前記シーンの

10

20

特定された前記少なくとも1つのサブ領域に対応する前記データと、を結合することによって、前記シーンをレンダリングするステップと、を含む、方法。

【請求項2】

前記グラフィックス処理パイプラインが、前記初期処理ステージと前記さらなる処理ステージとの間に中間処理ステージを含み、前記方法が、

前記中間処理ステージにおいて、前記第1の解像度で前記シーンのデータを処理するステップと、

前記中間処理ステージにおいて、前記第2の解像度で前記シーンの特定された前記少なくとも1つのサブ領域に対応するデータのみを処理するステップと、を含む、

10

前記注視追跡データが、前記中間処理ステージにおいて前記第1の解像度及び前記第2の解像度で前記シーンのデータを処理する前に受信され、前記中間処理ステージにおいて前記第1の解像度及び前記第2の解像度で前記シーンのデータを処理する前の前記バーチャルリアリティユーザデバイスのユーザの少なくとも片目の現在位置に関する、請求項1に記載の方法。

【請求項3】

前記グラフィックス処理パイプラインが、前記初期処理ステージと前記さらなる処理ステージとの間に中間処理ステージを含み、前記方法が、

前記中間処理ステージにおいて、前記第1の解像度で前記シーンのデータを処理するステップと、

20

前記中間処理ステージにおいて、前記第2の解像度で前記シーンのデータを処理するステップと、を含む、

前記注視追跡データが、前記中間処理ステージにおいて前記第1の解像度及び前記第2の解像度で前記シーンのデータを処理した後に受信され、前記中間処理ステージにおいて前記第1の解像度及び前記第2の解像度で前記シーンのデータを処理した後の前記バーチャルリアリティユーザデバイスのユーザの少なくとも片目の現在の位置付けに関する、請求項1に記載の方法。

【請求項4】

30

前記さらなる処理ステージの処理が、前記中間処理ステージの処理よりも高い計算負荷を有する、請求項2又は3に記載の方法。

【請求項5】

前記中間処理ステージが、1つ又は複数のジオメトリ演算を含む、請求項2～4のいずれか一項に記載の方法。

【請求項6】

前記さらなる処理ステージの処理が、前記初期処理ステージの処理よりも高い計算負荷を有する、請求項1～5のいずれか一項に記載の方法。

【請求項7】

前記さらなる処理ステージが、1つ又は複数のピクセル処理演算を含む、請求項1～6のいずれか一項に記載の方法。

40

【請求項8】

前記初期処理ステージが、1つ又は複数のデータバッファリング操作を含む、請求項1～7のいずれか一項に記載の方法。

【請求項9】

前記グラフィックス処理パイプラインの各ステージにおいてシーンのデータを処理するステップが、前記グラフィックス処理パイプラインの各ステージにおいて前記シーンのデータのフレームを処理するサブステップを含む、請求項1～8のいずれか一項に記載の方法。

【請求項10】

50

前記第 1 の解像度が前記第 2 の解像度より低い、請求項 1 ~ 9 のいずれか一項に記載の方法。

【請求項 1 1】

前記第 1 の解像度と前記第 2 の解像度とが所定の関係を有する、請求項 1 ~ 1 0 のいずれか一項に記載の方法。

【請求項 1 2】

前記所定の関係が、2 のべき乗を含む、請求項 1 1 に記載の方法。

【請求項 1 3】

前記所定の関係は、前記第 1 の解像度が前記第 2 の解像度の 1 / 4 であることである、請求項 1 1 又は 1 2 に記載の方法。

10

【請求項 1 4】

受信された前記注視追跡データが、前記バーチャルリアリティユーザデバイスのユーザの少なくとも片目の中心窩の現在の固視点に関し、

前記シーンの特定された前記少なくとも 1 つのサブ領域が、前記バーチャルリアリティユーザデバイスのユーザの前記少なくとも片目の前記中心窩の前記現在の固視点を中心とするサブ領域に関する、請求項 1 ~ 1 3 のいずれか一項に記載の方法。

【請求項 1 5】

前記シーンのデータが、左目ビューに関連するデータと右目ビューに関連するデータとを含み、前記グラフィックス処理パイプラインの各ステージの前記データを処理するステップが、前記左目ビューに関連するデータを処理するサブステップと、前記右目ビューに関連するデータを処理するサブステップと、を含み、

20

受信された前記注視追跡データが、前記バーチャルリアリティユーザデバイスのユーザの左目の中心窩の現在の位置付けと、前記バーチャルリアリティユーザデバイスのユーザの右目の中心窩の現在の位置付けと、に関する、請求項 1 ~ 1 4 のいずれか一項に記載の方法。

【請求項 1 6】

前記シーンの特定された前記少なくとも 1 つのサブ領域が、前記バーチャルリアリティユーザデバイスの前記ユーザの前記左目の前記中心窩の前記現在の位置付けを中心とする左目サブ領域と、前記バーチャルリアリティユーザデバイスの前記ユーザの前記右目の前記中心窩の前記現在の位置付けを中心とする右目サブ領域と、に関し、

30

前記シーンをレンダリングするステップが、

前記さらなる処理ステージにおいて前記第 1 の解像度で処理された前記シーンの左目データと、前記さらなる処理ステージにおいて前記第 2 の解像度で処理された前記シーンの特定された前記左目サブ領域に対応するデータと、を結合するサブステップと、

前記さらなる処理ステージにおいて前記第 1 の解像度で処理された前記シーンの右目データと、前記さらなる処理ステージにおいて前記第 2 の解像度で処理された前記シーンの特定された前記右目サブ領域に対応するデータと、を結合するサブステップと、を含む、請求項 1 5 に記載の方法。

【請求項 1 7】

特定された前記左目サブ領域のうちの少なくとも一部が、特定された前記右目サブ領域のうちの少なくとも一部と異なる、請求項 1 6 に記載の方法。

40

【請求項 1 8】

前記初期処理ステージにおいて、第 3 の解像度で前記シーンのデータを処理するステップを含み、

前記注視追跡データが、前記初期処理ステージにおいて前記第 3 の解像度で前記シーンのデータを処理した後に受信され、前記初期処理ステージにおいて前記第 3 の解像度で前記シーンのデータを処理した後の前記バーチャルリアリティユーザデバイスのユーザの少なくとも片目の現在の位置付けに関し、前記方法が、

前記さらなる処理ステージにおいて、前記第 3 の解像度で前記シーンのデータを処理す

50

るステップと、

前記さらなる処理ステージにおいて、前記第3の解像度で前記シーンのさらなるサブ領域に対応するデータのみを処理するステップと、
を含み、

前記シーンをレンダリングするステップが、前記さらなる処理ステージにおいて前記第3の解像度で処理された前記シーンの前記データをさらに結合するサブステップを含む、請求項1～17のいずれか一項に記載の方法。

【請求項19】

受信された前記注視追跡データから前記シーンの前記さらなるサブ領域を特定するステップを含む、請求項18に記載の方法。

10

【請求項20】

前記シーンの特定された前記少なくとも1つのサブ領域に基づいて前記シーンの前記さらなるサブ領域を計算するステップを含む、請求項18に記載の方法。

【請求項21】

グラフィックスプロセッサを動作させる方法であって、前記グラフィックスプロセッサが、少なくとも初期処理ステージとさらなる処理ステージとを含むグラフィックス処理パイプラインに関連付けられ、前記方法が、

前記初期処理ステージにおいて、第1の解像度でシーンのデータを処理するステップと、

前記初期処理ステージにおいて、第2の解像度で前記シーンのデータを処理するステップと、

20

前記初期処理ステージにおいて前記第1の解像度及び前記第2の解像度で前記シーンのデータを処理した後、バーチャルリアリティユーザデバイスのユーザの少なくとも片目の現在の位置付けに関する注視追跡データを受信するステップと、

受信された前記注視追跡データから、前記シーンの少なくとも1つのサブ領域を特定するステップと、

前記さらなる処理ステージにおいて、前記第1の解像度で前記シーンのデータを処理するステップと、

前記さらなる処理ステージにおいて、前記第2の解像度で前記シーンの特定された前記少なくとも1つのサブ領域に対応するデータのみを処理するステップと、

30

前記さらなる処理ステージにおいて前記第1の解像度で処理された前記シーンの前記データと、前記さらなる処理ステージにおいて前記第2の解像度で処理された前記シーンの特定された前記少なくとも1つのサブ領域に対応する前記データと、を結合することによって、前記シーンをレンダリングするステップと、
を含む、方法。

【請求項22】

グラフィックス処理システムにおいて実行された場合に、請求項1～21のいずれか一項に記載の方法を実施するように構成されたソフトウェアコードを含む、コンピュータプログラム。

【請求項23】

40

グラフィックス処理パイプラインを含むグラフィックス処理システムであって、前記グラフィックス処理パイプラインが、少なくとも初期処理ステージとさらなる処理ステージとを含み、前記グラフィックス処理システムが、

前記初期処理ステージにおいて、第1の解像度でシーンのデータを処理し、

前記初期処理ステージにおいて、第2の解像度で前記シーンのデータを処理し、

前記初期処理ステージにおいて前記第1の解像度及び前記第2の解像度で前記シーンのデータを処理した後、バーチャルリアリティユーザデバイスのユーザの少なくとも片目の現在の位置付けに関する注視追跡データを受信し、

受信された前記注視追跡データから、前記シーンの少なくとも1つのサブ領域を特定し、

50

前記さらなる処理ステージにおいて、前記第 1 の解像度で前記シーンのデータを処理し

、
前記さらなる処理ステージにおいて、前記第 2 の解像度で前記シーンの特定された前記少なくとも 1 つのサブ領域に対応するデータのみを処理し、

前記さらなる処理ステージにおいて前記第 1 の解像度で処理された前記シーンの前記データと、前記さらなる処理ステージにおいて前記第 2 の解像度で処理された前記シーンの特定された前記少なくとも 1 つのサブ領域に対応する前記データと、を結合することによって、前記シーンをレンダリングする、
ように構成された、グラフィックス処理システム。

【請求項 2 4】

請求項 2 3 に記載のグラフィックス処理システムを備える、バーチャルリアリティユーザデバイス。

【請求項 2 5】

少なくとも初期処理ステージとさらなる処理ステージとを含むグラフィックス処理パイプラインに関連付けられたグラフィックスプロセッサであって、

前記初期処理ステージにおいて、第 1 の解像度でシーンのデータを処理し、

前記初期処理ステージにおいて、第 2 の解像度で前記シーンのデータを処理し、

前記初期処理ステージにおいて前記第 1 の解像度及び前記第 2 の解像度で前記シーンのデータを処理した後、バーチャルリアリティユーザデバイスのユーザの少なくとも片目の現在の位置付けに関する注視追跡データを受信し、

受信された前記注視追跡データから、前記シーンの少なくとも 1 つのサブ領域を特定し

、
前記さらなる処理ステージにおいて、前記第 1 の解像度で前記シーンのデータを処理し

、
前記さらなる処理ステージにおいて、前記第 2 の解像度で前記シーンの特定された前記少なくとも 1 つのサブ領域に対応するデータのみを処理し、

前記さらなる処理ステージにおいて前記第 1 の解像度で処理された前記シーンの前記データと、前記さらなる処理ステージにおいて前記第 2 の解像度で処理された前記シーンの特定された前記少なくとも 1 つのサブ領域に対応する前記データと、を結合することによって、前記シーンをレンダリングする、
ように構成された、グラフィックスプロセッサ。

【請求項 2 6】

請求項 2 5 に記載のグラフィックスプロセッサを備える、バーチャルリアリティユーザデバイス。

【発明の詳細な説明】

【技術分野】

【0001】

本開示は、グラフィックス処理システム及びグラフィックスプロセッサに関する。

【背景】

【0002】

中心窩レンダリング (foveated rendering) は、表示される画像の一部がより高い解像度でレンダリングされ、その画像の 1 つ又は複数の他の部分がより低い解像度でレンダリングされるレンダリング技法である。この理由は、ユーザが直接見ている画像の一部は、視覚的に受容可能となるようにより高い解像度でレンダリングされる必要があり得、一方で、ユーザが直接見ていない、画像の周辺領域は、依然として視覚的に受容可能に見えつつも、より低い解像度でレンダリングされてもよいためである。中心窩レンダリングは、画像全体を最高解像度でレンダリングするのではなく、画像の周辺領域をより低い解像度でレンダリングすることによって、グラフィックス処理ユニット (GPU) へのレンダリング負荷を削減するために使用され得る。

【0003】

10

20

30

40

50

中心窩レンダリングは、画像のより高解像度のバージョンがレンダリングされる1つ又は複数の固視点を特定することを含み得、1つ又は複数の固視点から離れた領域がより低い解像度でレンダリングされる。画像の最高解像度の領域の位置、すなわち、1つ又は複数の固視点が決定され得る、異なる方法が存在する。たとえば、頭部追跡システム又は視線追跡システムが、ユーザが画像内で見ている場所の特定を試みるために使用され得る。

【0004】

中心窩レンダリングの1つの用途は、バーチャルリアリティ（VR）ディスプレイ、たとえば、バーチャルリアリティヘッドマウントディスプレイ（VR HMD）に対して画像をレンダリングする場合である。高解像度VR HMDは、深刻な樽形歪みを特徴とするレンズを使用し得る。この効果は、目ごとにディスプレイの中心に向かってレンダリ
10
ングされる画像が拡大され、周辺エリアが全て縮小されるというものである。したがって、周辺領域は、中心の拡大される領域よりも低い品質で、ユーザにとって全体の視覚効果の著しい損失なしにレンダリングされ得る。

【0005】

しかしながら、たとえば、中心窩レンダリングを行う場合に、グラフィックスプロセッサ及びグラフィックス処理システムの動作に改善の余地が残っている。

【概要】

【0006】

本開示の第1の態様によれば、グラフィックス処理システムを動作させる方法であって、上記システムが、少なくとも初期処理ステージとさらなる処理ステージとを含むグラフィ
20
ックス処理パイプラインを含み、上記方法が、初期処理ステージにおいて、第1の解像度でシーンのデータを処理するステップと、初期処理ステージにおいて、第2の解像度でシーンのデータを処理するステップと、初期処理ステージにおいて第1の解像度及び第2の解像度でシーンのデータを処理した後、バーチャルリアリティユーザデバイスのユーザの少なくとも片目の現在の位置付けに関する注視追跡データを受信するステップと、受信された注視追跡データから、シーンの少なくとも1つのサブ領域を特定するステップと、さらなる処理ステージにおいて、第1の解像度でシーンのデータを処理するステップと、さらなる処理ステージにおいて、第2の解像度でシーンの特定された少なく
30
とも1つのサブ領域に対応するデータのみを処理するステップと、さらなる処理ステージにおいて第1の解像度で処理されたシーンのデータと、さらなる処理ステージにおいて第2の解像度で処理されたシーンの特定された少なくとも1つのサブ領域に対応するデータとを結合することによって、シーンをレンダリングするステップとを含む、方法が提供される。

【0007】

本開示の第2の態様によれば、グラフィックス処理パイプラインを含むグラフィックス処理システムを動作させる方法であって、グラフィックス処理パイプラインにおいてレイ
40
トラッチ（late-latched）の視線追跡データを用いて、シーンの1つ又は複数の領域であり、シーンの1つ又は複数の他の領域の解像度に比べて比較的高い解像度でレンダリングされるべき1つ又は複数の領域を特定するステップを含む、方法が提供される。

【0008】

本開示の第3の態様によれば、グラフィックス処理システムを動作させる方法であって、上記システムが、少なくとも初期処理ステージとさらなる処理ステージとを含むグラフィ
50
ックス処理パイプラインを含み、上記方法が、グラフィックス処理ワークアイテムについての入力データをグラフィックス処理パイプラインに第1の時間に提供するステップと、グラフィックス処理パイプラインがグラフィックス処理ワークアイテムを処理している間第2のより遅い時間に、バーチャルリアリティユーザデバイスのユーザの少なくとも片目

の現在の位置付けに関する注視追跡データをグラフィックス処理パイプラインに提供するステップと
を含む、方法が提供される。

【0009】

本開示の第4の態様によれば、グラフィックスプロセッサを動作させる方法であって、上記グラフィックスプロセッサが、少なくとも初期処理ステージとさらなる処理ステージとを含むグラフィックス処理パイプラインに関連付けられ、上記方法が、初期処理ステージにおいて、第1の解像度でシーンのデータを処理するステップと、初期処理ステージにおいて、第2の解像度でシーンのデータを処理するステップと、初期処理ステージにおいて第1の解像度及び第2の解像度でシーンのデータを処理した後、
10 パーチャルリアリティユーザデバイスのユーザの少なくとも片目の現在の位置付けに関する注視追跡データを受信するステップと、
受信された注視追跡データから、シーンの少なくとも1つのサブ領域を特定するステップと、
さらなる処理ステージにおいて、第1の解像度でシーンのデータを処理するステップと、さらなる処理ステージにおいて、第2の解像度でシーンの特定された少なくとも1つのサブ領域に対応するデータのみを処理するステップと、
さらなる処理ステージにおいて第1の解像度で処理されたシーンのデータと、さらなる処理ステージにおいて第2の解像度で処理されたシーンの特定された少なくとも1つのサブ領域に対応するデータとを結合することによって、シーンをレンダリングするステップと
20 を含む、方法が提供される。

【0010】

本開示の第5の態様によれば、グラフィックス処理システムにおいて実行された場合に、1つ又は複数のそのような方法を実施するように構成されたソフトウェアコードを含む、コンピュータプログラムが提供される。

【0011】

本開示の第6の態様によれば、グラフィックス処理パイプラインを含むグラフィックス処理システムであって、上記グラフィックス処理パイプラインが、少なくとも初期処理ステージとさらなる処理ステージとを含み、上記グラフィックス処理システムが、初期処理ステージにおいて、第1の解像度でシーンのデータを処理し、
30 初期処理ステージにおいて、第2の解像度でシーンのデータを処理し、
初期処理ステージにおいて第1の解像度及び第2の解像度でシーンのデータを処理した後、
パーチャルリアリティユーザデバイスのユーザの少なくとも片目の現在の位置付けに関する注視追跡データを受信し、
受信された注視追跡データから、シーンの少なくとも1つのサブ領域を特定し、
さらなる処理ステージにおいて、第1の解像度でシーンのデータを処理し、
さらなる処理ステージにおいて、第2の解像度でシーンの特定された少なくとも1つのサブ領域に対応するデータのみを処理し、
さらなる処理ステージにおいて第1の解像度で処理されたシーンのデータと、さらなる処理ステージにおいて第2の解像度で処理されたシーンの特定された少なくとも1つのサブ領域に対応するデータとを結合することによって、シーンをレンダリングする
40 ように構成された、グラフィックス処理システムが提供される。

【0012】

本開示の第7の態様によれば、グラフィックス処理パイプラインを含むグラフィックス処理システムであって、グラフィックス処理パイプラインにおいてレイトラッチの視線追跡データを用いて、シーンの1つ又は複数の領域であり、シーンの1つ又は複数の他の領域の解像度に比べて比較的高い解像度でレンダリングされるべき1つ又は複数の領域を特定するように構成された、グラフィックス処理システムが提供される。

【0013】

本開示の第8の態様によれば、少なくとも初期処理ステージとさらなる処理ステージと

10

20

30

40

50

を含むグラフィックス処理パイプラインを含むグラフィックス処理システムであって、グラフィックス処理ワークアイテムについての入力データをグラフィックス処理パイプラインに第1の時間に提供し、グラフィックス処理パイプラインがグラフィックス処理ワークアイテムを処理している間第2のより遅い時間に、バーチャルリアリティユーザデバイスのユーザの少なくとも片目の現在の位置付けに関する注視追跡データをグラフィックス処理パイプラインに提供するように構成された、グラフィックス処理システムが提供される。

【0014】

本開示の第9の態様によれば、1つ又は複数のそのようなグラフィックス処理システムを備える、バーチャルリアリティユーザデバイスが提供される。

10

【0015】

本開示の第10の態様によれば、少なくとも初期処理ステージとさらなる処理ステージとを含むグラフィックス処理パイプラインに関連付けられたグラフィックスプロセッサであって、

初期処理ステージにおいて、第1の解像度でシーンのデータを処理し、

初期処理ステージにおいて、第2の解像度でシーンのデータを処理し、

初期処理ステージにおいて第1の解像度及び第2の解像度でシーンのデータを処理した後、バーチャルリアリティユーザデバイスのユーザの少なくとも片目の現在の位置付けに関する注視追跡データを受信し、

受信された注視追跡データから、シーンの少なくとも1つのサブ領域を特定し、

20

さらなる処理ステージにおいて、第1の解像度でシーンのデータを処理し、

さらなる処理ステージにおいて、第2の解像度でシーンの特定された少なくとも1つのサブ領域に対応するデータのみを処理し、

さらなる処理ステージにおいて第1の解像度で処理されたシーンのデータと、さらなる処理ステージにおいて第2の解像度で処理されたシーンの特定された少なくとも1つのサブ領域に対応するデータとを結合することによって、シーンをレンダリングするように構成された、グラフィックスプロセッサが提供される。

【0016】

本開示の第11の態様によれば、そのようなグラフィックスプロセッサを備える、バーチャルリアリティユーザデバイスが提供される。

30

【0017】

さらなる特徴及び利点は、添付の図面を参照してなされる、単なる例として与えられる以下の説明から明らかとなる。

【図面の簡単な説明】

【0018】

【図1】本開示の一実施形態によるグラフィックス処理システムの一例の概略図である。

【図2】本開示の一実施形態によるグラフィックス処理パイプラインの一例の概略図である。

【図3】本開示の一実施形態によるグラフィックス処理パイプラインの他の例の概略図である。

40

【図4】本開示の一実施形態によるフレームパイプライン化の一例の概略図である。

【図5】本開示の一実施形態によるグラフィックス処理パイプラインの他の例の概略図である。

【図6】本開示の一実施形態によるグラフィックス処理パイプラインの他の例の概略図である。

【図7】本開示の一実施形態によるグラフィックス処理パイプラインの他の例の概略図である。

【図8】グラフィックス処理パイプラインの他の例の概略図である。

【詳細な説明】

【0019】

50

本開示の実施形態によるいくつかの例が、表示用のコンピュータグラフィックスの処理の文脈においてここで説明される。

【0020】

図1を参照すると、グラフィックス処理システム100の一例が概略的に示されている。

【0021】

この例では、ホストプロセッサ110上で実行されるゲームなどのアプリケーション105は、関連付けられたGPU115によってグラフィックス処理演算が実行されるように要求する。これを行うために、アプリケーション105は、GPU115のドライバ120によって解釈されるアプリケーションプログラミングインターフェース(API)コールを生成する。ドライバ120はホストプロセッサ110上で動作する。ドライバ120は、アプリケーション105によって要求されたグラフィックス出力を生成するためのGPU115への適切なコマンドを生成する。アプリケーション105からのコマンドに応答して、コマンドのセットがGPU115に提供される。コマンドは、ディスプレイに表示されるフレームを生成するためのものであり得る。

10

【0022】

図2を参照すると、グラフィックス処理パイプライン200の一例が概略的に示されている。グラフィックス処理パイプライン200は、GPUによって実行され得るアクションのシーケンスを指す。

【0023】

この例では、GPUはタイルベースのレンダラである。したがって、GPUは、生成されるべきレンダー出力データアレイのタイルを生成する。レンダー出力データアレイは、出力フレームであり得る。タイルベースのレンダリングは、レンダー出力全体が一度に処理されるのではなく、レンダー出力が複数のより小さいサブ領域(又は「エリア」)に分割されるという点で、即時モードレンダリングと異なる。それらのサブ領域は、本明細書ではタイルと呼ばれる。各タイルは別々にレンダリングされる。たとえば、各タイルは次々にレンダリングされ得る。次いで、レンダリングされたタイルは再結合されて、表示用の完全なレンダー出力を提供する。タイルベースのレンダリングでは、レンダー出力は、規則的な大きさ及び形状のタイルに分割され得る。タイルは四角形又は他の形状であり得る。

20

30

【0024】

「タイル化」及び「タイルベースの」レンダリングに使用され得る他の用語は、レンダリングタイルが「チャンク」と呼ばれる「チャンキング」、及び「バケット」レンダリングを含む。「タイル」及び「タイル化」という用語が以降便宜上使用されるが、これらの用語が全ての代替的及び等価的な用語及び技法を包含するよう意図されていることは理解されたい。

【0025】

レンダー出力データアレイは、スクリーン又はプリンタなどの表示デバイスに表示するための出力フレームであり得る。レンダー出力はまた、たとえば、後のレンダリングパスで使用するための中間データを含むことができる。この一例は、「テクスチャへのレンダリング(render to texture)」出力である。

40

【0026】

コンピュータグラフィックス画像が表示される場合、最初にジオメトリのセットとして、たとえばプリミティブのシリーズとして定義され得る。プリミティブの一例はポリゴンである。次いで、ジオメトリはラスタライズ処理においてグラフィックスフラグメントに分割される。これにグラフィックスレンダリングが続く。グラフィックスレンダリング演算の間に、レンダラは各フラグメントに関連するデータを修正して、フラグメントが正しく表示できるようにすることができる。そのようなデータの例は、色及び透明度を含む。フラグメントがレンダラを完全に横断すると、それらに関連するデータ値がメモリに記憶され、出力の準備ができる。

50

【 0 0 2 7 】

図 2 は、本明細書に記載の実施形態の動作に関する、GPUに関連付けられる様々な要素及びパイプラインステージを示す。しかしながら、図 2 に示されていないグラフィックス処理パイプラインの他の要素及びステージがあり得る。図 2 が単に概略的であること、また、たとえば、実際には、図示された機能ユニット及びパイプラインステージが、図 2 では別々のステージとして概略的に示されていても、重要なハードウェア回路を共有できることにも留意されたい。グラフィックス処理パイプライン 200 のステージ、要素及びユニットなどのそれぞれが、望み通りに実装され得、したがって、たとえば、関連付けられた動作及び機能を実行するための適切な回路及び/又は処理ロジックなどを含むことも理解されよう。

10

【 0 0 2 8 】

図 2 に示されるように、GPUが実行するグラフィックス処理パイプライン 220 は、いくつかのステージ、たとえば、頂点シェーダ 205、ハルシェーダ 210、テッセレータ 215、ドメインシェーダ 220、ジオメトリシェーダ 225、タイラ 230、ラスタライズステージ 235、早期 (early: アーリー) Z (又は「深度 (depth: デプス) 」) 及びステンシルテストステージ 240、フラグメントシェーディングステージ 245 の形態のレンダラ、後期 (late: レイト) Z (又は「深度」) 及びステンシルテストステージ 250、ブレンドステージ 255、タイルバッファ 260、並びにダウンサンプリング及び書き出しステージ 265 を含む。しかしながら、GPUの他の構成が可能である。

20

【 0 0 2 9 】

頂点シェーダ 205 は、生成されるべき出力のために定義された頂点に関連する入力データ値を受信する。頂点シェーダ 205 は、それらのデータ値を処理して、グラフィックス処理パイプライン 200 の後続のステージによって使用するための、対応する頂点シェーディングされた出力データ値のセットを生成する。

【 0 0 3 0 】

処理されるべき各プリミティブは、頂点のセットによって定義され表現され得る。プリミティブの各頂点は、属性のセットが関連付けられ得る。属性のセットは、頂点についてのデータ値のセットである。これらの属性は、位置データ及び他の非位置データ (又は「varying」) を含むことができる。非位置データは、たとえば、問題の頂点についての色、光、法線及び/又はテクスチャ座標を定義することができる。

30

【 0 0 3 1 】

頂点のセットが、GPUによって出力される所与の出力について定義される。出力のために処理されるべきプリミティブは、頂点のセットにおける所与の頂点を含む。頂点シェーディング演算は、各頂点の属性を、後続のグラフィックス処理演算のために所望の形態に変換する。これは、たとえば、頂点位置属性を、それらが最初に定義されたワールド又はユーザ空間から、グラフィックス処理システムの出力が表示されるスクリーン空間に変換することを含み得る。これは、たとえば、レンダリングされるべき画像内のライティングの効果を考慮するように入力データを修正することも含み得る。

40

【 0 0 3 2 】

ハルシェーダ 210 は、パッチ制御点のセットに演算を行い、パッチ定数として知られている追加データを生成する。

【 0 0 3 3 】

テッセレーションステージ 215 は、ジオメトリを細分化して、ハルの高次表現を作成する。

【 0 0 3 4 】

ドメインシェーダ 220 は、頂点シェーダ 205 と同様の方法で、テッセレーションステージによって出力された頂点に演算を行う。

【 0 0 3 5 】

ジオメトリシェーダ 225 は、三角形、点又は線などのプリミティブ全体を処理する。

50

【 0 0 3 6 】

頂点シェーダ 2 0 5、ハルシェーダ 2 1 0、テッセレータ 2 1 5、ドメインシェーダ 2 2 0 及びジオメトリシェーダ 2 2 5 は、GPU に提供されたコマンド及び頂点データに
10 応答して、フラグメントフロントエンド演算、たとえば、変換及びライティング演算、並び
にレンダリングされるべきプリミティブをセットアップするためのプリミティブセットア
ップを行う。

【 0 0 3 7 】

レンダリングされるべき全てのプリミティブが適切にセットアップされると、タイラ 2
3 0 は、レンダラ出力が処理用に分割されたタイルごとに処理されるべきプリミティブを
10 決定する。これを行うために、タイラ 2 3 0 は、処理されるべき各プリミティブの位置を
タイル位置と比較し、プリミティブが入る可能性があるとして決定したタイルごとに、プリミ
ティブをそれぞれのプリミティブリストに追加する。プリミティブをソートし、タイルリス
トにビニングするための任意の適切な所望の技法、たとえば正確なビニング、若しくは
バウンディングボックスビニング、又はその間にあるものが、タイル化処理に使用され得
る。

【 0 0 3 8 】

レンダリングされるべきプリミティブのリスト（又は「プリミティブリスト」）がこの
ようにレンダリングタイルごとに準備されると、プリミティブリストが使用のために記憶
20 される。プリミティブリストによって、システムは、問題のタイルがレンダリングされる
場合に、考慮されレンダリングされるべきプリミティブを特定することが可能になる。

【 0 0 3 9 】

タイラ 2 3 0 がタイルリストの全てを準備すると、各タイルがレンダリングされ得る。
これを行うために、タイラ 2 3 0 に続くグラフィックス処理パイプラインステージによっ
て各タイルが処理される。

【 0 0 4 0 】

所与のタイルが処理される場合、そのタイルについて処理されるべき各プリミティブが
ラスタライザ 2 3 5 に渡される。グラフィックス処理パイプライン 2 0 0 のラスタライズ
30 ステージ 2 3 5 は、プリミティブを処理用に個々のグラフィックスフラグメントにラスタ
ライズするように動作する。これを行うために、ラスタライザ 2 3 5 は、プリミティブを
サンプリング点にラスタライズし、プリミティブをレンダリングするための適切な位置を
有するグラフィックスフラグメントを生成する。ラスタライザ 2 3 5 によって生成された
フラグメントは次いでその先へ、パイプライン 2 0 0 の残りへ送信されて処理される。

【 0 0 4 1 】

早期 Z 及びステンシルテストステージ 2 4 0 は、ラスタライザ 2 3 5 から受信したフラ
グメントに Z（又は「深度」）テストを行って、任意のフラグメントがこのステージで破
棄（又は「淘汰（cull ed）」）され得るかを確認する。これを行うために、早期 Z
40 及びステンシルテストステージ 2 4 0 は、ラスタライザ 2 3 5 によって発行されたフラグ
メントの深度値を、レンダリング済みのフラグメントの深度値と比較する。レンダリング
済みのフラグメントの深度値は、タイルバッファ 2 6 0 の一部である深度バッファに記憶
される。早期 Z 及びステンシルテストステージ 2 4 0 によって行われる比較は、新たなフ
ラグメントがレンダリング済みのフラグメントによって閉塞されるか否かを判定するた
めのものである。同時に、早期ステンシルテストが実施される。フラグメント早期 Z 及び
50 ステンシルテストステージ 2 4 0 を通過したフラグメントは、フラグメントシェーディング
ステージ 2 4 5 に送信される。フラグメントシェーディングステージ 2 4 5 は、早期 Z 及
びステンシルテストを通過したフラグメントに適切なフラグメント処理演算を行って、適
切なレンダリングされたフラグメントデータを生成する。このフラグメント処理は、任意
の適切なフラグメントシェーディング処理、たとえば、フラグメントシェーダプログラム
をフラグメントに実行して適切なフラグメントデータを生成すること、テクスチャをフラ
グメントに適用すること、フォギング又は他の演算をフラグメントに適用することなど
を含むことができる。フラグメントシェーディングステージ 2 4 5 は、プログラマブルフラ

グメントシェーダであり得る。

【 0 0 4 2 】

次いで、後期フラグメント Z 及びステンシルテストステージ 2 5 0 があり、これはとりわけ、パイプライン深度テストの残りをシェーディングされたフラグメントに実施して、レンダリングされたフラグメントが実際に最終画像で見られるかを判定するものである。この深度テストは、タイルバッファ 2 6 0 内の Z バッファに記憶されたフラグメントの位置についての Z バッファ値を用いて、新たなフラグメントのフラグメントデータがレンダリング済みのフラグメントのフラグメントデータを置き換えるべきであることを判定する。これは、フラグメントシェーディングステージ 2 4 5 によって発行されたフラグメントの深度値を、Z バッファに記憶されたレンダリング済みのフラグメントの深度値と比較することを必要とし得る。この後期フラグメント深度及びステンシルテストステージ 2 5 0 は、後期アルファ及びノ又はステンシルテストをフラグメントに実施することもできる。

10

【 0 0 4 3 】

後期フラグメントテストステージ 2 5 0 を通過したフラグメントは次いで、ブレンダ 2 5 5 において、タイルバッファ 2 6 0 に記憶済みのフラグメントとの任意のブレンド演算が施され得る。フラグメントに必要な任意の他の残りの演算、たとえばディザなど（図示せず）も、このステージで実施される。

【 0 0 4 4 】

最後に、出力フラグメントデータ（又は「値」）がタイルバッファ 2 6 0 に書き込まれる。出力フラグメントデータは次いで、表示用にフレームバッファ 2 7 0 に出力され得る。出力フラグメントの深度値も、タイルバッファ 2 6 0 内の Z バッファに適切に書き込まれる。タイルバッファ 2 6 0 は、色バッファ及び深度バッファを記憶し、これらは、バッファが表すサンプリング点ごとに、それぞれ適切な色など、又は Z 値を記憶するものである。これらのバッファは、レンダラ出力全体の一部、この例ではタイルを表すフラグメントデータのアレイを記憶し、バッファ内のそれぞれのサンプル値のセットは、レンダラ出力全体のそれぞれのピクセルに対応する。たとえば、2 × 2 のサンプル値の各セットが出力ピクセルに対応し得、ここで 4 倍のマルチサンプリングが使用される。

20

【 0 0 4 5 】

タイルバッファ 2 6 0 は、グラフィックス処理パイプライン 2 0 0 にとってローカルであるランダムアクセスメモリ（RAM）の一部として設けられる。換言すれば、タイルバッファ 2 6 0 はオンチップメモリに設けられる。

30

【 0 0 4 6 】

タイルバッファ 2 6 0 からのデータは、ダウンサンプリング書き出しユニット 2 6 5 に入力され、次いで、外部メモリ出力バッファ、たとえば表示デバイス（図示せず）のフレームバッファ 2 7 0 に出力される。表示デバイスは、たとえば、ピクセルのアレイを備えるディスプレイ、たとえばコンピュータモニタ又はプリンタを具備することができる。

【 0 0 4 7 】

ダウンサンプリング及び書き出しユニット 2 6 5 は、バッファ 2 6 0 に記憶されたフラグメントデータを、出力バッファ及びデバイスに適切な解像度にダウンサンプリングして、出力デバイスのピクセルに対応するピクセルデータのアレイが生成されるようにする。この結果、出力値が、出力バッファ 2 7 0 への出力用のピクセルの形式となる。

40

【 0 0 4 8 】

レンダラ出力のタイルが処理され、そのデータが記憶用のメインメモリへ、たとえば、メインメモリ内のフレームバッファ 2 7 0 へエクスポートされると、次のタイルが処理され、同様に続き、これはレンダラ出力全体を生成するのに十分なタイルが処理されるまで行われる。次いで、この処理が次のレンダラ出力について反復され、同様に続く。

【 0 0 4 9 】

図 2 からわかるように、グラフィックス処理パイプライン 2 0 0 は、いくつかのプログラマブル処理又は「シェーダ」ステージ、すなわち、頂点シェーダ 2 0 5、ハルシェーダ 2 1 0、ドメインシェーダ 2 2 0、ジオメトリシェーダ 2 2 5、及びフラグメントシェー

50

ダ 2 4 5 を含む。これらのプログラマブルシェーダステージは、1 つ又は複数の入力変数を有し、出力変数のセットを生成するそれぞれのシェーダプログラムを実行する。問題のシェーダプログラムは、処理されるべきワークアイテムごとに、たとえば、頂点シェーダ 2 0 5 の場合は頂点ごとに実行され得る。実行スレッドは、処理されるべきワークアイテムごとに発行され得、スレッドは次いで、シェーダプログラム内の命令を実行して、所望のシェーディングされた出力データを生成する。

【 0 0 5 0 】

図 1 に関連して上述したアプリケーション 1 0 5 などのアプリケーションは、高水準のシェーダプログラミング言語、たとえば、OpenGL (登録商標) シェーディング言語 (GLSL)、高水準のシェーディング言語 (HLSL)、オープンコンピューティング言語 (OpenCL) などを用いて、実行されるべきシェーダプログラムを提供する。これらのシェーダプログラムは次いでシェーダ言語コンパイラによって、対象のグラフィックス処理パイプライン 2 0 0 用のバイナリコードに翻訳される。これは、コンパイラにおいてプログラムの 1 つ又は複数の内部の中間表現を作成することを含むことができる。コンパイラは、たとえばドライバ 1 2 0 の一部であり得、コンパイラを動作させる特別な API コールが存在する。したがって、コンパイラの実行は、ドライバによって、アプリケーション 1 0 5 により生成された API コールに応答してなされる描画コールの準備の一部と見ることができる。

【 0 0 5 1 】

ここで中心窩レンダリングに戻ると、視線追跡としても知られている注視追跡は、中心窩レンダリングを含む VR のいくつかのエリアに影響を与える可能性が高い。

【 0 0 5 2 】

人間の目はその受容器のほとんどを、鮮明な中心視力を担う目のエリアである中心窩に有する。中心窩は全視界に比べて小さい。ニアアイディスプレイ、たとえば VR システムで使用されるものでは、目はいかなる瞬間にもディスプレイ上の情報のサブセットしか知覚することができない。注視追跡型の中心窩レンダリングは、中心窩が見ることができる表示エリアにおいて完全な計算を維持するようにレンダリングを構成し、その他の場所での計算を削減する。

【 0 0 5 3 】

図 3 を参照すると、グラフィックス処理パイプライン 3 0 0 の一例が概略的に示されている。グラフィックス処理パイプライン 3 0 0 は、グラフィックス処理システム内に構成される。グラフィックス処理パイプライン 3 0 0 は、初期処理ステージ 3 0 5 と、さらなる処理ステージ 3 1 0 とを少なくとも含む。グラフィックス処理パイプライン 3 0 0 は、注視追跡型の中心窩レンダリングを行うために使用され得る。

【 0 0 5 4 】

注視追跡データ 3 1 5 は、初期処理ステージ 3 0 5 又はさらなる処理ステージ 3 1 0 におけるあらゆるデータ処理の前に受信される。注視追跡データ 3 1 5 は、バーチャルリアリティユーザデバイスのユーザの少なくとも片目の位置付けに関する。シーンの少なくとも 1 つのサブ領域は、受信された注視追跡データから特定される。シーンの特定された少なくとも 1 つのサブ領域は、たとえば、ユーザが見ているシーンの 1 つ又は複数のエリアに対応し得る。第 1 の解像度のシーンのデータは、初期処理ステージ 3 0 5 及び / 又はさらなる処理ステージ 3 1 0 において処理される。しかしながら、第 2 の解像度のシーンの特定された少なくとも 1 つのサブ領域に対応するデータのみが、初期処理ステージ 3 0 5 及び / 又はさらなる処理ステージ 3 1 0 において処理される。第 1 の解像度は、第 2 の解像度に比べて比較的高いものであり得る。次いで、シーンは、さらなる処理ステージ 3 1 0 により出力されたデータを結合することによって、レンダリングされ得る。したがって、注視追跡型の中心窩レンダリングは、第 1 の解像度が第 2 の解像度より低い場合に提供され得、その理由は、注視追跡データ 3 1 5 がグラフィックスパイプライン 3 0 0 内の処理に影響を与えるために使用されるためである。

【 0 0 5 5 】

そのような例は注視追跡型の中心窩レンダリングを提供するが、注視追跡型の中心窩レンダリングの成功は様々な要因に依存する。1つのそのような要因は、視線追跡データ315の品質である。他のそのような要因は、視線追跡器デバイスにより取得された視線追跡情報がレンダリングシステムに渡され使用される応答レイテンシである。レイテンシによって結果が古くなったために、及び/又は低品質のトラッキングデータによって、視線追跡データの精度が低いほど、中心窩レンダリングの結果はより成功しなくなる。

【0056】

レイテンシの観点から、いくつかの例では、GPUは、たとえば、ジオメトリが最初のフレームで提出されるか、又はジオメトリが次のフレームの開始まで遅延するかに応じて、2フレーム又は3フレームのパイプラインを有する。GPUは通常は遅延モードで動作し得、その場合、パイプラインは3フレームとなる。

10

【0057】

図4を参照すると、フレームパイプライン化400の一例が概略的に示されている。この例では、GPUは、中央処理装置(CPU)バッファリング405がフレーム0で行われ、ジオメトリフェーズ410がフレーム1で行われ、ピクセルフェーズ415がフレーム2で行われる3フレームパイプラインを有する。60Hzの表示レートにおいて、これは、新たなデータがグラフィックスAPIを介してグラフィックスパイプラインに提供される時刻と、結果がディスプレイなどに出力される準備ができるまでとの間に50msのレイテンシを与える。真のレイテンシは、画像センサのキャプチャ、処理及び転送を含むので、これよりも非常に大きくなり得る。

20

【0058】

VRシステムに対して20msの運動-光子(motion-to-photons)レイテンシの目標知覚閾値が与えられた場合、50msを超えるレイテンシは、ユーザー体験が懸念される限り、長すぎる可能性がある。

【0059】

再び中心窩レンダリングに戻ると、VRの文脈におけるコンテンツのレンダリングは、シーンを複数のビューに、目ごとに少なくとも1つのビューを描画することを必要とする。OVR_multiviewの拡張ファミリ(以下、「マルチビュー」)は、組み込みシステム用OpenGL(登録商標)(OpenGL(登録商標)ES)において、シーンの複数のビューを生成するために使用され得る。原理的には、マルチビューを用いて、4つのビュー、すなわち、高解像度の左目ビュー、高解像度の右目ビュー、低解像度の左目ビュー、及び低解像度の右目ビューが使用される中心窩レンダリングを処理することができる。たとえば、マルチビューを用いて、4つのビューを同時に、すなわち、目ごとに内側及び外側ビューをレンダリングすることができる。

30

【0060】

しかしながら、マルチビューの現在のバージョンにおける全てのビューは、互いと同一の解像度を有さなければならない。これは、たとえば非注視追跡型の実装に関して、内側及び外側両方の領域について単一の値が発見され得る場合に、許容可能であり得る。しかしながら、マルチビューの現在のバージョンは、注視追跡型の実装で使用するためのしっかりと焦点を合わせた中心窩領域の使用を許可しない。

40

【0061】

さらに、マルチビューの現在のバージョンは、4つのビューまでしか許可しない。新たなマルチビュー拡張で使用され得るビューの数を拡張することが可能であり得るが、追加のビューも単一解像度のルールの対象となる可能性が高い。

【0062】

さらに、焦点位置に関連する構成を含む各ビューの構成は、アプリケーションによってグラフィックスパイプラインの開始時に提供される。したがって、構成は、少なくともグラフィックスパイプラインと同じ長さのレイテンシを有する。したがって、上述の例では、視聴者の目の位置の追跡と、グラフィックス出力の関連付けとの間のレイテンシは、常に少なくとも50msとなる。そのようなレイテンシを念頭に入れて、グラフィックスパ

50

イブラインは、出力の高解像度領域の大きさを決定する際に保守的な手法を取り得る。たとえば、グラフィックスパイプラインは、グラフィックスパイプラインの開始時に視聴者の焦点位置を中心とする比較的大きい領域を使用することができ、これは視聴者が、レンダリングが完了したときにその比較的大きい領域内のどこかを見ている可能性が高く、ひいては出力の高解像度部分を見ている可能性が高いことを予想したものである。しかしながら、そのような保守的な手法は、出力が表示されているときに視聴者が見ている場所に関する不確実性のために、シーンの比較的大きい部分をより高解像度でレンダリングすることを必要とする。

【 0 0 6 3 】

マルチビューと互換性のある措置であって、マルチビューを拡張して、低いレイテンシで変化し得る中心窩領域へのレンダリングを可能にし、及び/又は中心窩領域の解像度がマルチビュー解像度と異なることを可能にする措置を提供することが望ましいであろう。

【 0 0 6 4 】

グラフィックスプロセッサを用いて、シーンの同一のビューを表すが異なる解像度である複数の画像をレンダリングすることによって、中心窩レンダリングを行う例がここで説明される。次いで、異なる解像度のビューは結合されて、解像度が画像にわたって変化する出力中心窩画像を提供する。これらの例では、上述のパイプラインレイテンシ、及び異なる解像度におけるレンダリングのオーバーヘッドもまた、削減され得る。

【 0 0 6 5 】

これらの例では、グラフィックスパイプラインの開始後に、たとえば、グラフィックスパイプラインの半ばに、視線追跡データが消費される。したがって、グラフィックスパイプラインの開始時に使用される視線追跡データに比べて、より最近の（すなわち、より直前に受信又は取得された）視線追跡データが使用され得る。レンダリングパイプラインにおける視線追跡データの消費のレイテンシを削減することによって、中心窩レンダリングの利益が増加し得る。さらに、本明細書に記載のようなそのような措置の実施は、既存のアーキテクチャと容易に統合することができ、非常に効率的な実装となる。

【 0 0 6 6 】

図5を参照すると、グラフィックス処理パイプライン500の一例が概略的に示されている。

【 0 0 6 7 】

グラフィックス処理パイプライン500は、グラフィックス処理システム内に構成される。グラフィックス処理パイプライン500は、初期処理ステージ505と、さらなる処理ステージ510とを少なくとも含む。さらなる処理ステージ510の処理は、初期処理ステージ505の処理よりも高い計算負荷を有し得る。

【 0 0 6 8 】

いくつかの例では、さらなる処理ステージ510は、1つ又は複数のピクセル処理演算、たとえば、ラスタライズ及び/又はフラグメントシェーディングを含み、初期処理ステージ505は、1つ又は複数のジオメトリ演算、たとえば、頂点シェーディングを含む。いくつかの例では、さらなる処理ステージ510は、1つ又は複数のピクセル処理演算、たとえば、ラスタライズ及び/又はフラグメントシェーディングを含み、初期処理ステージ505は、1つ又は複数のデータバッファリング操作、たとえば、CPUバッファリングを含む。いくつかの例では、さらなる処理ステージ510は、1つ又は複数のジオメトリ演算、たとえば、フラグメントシェーディングを含み、初期処理ステージ505は、1つ又は複数のデータバッファリング操作、たとえば、CPUバッファリングを含む。

【 0 0 6 9 】

第1の解像度のシーンのデータ及び第2の解像度のシーンのデータが、初期処理ステージ505において処理される。初期処理ステージ505において第1の解像度及び第2の解像度のシーンのデータを処理した後に、バーチャルリアリティーユーザデバイスのユーザの少なくとも片目の現在の位置付けに関する注視追跡データ515が受信される。シーンの少なくとも1つのサブ領域が、受信された注視追跡データ515から特定される。

10

20

30

40

50

【 0 0 7 0 】

受信された注視追跡データ 5 1 5 は、バーチャルリアリティユーザデバイスのユーザの少なくとも片目の中心窩の現在の固視点に関し得る。シーンの特定された少なくとも 1 つのサブ領域は、バーチャルリアリティユーザデバイスのユーザの少なくとも片目の中心窩の現在の固視点を中心とするサブ領域に関し得る。

【 0 0 7 1 】

第 1 の解像度のシーンのデータは、さらなる処理ステージ 5 1 0 において処理される。しかしながら、第 2 の解像度のシーンの特定された少なくとも 1 つのサブ領域に対応するデータのみが、さらなる処理ステージ 5 1 0 において処理される。

【 0 0 7 2 】

グラフィックス処理パイプライン 5 0 0 の各ステージにおけるシーンのデータの処理は、グラフィックス処理パイプライン 5 0 0 の各ステージにおけるシーンのデータのフレームの処理を含み得る。

【 0 0 7 3 】

シーンのデータは、左目ビューに関連するデータと、右目ビューに関連するデータとを含み得る。グラフィックス処理パイプラインの各ステージのデータの処理は、左目ビューに関連するデータの処理と、右目ビューに関連するデータの処理とを含み得る。受信された注視追跡データ 5 1 5 は、バーチャルリアリティユーザデバイスのユーザの左目の中心窩の現在の位置付けと、バーチャルリアリティユーザデバイスのユーザの右目の中心窩の現在の位置付けとに関し得る。

【 0 0 7 4 】

シーンの特定された少なくとも 1 つのサブ領域は、バーチャルリアリティユーザデバイスのユーザの左目の中心窩の現在の位置付けを中心とする左目サブ領域と、バーチャルリアリティユーザデバイスのユーザの右目の中心窩の現在の位置付けを中心とする右目サブ領域とに関し得る。

【 0 0 7 5 】

シーンは、第 1 の解像度でさらなる処理ステージ 5 1 0 において処理されたシーンのデータと、第 2 の解像度でさらなる処理ステージ 5 1 0 において処理されたシーンの特定された少なくとも 1 つのサブ領域に対応するデータとを結合することによってレンダリングされる。

【 0 0 7 6 】

シーンのレンダリングは、最初に、第 1 の解像度でさらなる処理ステージ 5 1 0 において処理されたシーンの左目データと、第 2 の解像度でさらなる処理ステージ 5 1 0 において処理されたシーンの特定された左目サブ領域に対応するデータとを結合し、次に、第 1 の解像度でさらなる処理ステージ 5 1 0 において処理されたシーンの右目データと、第 2 の解像度でさらなる処理ステージ 5 1 0 において処理されたシーンの特定された右目サブ領域に対応するデータとを結合することを含み得る。特定された左目サブ領域の少なくとも一部は、特定された右目サブ領域の少なくとも一部と異なり得る。

【 0 0 7 7 】

いくつかの例では、第 3 の解像度のシーンのデータが、初期処理ステージ 5 0 5 において処理される。注視追跡データ 5 2 0 は、初期処理ステージ 5 0 5 において第 3 の解像度のシーンのデータを処理した後に受信され、初期処理ステージ 5 0 5 において第 3 の解像度のシーンのデータを処理した後のバーチャルリアリティユーザデバイスのユーザの少なくとも片目の現在の位置付けに関する。第 3 の解像度のシーンのデータは、さらなる処理ステージ 5 1 0 において処理される。しかしながら、第 3 の解像度のシーンのさらなるサブ領域に対応するデータのみが、さらなる処理ステージ 5 1 0 において処理される。シーンのレンダリングは、第 3 の解像度でさらなる処理ステージ 5 1 0 において処理されたシーンのデータをさらに結合することを含む。シーンのさらなるサブ領域は、受信された注視追跡データ 5 1 5 から特定され得る。シーンのさらなるサブ領域は、シーンの特定された少なくとも 1 つのサブ領域に基づいて計算され得る。

10

20

30

40

50

【 0 0 7 8 】

図 6 を参照すると、グラフィックス処理パイプライン 6 0 0 の一例が概略的に示されている。グラフィックス処理パイプライン 6 0 0 は、グラフィックス処理システム内に構成される。グラフィックス処理パイプライン 6 0 0 は、初期処理ステージ 6 0 5 と、さらなる処理ステージ 6 1 0 とを含む。

【 0 0 7 9 】

グラフィックス処理パイプライン 6 0 0 は、初期処理ステージ 6 0 5 と、さらなる処理ステージ 6 1 0 とを含む点で、上述のグラフィックス処理パイプライン 5 0 0 と類似している。

【 0 0 8 0 】

しかしながら、この例では、グラフィックス処理パイプライン 6 0 0 は、初期処理ステージ 6 0 5 と、さらなる処理ステージ 6 1 0 との間に中間処理ステージ 6 2 0 をさらに含む。

【 0 0 8 1 】

さらなる処理ステージ 6 1 0 の処理は、中間処理ステージ 6 2 0 の処理よりも高い計算負荷を有し得る。たとえば、さらなる処理ステージ 6 1 0 は、1 つ又は複数のピクセル処理演算、たとえば、ラスタライズ及びノ又はフラグメントシェーディングを含み得、中間処理ステージ 6 2 0 は、1 つ又は複数のジオメトリ演算、たとえば、頂点シェーディングを含み得る。

【 0 0 8 2 】

第 1 の解像度のシーンのデータ及び第 2 の解像度のシーンのデータが、中間処理ステージ 6 2 0 において処理される。中間処理ステージ 6 2 0 において第 1 の解像度及び第 2 の解像度のシーンのデータを処理した後に、注視追跡データ 6 1 5 が受信される。この例では、注視追跡データ 6 1 5 は、中間処理ステージ 6 2 0 において第 1 の解像度及び第 2 の解像度のシーンのデータを処理した後のバーチャルリアリティユーザデバイスのユーザの少なくとも片目の現在の位置付けに関する。注視追跡データ 6 1 5 は、グラフィックス処理パイプライン 6 0 0 の比較的遅くに受信され、したがって、視聴者の 1 つ又は複数の現在の焦点位置を正確に反映している可能性がより高い。しかしながら、グラフィックス処理パイプライン 6 0 0 の比較的遅くに受信されるので、グラフィックス処理パイプライン 6 0 0 において行われる処理全体に与える影響が少ない。

【 0 0 8 3 】

図 7 を参照すると、グラフィックス処理パイプライン 7 0 0 の一例が概略的に示されている。グラフィックス処理パイプライン 7 0 0 は、グラフィックス処理システム内に構成される。グラフィックス処理パイプライン 7 0 0 は、初期処理ステージ 7 0 5 と、中間処理ステージ 7 2 0 と、さらなる処理ステージ 7 1 0 とを含む。

【 0 0 8 4 】

グラフィックス処理パイプライン 7 0 0 は、初期処理ステージ 7 0 5 と、中間処理ステージ 7 2 0 と、さらなる処理ステージ 7 1 0 とを含む点で、図 6 に関連して上述したグラフィックス処理パイプライン 6 0 0 と類似している。

【 0 0 8 5 】

しかしながら、グラフィックス処理パイプライン 6 0 0 では、中間処理ステージ 6 2 0 において第 1 の解像度及び第 2 の解像度のシーンのデータを処理した後に、注視追跡データ 6 1 5 が受信されるが、グラフィックス処理パイプライン 7 0 0 では、中間処理ステージ 7 2 0 において第 1 の解像度及び第 2 の解像度でシーンのデータを処理する前に、注視追跡データ 7 1 5 が受信される。さらに、この例では、注視追跡データ 7 1 5 は、中間処理ステージ 7 2 0 において第 1 の解像度及び第 2 の解像度でシーンのデータを処理する前のバーチャルリアリティユーザデバイスのユーザの少なくとも片目の現在の位置に関する。したがって、この例では、第 1 の解像度のシーンのデータが、中間処理ステージ 7 2 0 において処理される。しかしながら、第 2 の解像度のシーンの特定された少なくとも 1 つのサブ領域に対応するデータのみが、中間処理ステージ 7 2 0 において処理される。加え

10

20

30

40

50

て、上述のように、第1の解像度のシーンのデータ、及び第2の解像度のシーンの特定された少なくとも1つのサブ領域に対応するデータのみが、さらなる処理ステージ710において処理される。注視追跡データ715は、グラフィックス処理パイプライン700のかなり初期の後にやはり受信されるが、上述の注視追跡データ615ほどグラフィックス処理パイプライン700の遅くではない。したがって、注視追跡データ715は、注視追跡データ615に比べて、視聴者の1つ又は複数の現在の焦点を正確に反映している可能性が低い。しかしながら、注視追跡データ715は注視追跡データ615よりもグラフィックス処理パイプライン700の早期に受信されるので、グラフィックス処理パイプライン700において行われる処理全体に与える影響が大きくなり得る。

【0086】

図8を参照すると、グラフィックス処理パイプライン800の一例が概略的に示されている。グラフィックス処理パイプライン800は、グラフィックス処理システム内に構成される。グラフィックス処理パイプライン800は、初期処理ステージ805と、中間処理ステージ820と、さらなる処理ステージ810とを含む。

【0087】

グラフィックス処理パイプライン800は、初期処理ステージ805と、中間処理ステージ820と、さらなる処理ステージ810とを含む点で、図6及び図7に関連して上述したグラフィックス処理パイプライン600、700と類似している。

【0088】

この例では、中間処理ステージ820において第1の解像度及び第2の解像度でシーンのデータを処理する前に、第1の注視追跡データ815が受信される。シーンの少なくとも1つの第1のサブ領域が、第1の注視追跡データ815から特定される。この例では、注視追跡データ815は、中間処理ステージ820において第1の解像度及び第2の解像度でシーンのデータを処理する前のバーチャルリアリティユーザデバイスのユーザの少なくとも片目の現在の位置に関する。したがって、この例では、第1の解像度のシーンのデータが、中間処理ステージ820において処理される。しかしながら、第2の解像度のシーンの特定された少なくとも1つの第1のサブ領域に対応するデータのみが、中間処理ステージ820において処理される。

【0089】

この例では、中間処理ステージ820において第1の解像度及び第2の解像度のシーンのデータを処理した後に、第2の注視追跡データ825が受信される。シーンの少なくとも1つの第2のサブ領域が、第2の注視追跡データ825から特定される。この例では、第2の注視追跡データ825は、中間処理ステージ820において第1の解像度及び第2の解像度のシーンのデータを処理した後のバーチャルリアリティユーザデバイスのユーザの少なくとも片目の現在の位置付けに関する。

【0090】

シーンの少なくとも1つの第2のサブ領域は、シーンの少なくとも1つの第1のサブ領域に含まれ得る。たとえば、第1の注視追跡データ815を用いて、第1の注視追跡データ815に基づいて視聴者が見ている可能性が高いシーンの1つ又は複数の第1の部分に処理を制限することができる。これは、中間処理ステージ820における第2の解像度のシーン全体の処理を節約する。第2の注視追跡データ825を用いて、第2の注視追跡データ825に基づいて視聴者が見ている可能性が高いシーンの1つ又は複数の第2の部分であって、シーンの1つ又は複数の第1の部分より小さいシーンの1つ又は複数の第2の部分にさらなる処理を制限することができる。第2の注視追跡データ825は、シーンが出力されたときにユーザが見ているシーンの部分を指す可能性がより高い。

【0091】

システムは、2つ以上のビューのセットに同時にレンダリングするように構成され得る。これは、混合解像度への効率的なレンダリングを容易にし、レイトラッチの視線追跡データを可能にし得る。レイトラッチの視線追跡データは、グラフィックスパイプラインの初期ステージの後の視線追跡データの消費に関する。本明細書に記載の例では、グラフィ

10

20

30

40

50

ックスパイプラインが提出された後に、視線追跡データのレイトラッチが行われる。

【0092】

最初に視線追跡なしの、複数の解像度にレンダリングするための例示的な技法がここで説明される。上述のように、マルチビューは、同一解像度の単一のビューのセットへのレンダリングを可能にする。これは、`GL_TEXTURE_2D_ARRAY`からの連続した範囲を`GL_DRAW_FRAMEBUFFER`ターゲットへ、たとえば、色、深度及びステンシルなどの1つ又は複数のアタッチメントポイントにおいてバインドする`OpenGL (登録商標) ES API`によって説明される。`GL_DRAW_FRAMEBUFFER`ターゲットは、本明細書では描画ターゲットとも呼ばれる。

【0093】

たとえば、2つのビューへの深度テストを伴うレンダリングは、幅×高さ×`num_views`の深度テクスチャの`GL_TEXTURE_2D_ARRAY`を作成し、幅×高さ×`num_views`の色テクスチャの`GL_TEXTURE_2D_ARRAY`を作成し、フレームバッファオブジェクトを作成し、深度範囲をフレームバッファ描画ターゲットの深度アタッチメントにバインドし、色範囲をフレームバッファ描画ターゲットの色アタッチメントにバインドし、そして、`num_views`で、この例では、`num_views = 2`で出力するように構成されたシェーダによってジオメトリを描画するアプリケーションを必要とし得る。

【0094】

これは、`GL_DRAW_ALT_FRAMEBUFFER`とラベル付けされ、以下、`draw_alt`と称される新たなフレームバッファターゲットを導入することによって、拡張され得る。この新たなターゲットタイプは、異なるが関連する解像度で描画ターゲットに描画されたものの結果を複製するように作用する。`draw_alt`フレームバッファは、実際には描画フレームバッファをミラーリングする。バインド処理は、上述のものと同じであり得るが、簡単のため、`draw_alt`テクスチャは、元の描画構成と一致し得る。たとえば、解像度 X の深度及び色を有する2つのビューが描画ターゲットにバインドされた場合、`draw_alt`にバインドされた関連する解像度 $Y(X)$ の深度及び色を有する2つのビューが存在することになる。いくつかの例では、2つ以上の「`draw_alt`」ターゲットが使用される。いくつかのそのような例では、各「`draw_alt`」ターゲットは異なる関連する解像度を有する。この場合、ターゲットのレイが使用され得る。

【0095】

描画ターゲットのそれぞれは、完全に異なる解像度を有することが許可され得る。代替的には、ターゲットは、たとえば2のべき乗の拡大縮小を可能にするように、関連する解像度を有することができる。これによって、頂点及びタイル化ステージを通じて処理されたジオメトリを、以下説明されるように2つのセット間で再使用することが可能になる。したがって、上記で言及された第1の解像度と第2の解像度とは、たとえば $1/4$ などの2のべき乗を含む所定の関係を有することができる。しかしながら、両方のセットは、依然としてフラグメントシェーディングされ得る。

【0096】

`GPU`は、中心窩レンダリングを行う場合に、シーンの同一のビューの複数の異なる解像度のバージョンをレンダリングするように制御され得る。`GPU`は、たとえば、シーンの同一のビューの3つの異なる解像度のバージョン、すなわち、最高解像度のビュー、中間解像度のビュー、及び最低解像度のビューをレンダリングすることができる。ビューのそれらの異なる解像度の画像は次いで適切に結合(又は「合成」)されて、表示などのために出力中心窩画像を提供する。異なる解像度の画像のそれぞれは、レンダリングされるシーン全体の異なる視野(又は「部分」)を示し得る。一例では、固視点に関連する中心窩画像の領域は最高解像度を有し、固視点から離れた画像の1つ又は複数の領域はより低い解像度で表示される。

【0097】

10

20

30

40

50

レンダリングを要求するアプリケーションは、シーンのジオメトリを中心窩ビューに関連する倍率と共に複数のビューにレンダリングされるべきことをグラフィックスプロセッサのドライバに指示することができる。アプリケーションからのそのようなコマンドに回答して、ドライバは次いで、GPUに送信する適切なタイル化及びレンダリングのタスクを、本明細書に記載のように中心窩レンダリングを行うように構成する。

【0098】

一例では、シーンのジオメトリは、処理されタイル化され、換言すれば、一度のみレンダリングされる画像のそれぞれのレンダリングタイルにソートされる。これは単一のタイルジオメトリリストのセットを提供し、これは次いで、各それぞれの解像度の画像をレンダリングする場合に、すなわち、各それぞれの解像度の画像をレンダリングするためにフラグメント処理を行う場合に、共通で使用される（又は「共有される」）。特に、GPUは、出力画像に必要とされる最高解像度で、レンダリングされるシーンに対して一度のみ頂点位置処理及びジオメトリリスト生成処理の両方を行うように制御され得る。このようにジオメトリリストが準備されると、その単一のジオメトリリストのセットは次いで、ビューの一部又は全部を示す各それぞれの異なる解像度の画像をレンダリングする場合に使用され得る。

10

【0099】

最高解像度のビューについて、準備されたジオメトリリストは、その最高解像度の画像をレンダリングする場合に各レンダリングタイルについて処理されるべきジオメトリを特定するとき使用され得る。他のより低い解像度の画像について、最高解像度で準備されたジオメトリリスト及びそれらのリスト内のジオメトリを用いて、完全解像度のジオメトリリスト内のジオメトリを適切な倍率で縮小することによって、より低い解像度の画像についてレンダリングされるべきジオメトリを特定し定義する。倍率は、線形の拡大縮小などを用いて、ジオメトリリストが準備された最高解像度に、問題のより低い解像度の画像の解像度を関連付ける。この縮小は、頂点位置ロード操作の一部としてなされ得るが、他の構成も所望であれば可能である。

20

【0100】

いくつかの例では、最終画像の全部ではなく一部のみが異なるそれぞれの解像度でレンダリングされる。特に、最終的な合成で使用されない画像の領域はレンダリングされなくてもよい。各異なる解像度の画像の選択部分のみのレンダリングは、出力画像のレンダリングタイルごとに、そのタイル位置について生成されるべき1つ又は複数の解像度の画像を指示することによって実現され得る。この情報は、レンダリングタイルについてのジオメトリリストと共に提供され得る。他の構成、たとえば、GPUのドライバが、レンダリングタスクをグラフィックスプロセッサに提供し、及び/又は、画像についてタイルがレンダリングされるべきでないエリアを特定するレンダー除外領域を指示する場合に、レンダリングされるべき適切なタイルのリストを提供するように構成されたものが代わりに使用され得る。

30

【0101】

各異なる解像度の画像の適切な部分がGPUによってレンダリングされると、それらの画像はグラフィックス処理システムのメモリ内の適切なフレームバッファに記憶される。それらの画像は次いで適切に結合されて、表示される最終的な出力中心窩ビューを提供する。

40

【0102】

最終的な中心窩出力ビューを提供するためのビューの異なる解像度の画像の合成は、GPUが、各異なる解像度のビューをグラフィックステクスチャとして扱い、次いで、フラグメントシェーダの一部として実装され得るテクスチャマッピング演算を用いて、表示される最終的な出力中心窩ビューにわたって異なる解像度の画像を適切にサンプリングし結合することによって行われ得る。

【0103】

そのような中心窩レンダリングを行う場合、最高解像度の画像は、表示される最終出力

50

画像の中心に必ずしもある必要はないが、たとえば、視聴者が実際に見ている場所を特定するための注視追跡センサを使用して決定され得る1つ又は複数の固視点の位置に応じて、ビューの任意の場所にオフセットされ得る。

【0104】

最高解像度のセットをレンダリングしダウンサンプリングして他のセットを生成することは、同一の結果を生成しない。

【0105】

マルチサンプリングのシェーディングレートは、セット間で変更し得る。マルチビューでは、ターゲットにバインドされた全てのテクスチャ間で解像度が一致することが好ましいように、マルチサンプルカウントもそのように一致することが好ましい。しかしながら、解像度、マルチサンプルカウント、テクスチャフォーマット及びレイアウトは、異なるセット間で変更し得る。レンダリングされ得るテクスチャを記述するものは変更し得る。

10

【0106】

上記のように、本明細書に記載の例は、レイトラッチの視線追跡データを用いた中心窩レンダリングに関する。

【0107】

いくつかの例では、ハードウェアフラグメントジョブ、又はワークアイテムの記述子は、2Dの軸調整されたバウンディングボックスを含み、これはバウンディングボックスによって囲まれた領域のちょうど内側にフラグメントシェーディングを制限するものである。これは、完全解像度のビューのシェーディングを中心窩領域ちょうどに制限するための効率的なメカニズムであり、中心窩領域はバウンディングボックスに関連付けられる。バウンディングボックスデータは、フラグメントシェーディングの開始の直前に使用され、したがって、グラフィックスパイプラインの遅くに変更し得る。これは、使用されている注視追跡データの期間に関連して、せいぜい1フレームのレイテンシ、すなわち、60Hzで16.7msが存在することを意味する。これは、完全なグラフィックスパイプラインでの50msと対照的である。レイテンシは、単一のフレーム内でジオメトリが提出され、タイル化され、シェーディングされる場合にさらに削減され得る。

20

【0108】

以前に提出されたハードウェアフラグメントジョブ記述子にコピーされた任意の「後期の」バウンディングボックスデータが、それを正しく読み取るために必要とされ得る。新たなデータが与えられれば、これは、フラグメントジョブがまだ開始されていないことを知っている場合はドライバが、又はGPUコンピュートシェーダが、共有バッファからデータをコピーすることによってなされ得る。

30

【0109】

パイプラインの遅くにデータを提供することは、OpenGL(登録商標)ESでは一般的ではないが、データへの手動で同期されるアクセスはよく理解されている。OpenGL(登録商標)ESにおけるコンピュートシェーダは、デフォルトでは同期していない。ユーザは同期プリミティブを用いて正しい動作を実現する。これは、フラグメント記述子を適時に更新するのに十分なインフラストラクチャを提供する。一実装は、アトミック(atomic)及び共有バッファを活用する、又はカスタムAPIエントリーポイントを介して等価な機能を公開することができる。

40

【0110】

Vulkanは、OpenGL(登録商標)ESよりもローレベルであり、また、明示的に要求されない限り、いかなる同期も行わず、したがって、OpenGL(登録商標)ESと同様のメカニズムがVulkanで動作する。Vulkanの一つの特徴は、そのVkRenderPassオブジェクトが、シェーディングを指定領域に制限する2Dの軸調整されたバウンディングボックスを定義する明示的な「renderArea」特性を既に有していることである。Vulkan拡張の開発は、OpenGL(登録商標)ES拡張の開発よりも効率的であり得、その理由は、バウンディングボックスの概念がVulkanのAPIに既に存在するためである。VulkanはVkRenderPass

50

オブジェクトの拡張を導入することができ、これによってユーザは、その他を「ミラーリング」する第2のターゲットのセットを定義することができる。これは、上述の「GL_DRAW_ALT_FRAME_BUFFER」機能の等価物を可能にし得る。

【0111】

特定の例では、2つのビューを有するマルチビューを仮定すると、一実装は、完全解像度の2つのビューのテクスチャレイ（色、深度など）のセットを作成し、それらのレイをフレームバッファ内の「描画」ターゲットにバインドする。それらのテクスチャレイに対するフラグメントバウンディングボックスは、最も直前に報告された視線追跡データに適合するように更新される。システムは、他の2つのビューであるが、完全解像度のビューの解像度の4分の1のものについて作成された適合するテクスチャレイのセットを作成し、それらのビューを「draw_alt」ターゲットにバインドする。次いで、描画ターゲット及びdraw_altターゲットを消費することによって、シーンが描画される。したがって、この例では、第1の解像度と第2の解像度との間の所定の関係は1/4である。

10

【0112】

次いで、各フレームの開始時に、又は定期的な間隔で、全ての関連する提出された、まだ修正可能なフラグメントジョブは、最新のバウンディングボックスを使用するように更新される。このための理想的な時間は、新たなトラッキングデータが受信された後であり、これはメインのレンダリンググループとは非同期に動作することができる。

20

【0113】

この特定の例は、ステレオ画像全体についての4分の1の解像度の出力と、目の位置に一致する小さいが完全解像度の領域のペアとを生成する。これらは次いでオーバーレイされて、最終画像を生成する。このオーバーレイ処理は、後続のレンズ補正又は合成処理、たとえば、低レイテンシVRシステムに存在し、他の用語の中でも特に、「タイムワープ」、「非同期タイムワープ」、「電子ディスプレイ同期」として知られているものに含めるのに適している。これは、領域の位置及び大きさの知識を追跡して使用する。また、タイムワープ処理は、2つの領域間でブレンドして、よりソフトな遷移を作成する、又は完全解像度をより小さい代替的な形状にトリミングする、たとえば、ボックスを球にトリミングすることを選択することもできる。

30

【0114】

上記の実施形態は、例示的な例として理解されるべきである。さらなる実施形態が想定される。任意の1つの実施形態に関して記載された任意の特徴が、単独で又は記載の他の特徴と組み合わせて使用され得、また、任意の他の実施形態又は任意の他の実施形態の任意の組み合わせの1つ又は複数の特徴と組み合わせて使用され得ることは理解されたい。さらに、上記で説明されていない等価物及び修正物もまた、添付の特許請求の範囲に定義された、本開示の実施形態の範囲から逸脱することなく利用され得る。

[発明の項目]

[項目1]

グラフィックス処理システムを動作させる方法であって、前記システムが、少なくとも初期処理ステージとさらなる処理ステージとを含むグラフィックス処理パイプラインを含み、前記方法が、

40

前記初期処理ステージにおいて、第1の解像度でシーンのデータを処理するステップと、

前記初期処理ステージにおいて、第2の解像度で前記シーンのデータを処理するステップと、

前記初期処理ステージにおいて前記第1の解像度及び前記第2の解像度で前記シーンのデータを処理した後、バーチャルリアリティユーザデバイスのユーザの少なくとも片目の現在の位置付けに関する注視追跡データを受信するステップと、

受信された前記注視追跡データから、前記シーンの少なくとも1つのサブ領域を特定するステップと、

50

前記さらなる処理ステージにおいて、前記第 1 の解像度で前記シーンのデータを処理するステップと、

前記さらなる処理ステージにおいて、前記シーンの特定された前記少なくとも 1 つのサブ領域に対応するデータのみを前記第 2 の解像度で処理するステップと、

前記さらなる処理ステージにおいて前記第 1 の解像度で処理された前記シーンの前記データと、前記さらなる処理ステージにおいて前記第 2 の解像度で処理された前記シーンの特定された前記少なくとも 1 つのサブ領域に対応する前記データと、を結合することによって、前記シーンをレンダリングするステップと、
を含む、方法。

[項目 2]

前記グラフィックス処理パイプラインが、前記初期処理ステージと前記さらなる処理ステージとの間に中間処理ステージを含み、前記方法が、

前記中間処理ステージにおいて、前記第 1 の解像度で前記シーンのデータを処理するステップと、

前記中間処理ステージにおいて、前記第 2 の解像度で前記シーンの特定された前記少なくとも 1 つのサブ領域に対応するデータのみを処理するステップと、
を含み、

前記注視追跡データが、前記中間処理ステージにおいて前記第 1 の解像度及び前記第 2 の解像度で前記シーンのデータを処理する前に受信され、前記中間処理ステージにおいて前記第 1 の解像度及び前記第 2 の解像度で前記シーンのデータを処理する前の前記バーチャルリアリティユーザデバイスのユーザの少なくとも片目の現在位置に関する、項目 1 に記載の方法。

[項目 3]

前記グラフィックス処理パイプラインが、前記初期処理ステージと前記さらなる処理ステージとの間に中間処理ステージを含み、前記方法が、

前記中間処理ステージにおいて、前記第 1 の解像度で前記シーンのデータを処理するステップと、

前記中間処理ステージにおいて、前記第 2 の解像度で前記シーンのデータを処理するステップと、
を含み、

前記注視追跡データが、前記中間処理ステージにおいて前記第 1 の解像度及び前記第 2 の解像度で前記シーンのデータを処理した後に受信され、前記中間処理ステージにおいて前記第 1 の解像度及び前記第 2 の解像度で前記シーンのデータを処理した後の前記バーチャルリアリティユーザデバイスのユーザの少なくとも片目の現在の位置付けに関する、項目 1 に記載の方法。

[項目 4]

前記さらなる処理ステージの処理が、前記中間処理ステージの処理よりも高い計算負荷を有する、項目 2 又は 3 に記載の方法。

[項目 5]

前記中間処理ステージが、1 つ又は複数のジオメトリ演算を含む、項目 2 ~ 4 のいずれか一項に記載の方法。

[項目 6]

前記さらなる処理ステージの処理が、前記初期処理ステージの処理よりも高い計算負荷を有する、項目 1 ~ 5 のいずれか一項に記載の方法。

[項目 7]

前記さらなる処理ステージが、1 つ又は複数のピクセル処理演算を含む、項目 1 ~ 6 のいずれか一項に記載の方法。

[項目 8]

前記初期処理ステージが、1 つ又は複数のデータバッファリング操作を含む、項目 1 ~ 7 のいずれか一項に記載の方法。

10

20

30

40

50

[項目 9]

前記グラフィックス処理パイプラインの各ステージにおいてシーンのデータを処理するステップが、前記グラフィックス処理パイプラインの各ステージにおいて前記シーンのデータのフレームを処理するサブステップを含む、項目 1 ~ 8 のいずれか一項に記載の方法。

[項目 10]

前記第 1 の解像度が前記第 2 の解像度比べて比較的高い、項目 1 ~ 9 のいずれか一項に記載の方法。

[項目 11]

前記第 1 の解像度と前記第 2 の解像度とが所定の関係を有する、項目 1 ~ 10 のいずれか一項に記載の方法。

10

[項目 12]

前記所定の関係が、2 のべき乗を含む、項目 11 に記載の方法。

[項目 13]

前記所定の関係が、1 / 4 を含む、項目 11 又は 12 に記載の方法。

[項目 14]

受信された前記注視追跡データが、前記バーチャルリアリティユーザデバイスのユーザの少なくとも片目の中心窩の現在の固視点に関し、

前記シーンの特定された前記少なくとも 1 つのサブ領域が、前記バーチャルリアリティユーザデバイスのユーザの前記少なくとも片目の前記中心窩の前記現在の固視点を中心とするサブ領域に関する、項目 1 ~ 13 のいずれか一項に記載の方法。

20

[項目 15]

前記シーンのデータが、左目ビューに関連するデータと右目ビューに関連するデータとを含み、前記グラフィックス処理パイプラインの各ステージの前記データを処理するステップが、前記左目ビューに関連するデータを処理するサブステップと、前記右目ビューに関連するデータを処理するサブステップと、を含み、

受信された前記注視追跡データが、前記バーチャルリアリティユーザデバイスのユーザの左目の中心窩の現在の位置付けと、前記バーチャルリアリティユーザデバイスのユーザの右目の中心窩の現在の位置付けと、に関する、項目 1 ~ 14 のいずれか一項に記載の方法。

30

[項目 16]

前記シーンの特定された前記少なくとも 1 つのサブ領域が、前記バーチャルリアリティユーザデバイスの前記ユーザの前記左目の前記中心窩の前記現在の位置付けを中心とする左目サブ領域と、前記バーチャルリアリティユーザデバイスの前記ユーザの前記右目の前記中心窩の前記現在の位置付けを中心とする右目サブ領域と、に関し、

前記シーンをレンダリングするステップが、前記さらなる処理ステージにおいて前記第 1 の解像度で処理された前記シーンの左目データと、前記さらなる処理ステージにおいて前記第 2 の解像度で処理された前記シーンの特定された前記左目サブ領域に対応するデータと、を結合するサブステップと、

前記さらなる処理ステージにおいて前記第 1 の解像度で処理された前記シーンの右目データと、前記さらなる処理ステージにおいて前記第 2 の解像度で処理された前記シーンの特定された前記右目サブ領域に対応するデータと、を結合するサブステップと、を含む、項目 15 に記載の方法。

40

[項目 17]

特定された前記左目サブ領域のうちの少なくとも一部が、特定された前記右目サブ領域のうちの少なくとも一部と異なる、項目 16 に記載の方法。

[項目 18]

前記初期処理ステージにおいて、第 3 の解像度で前記シーンのデータを処理するステップ

を含み、

50

前記注視追跡データが、前記初期処理ステージにおいて前記第3の解像度で前記シーンのデータを処理した後に受信され、前記初期処理ステージにおいて前記第3の解像度で前記シーンのデータを処理した後の前記バーチャルリアリティユーザデバイスのユーザの少なくとも片目の現在の位置付けに関し、前記方法が、

前記さらなる処理ステージにおいて、前記第3の解像度で前記シーンのデータを処理するステップと、

前記さらなる処理ステージにおいて、前記第3の解像度で前記シーンのさらなるサブ領域に対応するデータのみを処理するステップと、
を含み、

前記シーンをレンダリングするステップが、前記さらなる処理ステージにおいて前記第3の解像度で処理された前記シーンの前記データをさらに結合するサブステップを含む、
項目1～17のいずれか一項に記載の方法。

[項目19]

受信された前記注視追跡データから前記シーンの前記さらなるサブ領域を特定するステップを含む、
項目18に記載の方法。

[項目20]

前記シーンの特定された前記少なくとも1つのサブ領域に基づいて前記シーンの前記さらなるサブ領域を計算するステップを含む、
項目18に記載の方法。

[項目21]

グラフィックス処理パイプラインを含むグラフィックス処理システムを動作させる方法であって、前記グラフィックス処理パイプラインにおいてレイトラッチの視線追跡データを用いて、シーンの1つ又は複数の領域であり、前記シーンの1つ又は複数の他の領域の解像度に比べて比較的高い解像度でレンダリングされるべき1つ又は複数の領域を特定するステップを含む、
方法。

[項目22]

グラフィックス処理システムを動作させる方法であって、前記システムが、少なくとも初期処理ステージとさらなる処理ステージとを含むグラフィックス処理パイプラインを含み、前記方法が、

グラフィックス処理ワークアイテムについての入力データを前記グラフィックス処理パイプラインに第1の時間に提供するステップと、

前記グラフィックス処理パイプラインが前記グラフィックス処理ワークアイテムを処理している間第2のより遅い時間に、バーチャルリアリティユーザデバイスのユーザの少なくとも片目の現在の位置付けに関する注視追跡データを、前記グラフィックス処理パイプラインに提供するステップと、

を含む、
方法。

[項目23]

グラフィックスプロセッサを動作させる方法であって、前記グラフィックスプロセッサが、少なくとも初期処理ステージとさらなる処理ステージとを含むグラフィックス処理パイプラインに関連付けられ、前記方法が、

前記初期処理ステージにおいて、第1の解像度でシーンのデータを処理するステップと

、
前記初期処理ステージにおいて、第2の解像度で前記シーンのデータを処理するステップと、

前記初期処理ステージにおいて前記第1の解像度及び前記第2の解像度で前記シーンのデータを処理した後、バーチャルリアリティユーザデバイスのユーザの少なくとも片目の現在の位置付けに関する注視追跡データを受信するステップと、

受信された前記注視追跡データから、前記シーンの少なくとも1つのサブ領域を特定するステップと、

前記さらなる処理ステージにおいて、前記第1の解像度で前記シーンのデータを処理するステップと、

10

20

30

40

50

前記さらなる処理ステージにおいて、前記第 2 の解像度で前記シーンの特定された前記少なくとも 1 つのサブ領域に対応するデータのみを処理するステップと、

前記さらなる処理ステージにおいて前記第 1 の解像度で処理された前記シーンの前記データと、前記さらなる処理ステージにおいて前記第 2 の解像度で処理された前記シーンの特定された前記少なくとも 1 つのサブ領域に対応する前記データと、を結合することによって、前記シーンをレンダリングするステップと、
を含む、方法。

[項目 2 4]

グラフィックス処理システムにおいて実行された場合に、項目 1 ~ 2 3 のいずれか一項に記載の方法を実施するように構成されたソフトウェアコードを含む、コンピュータプログラム。

[項目 2 5]

グラフィックス処理パイプラインを含むグラフィックス処理システムであって、前記グラフィックス処理パイプラインが、少なくとも初期処理ステージとさらなる処理ステージとを含み、前記グラフィックス処理システムが、

前記初期処理ステージにおいて、第 1 の解像度でシーンのデータを処理し、
前記初期処理ステージにおいて、第 2 の解像度で前記シーンのデータを処理し、
前記初期処理ステージにおいて前記第 1 の解像度及び前記第 2 の解像度で前記シーンのデータを処理した後、バーチャルリアリティユーザデバイスのユーザの少なくとも片目の現在の位置付けに関する注視追跡データを受信し、

受信された前記注視追跡データから、前記シーンの少なくとも 1 つのサブ領域を特定し

、
前記さらなる処理ステージにおいて、前記第 1 の解像度で前記シーンのデータを処理し

、
前記さらなる処理ステージにおいて、前記第 2 の解像度で前記シーンの特定された前記少なくとも 1 つのサブ領域に対応するデータのみを処理し、

前記さらなる処理ステージにおいて前記第 1 の解像度で処理された前記シーンの前記データと、前記さらなる処理ステージにおいて前記第 2 の解像度で処理された前記シーンの特定された前記少なくとも 1 つのサブ領域に対応する前記データと、を結合することによって、前記シーンをレンダリングする、
ように構成された、グラフィックス処理システム。

[項目 2 6]

グラフィックス処理パイプラインを含むグラフィックス処理システムであって、前記グラフィックス処理パイプラインにおいてレイトラッチの視線追跡データを用いて、シーンの 1 つ又は複数の領域であり、前記シーンの 1 つ又は複数の他の領域の解像度に比べて比較的高い解像度でレンダリングされるべき 1 つ又は複数の領域を特定するように構成された、グラフィックス処理システム。

[項目 2 7]

少なくとも初期処理ステージとさらなる処理ステージとを含むグラフィックス処理パイプラインを含むグラフィックス処理システムであって、

グラフィックス処理ワークアイテムについての入力データを前記グラフィックス処理パイプラインに第 1 の時間に提供し、

前記グラフィックス処理パイプラインが前記グラフィックス処理ワークアイテムを処理している間第 2 のより遅い時間に、バーチャルリアリティユーザデバイスのユーザの少なくとも片目の現在の位置付けに関する注視追跡データを、前記グラフィックス処理パイプラインに提供する、

ように構成された、グラフィックス処理システム。

[項目 2 8]

項目 2 5 ~ 2 7 のいずれか一項に記載のグラフィックス処理システムを備える、バーチャルリアリティユーザデバイス。

10

20

30

40

50

[項目 2 9]

少なくとも初期処理ステージとさらなる処理ステージとを含むグラフィックス処理パイプラインに関連付けられたグラフィックスプロセッサであって、

前記初期処理ステージにおいて、第 1 の解像度でシーンのデータを処理し、

前記初期処理ステージにおいて、第 2 の解像度で前記シーンのデータを処理し、

前記初期処理ステージにおいて前記第 1 の解像度及び前記第 2 の解像度で前記シーンのデータを処理した後、バーチャルリアリティユーザデバイスのユーザの少なくとも片目の現在の位置付けに関する注視追跡データを受信し、

受信された前記注視追跡データから、前記シーンの少なくとも 1 つのサブ領域を特定し

、

前記さらなる処理ステージにおいて、前記第 1 の解像度で前記シーンのデータを処理し

、

前記さらなる処理ステージにおいて、前記第 2 の解像度で前記シーンの特定された前記少なくとも 1 つのサブ領域に対応するデータのみを処理し、

前記さらなる処理ステージにおいて前記第 1 の解像度で処理された前記シーンの前記データと、前記さらなる処理ステージにおいて前記第 2 の解像度で処理された前記シーンの特定された前記少なくとも 1 つのサブ領域に対応する前記データと、を結合することによって、前記シーンをレンダリングする、

ように構成された、グラフィックスプロセッサ。

[項目 3 0]

項目 2 9 に記載のグラフィックスプロセッサを備える、バーチャルリアリティユーザデバイス。

[項目 3 1]

添付の図面を参照して本明細書において実質的に記載された、方法。

[項目 3 2]

添付の図面を参照して本明細書において実質的に記載された、グラフィックス処理システム。

[項目 3 3]

添付の図面を参照して本明細書に実質的に記載された、グラフィックスプロセッサ。

[項目 3 4]

添付の図面を参照して本明細書に実質的に記載された、グラフィックス処理パイプライン。

【 符号の説明 】【 0 1 1 5 】

- 1 0 5 アプリケーション
- 1 1 0 ホスト
- 1 1 5 GPU
- 1 2 0 ドライバ
- 2 0 5 頂点シェーダ
- 2 1 0 ハルシェーダ
- 2 1 5 テッセレータ
- 2 2 0 ドメインシェーダ
- 2 2 5 ジオメトリシェーダ
- 2 3 0 タイラ
- 2 3 5 ラスタライザ
- 2 4 0 早期深度及びステンシルテスト
- 2 4 5 フラグメントシェーダ
- 2 5 0 後期深度及びステンシルテスト
- 2 5 5 ブレンダ
- 2 6 0 タイルバッファ

10

20

30

40

50

- 2 6 5 書き出し
- 2 7 0 フレームバッファ
- 3 0 5 初期処理ステージ
- 3 1 0 さらなる処理ステージ
- 3 1 5 注視追跡データ
- 4 0 5 CPUバッファリング
- 4 1 0 ジオメトリフェーズ
- 4 1 5 ピクセルフェーズ
- 5 0 5 初期処理ステージ
- 5 1 0 さらなる処理ステージ
- 5 1 5 注視追跡データ
- 6 0 5 初期処理ステージ
- 6 1 0 さらなる処理ステージ
- 6 1 5 注視追跡データ
- 6 2 0 中間処理ステージ
- 7 0 5 初期処理ステージ
- 7 1 0 さらなる処理ステージ
- 7 1 5 注視追跡データ
- 7 2 0 中間処理ステージ
- 8 0 5 初期処理ステージ
- 8 1 0 さらなる処理ステージ
- 8 1 5 第1の注視追跡データ
- 8 2 0 中間処理ステージ
- 8 2 5 第2の注視追跡データ

10

20

【図1】

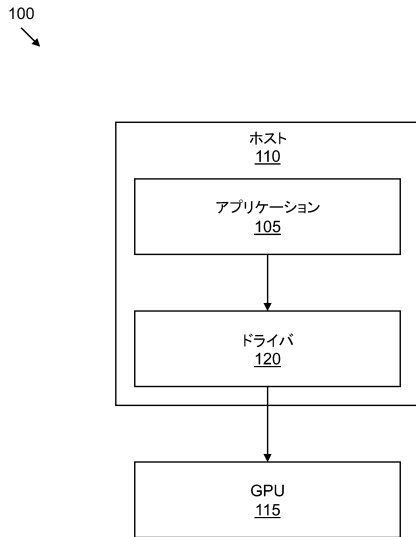


FIG. 1

【図2】

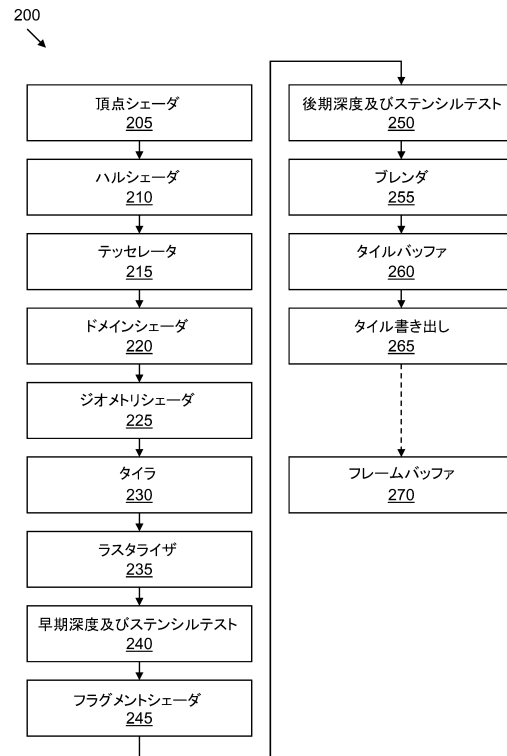


FIG. 2

【 図 3 】

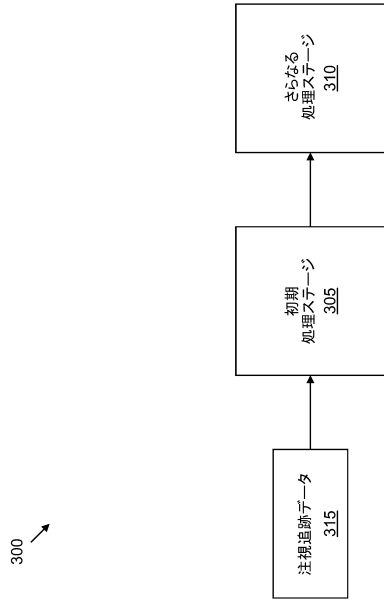


FIG. 3

【 図 4 】

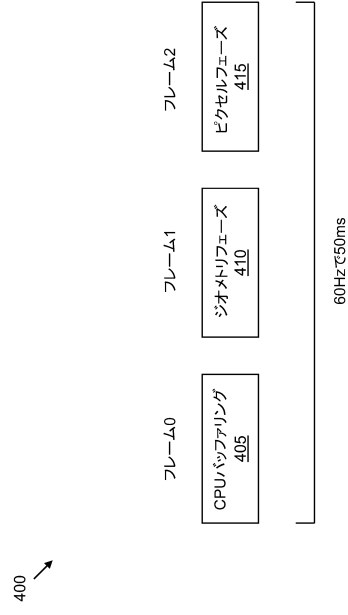


FIG. 4

【 図 5 】

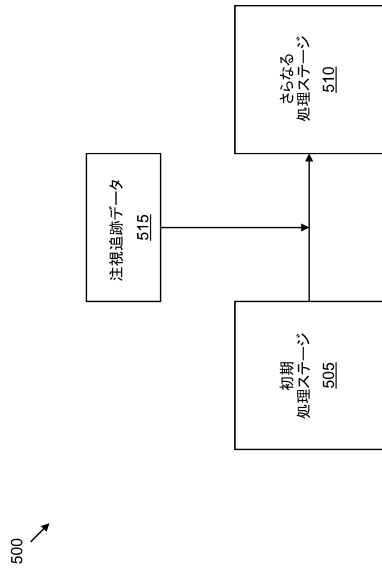


FIG. 5

【 図 6 】

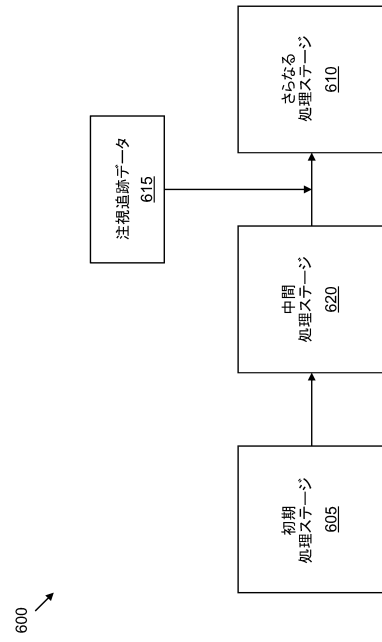


FIG. 6

【 図 7 】

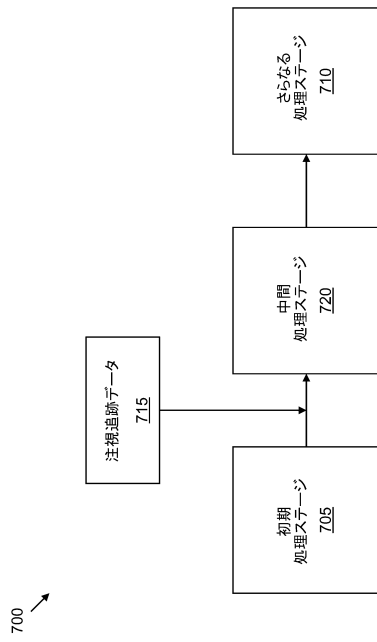


FIG. 7

【 図 8 】

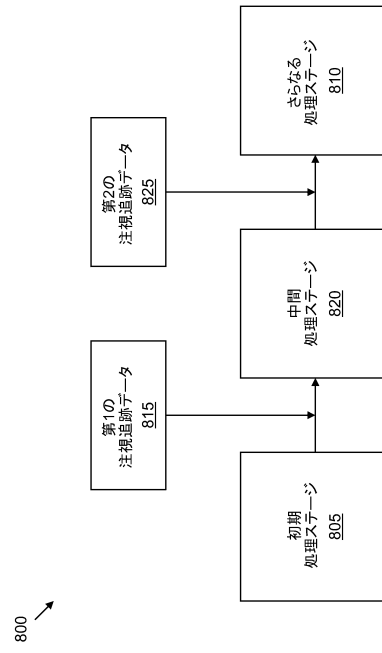


FIG. 8

フロントページの続き

(72)発明者 マーティン, サミュエル
イギリス, シービー1 9エヌジェイ, ケンブリッジシャー, ケンブリッジ, フルボーン
ロード 110, ケア オブ アーム リミテッド

審査官 片岡 利延

(56)参考文献 国際公開第2015/153165(WO, A1)
米国特許出願公開第2017/0200308(US, A1)
Brian Guenter et al., Foveated 3D Graphics, [online], 2012年11月, <https://dl.acm.org/doi/10.1145/2366145.2366183>

(58)調査した分野(Int.Cl., DB名)
G06T 15/00
G06T 19/00