

(19)日本国特許庁(JP)

## (12)特許公報(B2)

(11)特許番号

特許第7252209号

(P7252209)

(45)発行日 令和5年4月4日(2023.4.4)

(24)登録日 令和5年3月27日(2023.3.27)

(51)国際特許分類

F I

G 0 6 T 15/00 (2011.01)

G 0 6 T

15/00

5 0 1

G 0 6 T 15/04 (2011.01)

G 0 6 T

15/04

請求項の数 20 (全21頁)

(21)出願番号 特願2020-510572(P2020-510572)

(86)(22)出願日 平成30年8月22日(2018.8.22)

(65)公表番号 特表2020-531994(P2020-531994  
A)

(43)公表日 令和2年11月5日(2020.11.5)

(86)国際出願番号 PCT/US2018/047539

(87)国際公開番号 WO2019/040630

(87)国際公開日 平成31年2月28日(2019.2.28)

審査請求日 令和3年8月20日(2021.8.20)

(31)優先権主張番号 15/687,108

(32)優先日 平成29年8月25日(2017.8.25)

(33)優先権主張国・地域又は機関  
米国(US)

(73)特許権者 591016172

アドバンスト・マイクロ・ディバイス  
・インコーポレイテッドADVANCED MICRO DEVI  
CES INCORPORATEDアメリカ合衆国 9 5 0 5 4 カリフォル  
ニア州、 サンタ クララ、 オーガスティ  
ン ドライブ 2 4 8 5

(74)代理人 100108833

弁理士 早川 裕司

(74)代理人 100111615

弁理士 佐野 良太

(74)代理人 100162156

弁理士 村雨 圭介

(72)発明者 マキシム ヴィー . カザコフ

最終頁に続く

(54)【発明の名称】 圧縮メタデータを用いたテキスト常驻チェック

## (57)【特許請求の範囲】

## 【請求項 1】

テキストチャブロックと、前記テキストチャブロックの圧縮パラメータと前記テキストチャブロックがメモリに常驻しているかどうかを示す常驻状態とを符号化するメタデータと、を記憶するメモリにアクセスするように構成されたパイプラインと、

シェーディング操作を実行するために前記テキストチャブロック内のデータを要求すると共に、前記メタデータへのアクセスを要求するプロセッサと、を備え、

前記パイプラインは、前記テキストチャブロックが前記メモリに常驻していることを前記常驻状態が示しているかどうかに応じて、前記テキストチャブロック内の前記データを選択的に前記プロセッサに返す、

装置。

## 【請求項 2】

前記パイプラインは、前記テキストチャブロックが前記メモリに常驻していることを前記常驻状態が示していることに応じて、前記テキストチャブロック内の前記データを返す、

請求項 1 の装置。

## 【請求項 3】

前記プロセッサは、前記テキストチャブロックが前記メモリに常驻していないことを前記常驻状態が示していることに応じて、前記要求されたデータが利用可能でないことを示す信号を受信し、前記信号を受信したことに応じて、データを前記テキストチャブロックに投入し、前記データを前記テキストチャブロックに投入したことに応じて、前記テキストチャブ

ロックを再要求すると共に前記メタデータに再アクセスする、  
請求項 1 の装置。

【請求項 4】

前記プロセッサは、前記メタデータを含むメタデータ面の第 1 パスの間に、前記テキストチャブロックを含む複数のテキストチャブロックの常駐状態及びサンプリング状態のうち少なくとも 1 つを決定し、前記常駐状態及び前記サンプリング状態のうち前記少なくとも 1 つに基づいて前記複数のテキストチャブロックに選択的に投入し、第 2 パスの間に前記複数のテキストチャブロック内のテキストチャデータをサンプリングする、

請求項 1 ～ 3 の何れかの装置。

【請求項 5】

前記テキストチャブロックの前記圧縮パラメータ及び前記常駐状態を符号化する前記メタデータのコピーを記憶するエントリを含むキャッシュをさらに備える、

請求項 1 ～ 3 の何れかの装置。

【請求項 6】

前記キャッシュに記憶された前記メタデータのサンプリング状態は、前記テキストチャブロックが前記メモリに常駐していないことを前記テキストチャブロックの以前の常駐チェックが示したかどうかを示す、

請求項 5 の装置。

【請求項 7】

前記サンプリング状態は、前記キャッシュの前記エントリに記憶された前記メタデータへのアクセス要求に応じて変更される、

請求項 6 の装置。

【請求項 8】

前記プロセッサは、前記テキストチャブロックが前記メモリに常駐していないことを前記常駐状態が示していることと、前記キャッシュが前記エントリ内の前記メタデータにアクセスするための少なくとも 1 つの以前の要求を受信したことを前記サンプリング状態が示していることとに応じて、データを前記テキストチャブロックに投入する、

請求項 7 の装置。

【請求項 9】

テキストチャブロックの圧縮パラメータと前記テキストチャブロックがメモリに常駐しているかどうかを示す常駐状態とを符号化するメタデータにアクセスすると共に、前記メモリに記憶された前記テキストチャブロックのデータへのアクセスを要求することと、

データが前記メモリに常駐していることを前記常駐状態が示しているかどうかに応じて、前記テキストチャブロック内の前記データを選択的に受信することと、を含む、

方法。

【請求項 10】

前記テキストチャブロック内の前記データを選択的に受信することは、前記テキストチャブロックが前記メモリに常駐していることを前記常駐状態が示していることに応じて、前記テキストチャブロック内の前記データを受信することを含む、

請求項 9 の方法。

【請求項 11】

前記データを選択的に受信することは、

前記テキストチャブロックが前記メモリに常駐していないことを前記常駐状態が示していることに応じて、前記要求されたデータが利用可能でないことを示す信号を受信することを含む、

前記信号を受信したことに応じてデータを前記テキストチャブロックに投入することと、

前記データを前記テキストチャブロックに投入したことに応じて、前記テキストチャブロック内の前記データを再要求すると共に前記メタデータに再アクセスすることと、をさらに含む、

請求項 9 の方法。

10

20

30

40

50

## 【請求項 1 2】

前記メタデータを含むメタデータ面の第 1 パスの間に、前記テキストチャブロックを含む複数のテキストチャブロックの前記常駐状態及びサンプリング状態のうち少なくとも 1 つを決定することと、

前記常駐状態及び前記サンプリング状態のうち前記少なくとも 1 つに基づいて前記複数のテキストチャブロックに選択的に投入することと、

第 2 パスの間に前記複数のテキストチャブロック内のテキストチャデータをサンプリングすることと、をさらに含む、

請求項 9 ~ 1 1 の何れかの方法。

## 【請求項 1 3】

前記テキストチャブロックの前記圧縮パラメータ及び前記常駐状態を符号化する前記メタデータのコピーをキャッシュすることをさらに含む、

請求項 9 の方法。

## 【請求項 1 4】

キャッシュに記憶された前記メタデータのサンプリング状態は、前記テキストチャブロックが前記メモリに常駐していないことを前記テキストチャブロックの以前の常駐チェックが示したかどうかを示す、

請求項 1 3 の方法。

## 【請求項 1 5】

前記メタデータの前記キャッシュされたコピーへのアクセス要求に応じて前記サンプリング状態を変更することをさらに含む、

請求項 1 4 の方法。

## 【請求項 1 6】

前記テキストチャブロックが前記メモリに常駐していないことを前記常駐状態が示していることと、前記キャッシュが前記メタデータの前記キャッシュされたコピーにアクセスするための少なくとも 1 つの以前の要求を受信したことを前記サンプリング状態が示していることとに応じて、データを前記テキストチャブロックに投入することをさらに含む、

請求項 1 5 の方法。

## 【請求項 1 7】

複数のシェーダを含むグラフィックスパイプラインと、

テキストチャブロックと、前記テキストチャブロックの圧縮パラメータと前記テキストチャブロックがストレージリソースに常駐しているかどうかを示す常駐状態とを符号化するメタデータと、を記憶するストレージリソースと、を備え、

前記複数のシェーダのうち何れかのシェーダは、シェーディング操作を実行するために前記テキストチャブロックに要求すると共に前記メタデータにアクセスするように構成されており、

前記テキストチャブロック内のデータは、前記テキストチャブロックが前記ストレージリソースに常駐していることを前記常駐状態が示すかどうかに応じて、選択的に前記シェーダに返される、

グラフィックス処理ユニット。

## 【請求項 1 8】

前記テキストチャブロック内の前記データは、前記テキストチャブロックが前記ストレージリソースに常駐していることを前記常駐状態が示すことに応じて、前記シェーダに返される、請求項 1 7 のグラフィックス処理ユニット。

## 【請求項 1 9】

前記シェーダは、前記テキストチャブロックが前記ストレージリソースに常駐していないことを前記常駐状態が示すことに応じて、前記要求されたデータが利用可能でないことを示す信号を受信するように構成されている、

請求項 1 7 のグラフィックス処理ユニット。

## 【請求項 2 0】

10

20

30

40

50

前記シェーダは、前記信号を受信したことに応じて、データを前記テクスチャブロックに投入するように構成されており、

前記シェーダは、前記データを前記テクスチャブロックに投入したことに応じて、前記テクスチャブロックを再要求すると共に前記メタデータに再アクセスする、

請求項 19 のグラフィックス処理ユニット。

【発明の詳細な説明】

【背景技術】

【0001】

グラフィックス処理ユニット（GPU）は、通常、プログラマブルシェーダ及び固定機能ハードウェアブロックのシーケンスから形成されるグラフィックスパイプラインを使用して、三次元（3D）グラフィックスを処理する。例えば、フレーム内で可視であるオブジェクトの3Dモデルは、ユーザに表示するための画素の値を生成するように、グラフィックスパイプラインで処理される三角形、他の多角形又はパッチのセットによって表すことができる。三角形、他の多角形又はパッチは、まとめてプリミティブと呼ばれる。このプロセスは、プリミティブの解像度よりも高い解像度を有する視覚的な詳細を組み込むために、テクスチャをプリミティブにマッピングすることを含む。GPUは、テクスチャ値を記憶するために使用される専用メモリを備えているため、テクスチャ値は、グラフィックスパイプラインで処理されているプリミティブへのマッピングするに利用可能である。テクスチャは、ディスクに記憶することもできるし、グラフィックスパイプラインが必要とする場合にプロセッサに従って生成することができる。専用GPUメモリに記憶されたテクスチャデータは、テクスチャをディスクからロードするか、データをプロセッサに従って生成することによって取り込まれる。通常、専用GPUメモリは、比較的小さいメモリであり、専用GPUメモリに記憶可能なテクスチャデータの量を制限する。さらに、テクスチャデータを取り込むために必要なオーバーヘッドは、特に、最終的なスクリーン画像をレンダリングするためにテクスチャデータの小さなサブセットのみが使用される場合に、大きくなる可能性がある。例えば、ページ内のデータの僅かな部分のみを使用して画像をレンダリングする場合であっても、テクスチャは、ページ毎にディスクからロードされる。

【0002】

添付図面を参照することによって本開示をより良く理解することができ、その多数の特徴及び利点が当業者に明らかになる。異なる図面での同じ符号の使用は、類似又は同一のアイテムを示す。

【図面の簡単な説明】

【0003】

【図1】いくつかの実施形態による、ディスプレイへの出力を意図した視覚画像を生成するグラフィックス処理ユニット（GPU）を含む処理システムのブロック図である。

【図2】いくつかの実施形態による、三次元（3D）シーンのラスター化画像を所定の解像度で生成するために高次ジオメトリプリミティブを処理することができるグラフィックスパイプラインを示す図である。

【図3】いくつかの実施形態による、テクスチャブロック及び対応するメタデータを記憶するために使用されるメモリ構造のブロック図である。

【図4】いくつかの実施形態による、テクスチャデータを含むテクスチャブロックの常駐状態に基づいてテクスチャデータを選択的に返す方法のフロー図である。

【図5】いくつかの実施形態による、GPUメモリに関連するメタデータキャッシュを操作する方法のフロー図である。

【図6】いくつかの実施形態による、GPUメモリ及び関連するキャッシュにテクスチャデータを投入する方法のフロー図である。

【図7】いくつかの実施形態による、テクスチャブロックに関連するメタデータ面のブロック図である。

【図8】いくつかの実施形態による、メタデータ面をサンプリングすることによって非常

10

20

30

40

50

駐タイルのワークリストを生成する方法のフロー図である。

【図 9】いくつかの実施形態による、テクスチャデータを、非常駐の、以前にサンプリングされたタイルに投入する方法のフロー図である。

【発明を実施するための形態】

【0004】

テクスチャのサブセットを投入し、メモリ内に常駐するテクスチャの部分を示す常駐構造を使用することによって、専用GPUメモリ内のスペースを節約することができる。次に、テクスチャデータにアクセスする前に常駐構造にクエリして、テクスチャデータが専用GPUメモリに常駐していることを確認することができる。したがって、GPUは、テクスチャデータにアクセスする要求に応じて、専用GPUメモリに未だ常駐していないテクスチャの部分投入することのみが要求される。したがって、常駐構造を実装すると、専用GPUメモリに記憶された各テクスチャを完全に投入する必要がなくなるので、パフォーマンスを向上させることができる。しかしながら、ハードウェアアクセラレーションによる常駐チェックは、仮想メモリアドレス変換テーブルを用いて実装されているので、常駐チェックの粒度は、仮想メモリページの粒度（通常、64kB）によって決定される。各仮想メモリページ内のテクスチャデータのごく一部のみが、通常、グラフィックスパイプラインによって使用され、テクスチャデータを各仮想メモリページに投入するために大きなオーバーヘッドを招くハードウェアアクセラレーションされた常駐チェックが生じる。ソフトウェアベースの常駐チェックは、任意の小さな粒度を実装することができるが、ソフトウェアベースの常駐チェックを実装するアプリケーションは、全てのテクスチャサンプリング操作に対して常駐構造のトラバーサル（アプリケーションと常駐構造との間の少なくとも1つのコマンド/応答交換を含む）を実行する必要がある。したがって、レンダリングされた画素毎に常駐構造をトラバースするために、ソフトウェアベースの常駐チェックが必要になる可能性があり、これは、グラフィックスパイプラインのパフォーマンスを低下させ、レイテンシを増加させる。

【0005】

テクスチャブロックの圧縮パラメータ及びこのテクスチャブロックの常駐状態を符号化するメタデータを記憶することによって、任意の粒度のテクスチャブロックの常駐チェックをハードウェアにおいて、したがって、ソフトウェアベースの常駐チェックのコマンド/応答オーバーヘッドなしに実行することができる。メタデータは、GPUによって実装されるグラフィックスパイプラインによるアクセスのためにテクスチャブロックを記憶するのに使用されるGPUメモリに記憶される。グラフィックスパイプライン内のシェーダは、テクスチャブロックに対してシェーディング操作を実行するように要求すると共に、テクスチャブロックのメタデータにアクセスする。例えば、テクスチャブロックに要求すると同時にメタデータにアクセスすることができる。テクスチャブロックがGPUメモリに常駐していることをメタデータが示す場合に、テクスチャブロック内のデータに対する要求がシェーダに返される。したがって、テクスチャブロックがGPUメモリに常駐している場合に、常駐チェックによって追加のオーバーヘッドが生じない。テクスチャブロックがGPUメモリに常駐していないことをメタデータが示す場合に、要求されたデータがGPUメモリに常駐していないことを示す信号が、シェーダにフィードバックされる。このシェーダは、フィードバックの受信に応じて、（例えば、テクスチャブロックをディスクからロードするか、テクスチャブロックのデータをプロシージャに従って生成することによって）非常駐テクスチャブロックに投入することができる。次に、シェーダは、テクスチャブロックと共に（例えば、同時に）メタデータに再アクセスすることができる。テクスチャブロックがGPUメモリに常駐していない場合に常駐チェックを実行するのに必要な追加のオーバーヘッドは、サンプリング操作のフローにおいて、例えばテクスチャブロックにアクセスしようとすると共に又は同時に実行されるので、最小限である。さらに、従来のテクスチャ要求は、要求されたテクスチャの圧縮率を示すメタデータにアクセスする必要があるので、追加のオーバーヘッドは、従来のテクスチャ要求によって生じるオーバーヘッドと比較して最小限である。

10

20

30

40

50

## 【 0 0 0 6 】

いくつかの実施形態において、メタデータキャッシュは、シェーダが頻繁にアクセスするテクスチャブロックのメタデータを記憶する。メタデータキャッシュは、関連するテクスチャブロックがGPUメモリに常駐していなかったことを、以前の常駐チェック中にシェーダによってアクセスされたメタデータが示しているかどうかを示す情報を含むことができる。メタデータキャッシュのいくつかの実施形態は、テクスチャブロックの部分（例えば、タイル又はキャッシュライン）にアクセスするためのいくつかのタイプの要求に応じて、常駐状態を変更する。例えば、メタデータキャッシュから非常駐タイルのメタデータにアクセスしようとするテクスチャブロックフェッチ操作に応じて、タイルの常駐状態を「非常駐、サンプリング」に変更することができる。したがって、シェーダによって実行されたサンプリング操作に関わる非常駐タイルのメタデータは、キャッシュされたメタデータにおいてマークされ、メタデータキャッシュが、変更されたメタデータをエビクトしてグローバルメモリに返す場合に、最終的にメモリに書き戻される。例えば、アプリケーションが将来、非常駐のサンプリングされたタイルにアクセスすることが期待される場合に、アプリケーションは、メタデータを読み出し、非常駐のサンプリングされたタイルを特定し、例えば、アプリケーションが、将来、非常駐のサンプリングされたタイルにアクセスすることを期待する場合に、有効なデータを非常駐のサンプリングされたタイルに投入することができる。別の例では、テクスチャデータがタイルに書き込まれることに伴って、タイルの常駐状態を「常駐」に変更することができる。いくつかの実施形態では、タイルに書き込まれる圧縮された有効なデータの圧縮率を示すのに使用されるメタデータ値は、このタイルの常駐状態を表すのに使用される。例えば、メタデータ値を、メタデータ値の第1サブセットが、圧縮された有効なデータの実際の圧縮率を示し、常駐状態が「常駐」であることを示すように、符号化することができる。メタデータの第2サブセットは、予約されたビットパターンを使用して、常駐状態が「非常駐、サンプリング」であることを示す。

## 【 0 0 0 7 】

図1は、いくつかの実施形態による、ディスプレイ110への出力を意図した視覚画像を生成するグラフィックス処理ユニット（GPU）105を含む処理システム100のブロック図である。処理システム100は、メモリ115を含む。メモリ115のいくつかの実施形態は、ダイナミックランダムアクセスメモリ（DRAM）として実装される。しかしながら、メモリ115は、スタティックランダムアクセスメモリ（SRAM）、不揮発性RAM等を含む他のタイプのメモリを使用して実装されてもよい。図示した実施形態では、GPU105は、バス120を介してメモリ115と通信する。しかしながら、GPU105のいくつかの実施形態は、直接接続を介して又は他のバス、ブリッジ、スイッチ、ルータ等を介してメモリ115と通信する。GPU105は、メモリ115に記憶された命令を実行することができ、実行された命令の結果等の情報をメモリ115に記憶することができる。例えば、メモリ115は、GPU105によって実行されるプログラムコードからの命令のコピー125を記憶することができる。GPU105のいくつかの実施形態は、命令を同時に又は並列に独立して実行することができる複数のプロセッサコア（明確さのために示されていない）を含む。

## 【 0 0 0 8 】

処理システム100は、命令を実行する中央処理装置（CPU）130を含む。CPU130のいくつかの実施形態は、命令を同時に又は並列に独立して実行することができる複数のプロセッサコア（明確さのために示されていない）を含む。また、CPU130は、バス120に接続されているので、バス120を介してGPU105及びメモリ115と通信することができる。CPU130は、メモリ115に記憶されたプログラムコード135等の命令を実行することができ、実行された命令の結果等の情報をメモリ115に記憶することができる。また、CPU130は、ドローコールをGPU105に発行することによって、グラフィックス処理を開始することができる。ドローコールは、CPU130によって生成され、GPU105に送信され、GPU105がフレーム内のオブジェ

10

20

30

40

50

クト（又は、オブジェクトの一部）をレンダリングすることを指示するコマンドである。ドローコールのいくつかの実施形態は、GPU 105 がオブジェクト又はその一部をレンダリングするために使用するテクスチャ、状態、シェーダ、レンダリングオブジェクト、バッファ等を定義する情報を含む。ドローコールに含まれる情報は、状態情報を含む状態ベクトルと呼ぶことができる。GPU 105 は、オブジェクトをレンダリングして、ディスプレイ 110 に提供される画素値を生成し、ディスプレイは、画素値を使用して、レンダリングされたオブジェクトを表す画像を表示する。

#### 【0009】

入出力（I/O）エンジン 140 は、キーボード、マウス、プリンタ、外部ディスク等の処理システム 100 の他の要素と同様に、ディスプレイ 110 に関連する入出力動作を処理する。I/O エンジン 140 は、バス 120 に接続されており、GPU 105、メモリ 115 又は CPU 130 と通信することができる。図示した実施形態では、I/O エンジン 140 は、コンパクトディスク（CD）、デジタルビデオディスク（DVD）等の外部記憶媒体 145 に記憶された情報を読み出すように構成されている。

#### 【0010】

外部記憶媒体 145 は、ビデオゲーム等のアプリケーションを実装するのに使用されるプログラムコードを表す情報を記憶する。外部記憶媒体 145 のプログラムコードは、メモリ 115 に書き込まれ、GPU 105 又は CPU 130 によって実行される命令のコピー 125 を形成することができる。また、外部記憶媒体 145 は、ディスプレイ 110 上に提示される画像をレンダリングするのに使用されるテクスチャを表す情報を記憶する。外部記憶媒体 145 に記憶されるテクスチャの一部は、メモリ 115 に書き込まれ、テクスチャ情報 150 として記憶される。例えば、テクスチャ情報 150 は、テクスチャブロックと、テクスチャ情報 150 内のテクスチャブロックに適用される圧縮の度合いを示す対応する圧縮メタデータと、を含むことができる。

#### 【0011】

GPU 105 は、ドローコールに応じて異なるプリミティブ又はピンの同時処理のために構成された複数のステージを含むグラフィックスパイプライン（明確さのために図 1 に示されていない）を実装する。GPU 105 内のグラフィックスパイプラインのステージは、ビデオゲーム等のアプリケーションによって生成される異なるプリミティブを同時に処理することができる。プリミティブの処理は、例えばさらなる詳細を提供するために、テクスチャをプリミティブにマッピングすることを含む。GPU 105 は、メモリ 115 に記憶されたテクスチャ情報 150 からのテクスチャデータにアクセスする。しかしながら、本明細書で説明するように、全てのテクスチャブロックに対するテクスチャデータは、GPU 105 がテクスチャデータへのアクセスを要求する前に必ずしも投入されない。例えば、テクスチャデータは、外部記憶媒体 145 からメモリ 115 に書き込まれていなくてもよい。別の例では、CPU 130 は、プログラムコード 135 内の命令によって、テクスチャデータをプロシージャに従って生成しなくてもよい。

#### 【0012】

GPU 105 は、テクスチャ情報 150 に記憶されたテクスチャブロックへのアクセスを要求すると共にメタデータのクエリを行うことによって、要求されたテクスチャブロックのテクスチャデータがテクスチャ情報 150 において利用可能であるかどうかを判別する。本明細書に使用される場合に、「と共に」という用語は、GPU 105 が、テクスチャブロックにアクセスする要求毎にメタデータのクエリを発行することを示す。例えば、GPU 105 は、テクスチャブロックへのアクセスを要求すると同時にメタデータのクエリを行うことができる。別の例では、GPU 105 は、テクスチャブロックへのアクセスを要求する前にメタデータのクエリを行うことができ、これにより、テクスチャブロックの常駐状態を判別することができ、必要に応じて、アクセス要求の前にテクスチャブロックを投入することができる。

#### 【0013】

メタデータは、テクスチャブロックの圧縮率を符号化することに加えて、テクスチャブ

10

20

30

40

50

ロックの常駐状態を符号化する。例えば、メタデータは、テクスチャブロック内のテクスチャデータが投入されているかどうかを示す情報を符号化するので、テクスチャマッピング操作のためにGPU105に利用可能である。テクスチャブロックがメモリ115に常駐していることをメタデータが示すかどうかによって、テクスチャブロック内のデータがGPU105に選択的に返される。例えば、メモリ115は、テクスチャブロックがメモリ115に常駐していることを示すメタデータに応じて、テクスチャブロック内のデータを返す。別の例では、テクスチャブロックがメモリ115に常駐していないことを示すメタデータに応じて、要求されたデータが利用可能ではないことを示す信号がGPU105に返される。次に、GPU105は、外部記憶媒体145からメモリ115にデータを書き込ませることによって、データをプロシージャに従って生成してメモリ115に記憶することによって、又は、CPU130にデータをプロシージャに従って生成させてメモリ115に記憶させることによって、信号の受信に応じてデータをテクスチャブロックに投入することができる。GPU105は、データをテクスチャブロックに投入することに応じてテクスチャブロックを再要求すると共に（例えば、再要求と同時に）、メタデータに再アクセスすることができる。また、本明細書で説明するように、テクスチャデータ及びメタデータのキャッシングを実行することができる。

10

#### 【0014】

図2は、いくつかの実施形態による、三次元(3D)シーンのラスターライズ画像を所定の解像度で生成するために高次ジオメトリプリミティブを処理することができるグラフィックスパイプライン200を示す図である。グラフィックスパイプライン200は、図1に示すGPU105のいくつかの実施形態に実装される。グラフィックスパイプライン200は、バッファを実装し、頂点データ、テクスチャデータ等を記憶するのに使用される1つ以上のメモリ又はキャッシュの階層等のストレージリソース201にアクセスする。図1に示すメモリ115のいくつかの実施形態を使用して、ストレージリソース201を実装することができる。

20

#### 【0015】

入力アセンブラ202は、シーンのモデルの一部を表すオブジェクトを定義するために使用されるストレージリソース201からの情報にアクセスするように構成されている。頂点シェーダ203は、ソフトウェアで実装することができ、プリミティブの単一の頂点を入力として論理的に受信し、単一の頂点を出力する。頂点シェーダ203等のシェーダのいくつかの実施形態は、複数の頂点を同時に処理できるように、大規模な単一命令複数データ(SIMD)処理を実施する。図2に示すグラフィックスパイプライン200は、グラフィックスパイプライン200に含まれる全てのシェーダが、共有された大規模なSIMD計算ユニット上で同一の実行プラットフォームを含むように、統合型シェーダモデルを実装する。したがって、本明細書において統合型シェーダプール204と呼ばれる共通のリソースセットを使用して、頂点シェーダ203を含むシェーダが実装される。図1に示すGPU105内のプロセッサを使用して、統合型シェーダプール204のいくつかの実施形態が実装される。

30

#### 【0016】

ハルシェーダ205は、入力パッチを定義するのに使用される入力高次パッチ(input high-order patches)又は制御点に対して動作する。ハルシェーダ205は、テッセレーション係数及び他のパッチデータを出力する。ハルシェーダ205によって生成されたプリミティブを、オプションでテッセレータ206に提供することができる。テッセレータ206は、オブジェクト(パッチ等)をハルシェーダ205から受信し、例えば、ハルシェーダ205によってテッセレータ206に提供されたテッセレーション係数に基づいて入力オブジェクトをテッセレーションすることによって、入力オブジェクトに対応するプリミティブを識別する情報を生成する。テッセレーションは、例えば、テッセレーションプロセスによって生成されるプリミティブの粒度を指定するテッセレーション係数によって示されるように、パッチ等の入力高次プリミティブを、より詳細なレベルを表す低次出力プリミティブのセットに細分化する。したがって、シーンのモデルを、(メモリ又は

40

50



帯域幅を節約するために)少数の高次プリミティブによって表すことができ、高次プリミティブをデッセレーションすることによってさらなる詳細を追加することができる。

#### 【0017】

ドメインシェーダ207は、ドメイン位置及び(オプションで)他のパッチデータを入力する。ドメインシェーダ207は、提供された情報に対して動作し、入力ドメイン位置及び他の情報に基づいて出力用の単一の頂点を生成する。ジオメトリシェーダ208は、入力プリミティブを受信し、入力プリミティブに基づいてジオメトリシェーダ208によって生成される最大4つのプリミティブを出力する。プリミティブの1つのストリームがラスタライザ209に提供され、プリミティブの最大4つのストリームを、ストレージリソース201のバッファにおいて連結することができる。ラスタライザ209は、シェーディング操作や、クリッピング、パースペクティブ分割、シザリング及びビューポート選択等の他の操作を実行する。画素シェーダ210は、画素フローを入力し、入力画素フローに応じて0又は別の画素フローを出力する。出力マージブロック211は、画素シェーダ210から受信した画素に対してブレンド、深度、ステンシル又は他の動作を実行する。

#### 【0018】

グラフィックスパイプライン200内のいくつかの又は全てのシェーダは、ストレージリソース201に記憶されているテクスチャデータを使用してテクスチャマッピングを実行することができる。例えば、画素シェーダ210は、テクスチャデータをストレージリソース201から読み出し、このテクスチャデータを使用して1つ以上の画素をシェーディングすることができる。次に、シェーディングされた画素は、ユーザに提示するためにディスプレイ(図1に示すディスプレイ110等)に提供される。しかしながら、本明細書で説明するように、テクスチャデータは、グラフィックスパイプライン200内のシェーダによってテクスチャデータが必要とされる前に、必ずしもストレージリソース201に投入されない。したがって、画素シェーダ210等のシェーダは、シェーディング操作を実行することをテクスチャブロックに要求すると共に(例えば、要求すると同時に)、メタデータにアクセスするように構成されている。メタデータは、テクスチャブロックの圧縮率及びこのテクスチャブロックの常駐状態を符号化する。テクスチャブロック内のデータは、テクスチャブロックがストレージリソース201に常駐していることをメタデータが示すかどうかに応じて、シェーダに選択的に返される。

#### 【0019】

図3は、いくつかの実施形態による、テクスチャブロック及び対応するメタデータを記憶するために使用されるメモリ構造300のブロック図である。メモリ構造300は、図1に示す処理システム100及び図2に示すグラフィックスパイプライン200のいくつかの実施形態に実装される。メモリ構造300は、テクスチャブロック310(明確さのために符号によって1つのみを示す)を記憶するのに使用されるGPUメモリ305を含む。図1に示すメモリ115及び図2に示すストレージリソース201のいくつかの実施形態を使用して、GPUメモリ305を実装する。

#### 【0020】

図示した実施形態では、テクスチャブロック310は、テクスチャ情報のページ315のサブセットである。例えば、ページ315を、4~64kBの範囲内の値に設定することができる仮想マシンページ粒度で実装することができる。テクスチャブロック310は、ページ315の粒度の1/8、1/16又は1/32等の任意の粒度で構成されてもよい。例えば、各テクスチャブロック310は、キャッシュライン、タイル、又は、仮想マシンページの他の区画を表すことができる。また、GPUメモリ305は、対応するテクスチャブロック310のメタデータ320(明確さのために符号によって1つのみを示す)を記憶する。メタデータ320は、対応するテクスチャブロック310内の情報を圧縮するために使用される圧縮率を符号化する。例えば、テクスチャブロック310内のテクスチャデータが7つの圧縮率のセットのうち何れかで圧縮され得る場合には、3つのビットを使用して圧縮率の値を符号化することができる。

10

20

30

40

50

## 【 0 0 2 1 】

全てのテクスチャブロック 3 1 0 に有効なテクスチャデータが投入されるとは限らない。図示した実施形態では、有効なテクスチャデータが投入されたテクスチャブロック 3 1 0 をハッチングされたボックスで示し、有効なテクスチャデータが投入されていないテクスチャブロック 3 1 0 を白いボックスで示している。したがって、メタデータ 3 2 0 は、対応するテクスチャブロック 3 1 0 について有効なテクスチャデータが G P U メモリ 3 0 5 に常駐しているかどうかを示す常駐状態を含む。図示した実施形態では、有効なテクスチャデータが G P U メモリ 3 0 5 に常駐していることを示す符号化情報を含むメタデータ 3 2 0 を斜線付きのボックスで示し、有効なテクスチャデータが G P U メモリ 3 0 5 に常駐していないことを示す符号化情報を含むメタデータ 3 2 0 を白いボックスで示している。メタデータ 3 2 0 は、対応するテクスチャブロック 3 1 0 内のテクスチャデータが G P U メモリ 3 0 5 に常駐しているかどうかを示すために、異なる値に設定された別のビットを含むことができる。例えば、このビットを 1 の値に設定して、テクスチャデータが常駐していることを示すことができ、このビットを 0 の値に設定して、テクスチャデータが常駐していないことを示すことができる。また、メタデータ 3 2 0 は、圧縮率と組み合わせて常駐状態を符号化することができる。例えば、7 つの可能な圧縮率を符号化するのに使用される 3 つのビットは、圧縮率を示す 7 つの値が、テクスチャブロック 3 1 0 のテクスチャデータが G P U メモリ 3 0 5 に常駐していることも示す場合に、常駐状態を符号化するために使用されてもよい。8 番目の値は、テクスチャブロック 3 1 0 のテクスチャデータが G P U メモリ 3 0 5 に常駐していないことを示す。

10

20

## 【 0 0 2 2 】

メタデータ 3 2 0 のいくつかの実施形態は、メタデータ 3 2 0 のサンプリング状態 3 2 5 を示す情報を記憶（又は符号化）する。サンプリング状態 3 2 5 は、メタデータ 3 2 0 に関連するテクスチャブロック 3 1 0 の以前の常駐チェックが、テクスチャブロック 3 1 0 が G P U メモリ 3 0 5 に常駐していなかったことを示していたかどうかを示す。例えば、0 に等しいサンプリング状態 3 2 5 の値は、テクスチャブロック 3 1 0 が G P U メモリ 3 0 5 に常駐していなかったことを示していると以前の常駐チェックが示すものであり、1 に等しい値は、テクスチャブロック 3 1 0 が G P U メモリ 3 0 5 に常駐していなかったことを示していると少なくとも 1 つの以前の常駐チェックが示すものである。サンプリング状態 3 2 5 を使用して、ロード又は生成されたデータが投入されるタイルについてアプリケーションが生成するタスクリスト内の重複したエントリを排除することができる。サンプリング状態 3 2 5 は、図 3 においてビットの値によって示されているが、いくつかの実施形態は、メタデータ 3 2 0 を表すために使用されるビットの値を使用して、サンプリング状態 3 2 5 を符号化することができる。

30

## 【 0 0 2 3 】

メモリ構造 3 0 0 は、頻繁にアクセスされるメタデータ 3 3 0 を記憶するのに使用されるメタデータキャッシュ 3 3 0 を含む。キャッシュされたメタデータ 3 3 5 を、キャッシュ置換ポリシーに従って、メタデータキャッシュ 3 3 0 に追加したり、メタデータキャッシュ 3 3 0 からエビクトすることができる。メタデータ 3 3 0 は、対応するテクスチャブロック 3 1 0 の常駐状態を示し、例えば、ハッチングは、テクスチャデータが G P U メモリ 3 0 5 に常駐していることを示す。また、メタデータキャッシュは、キャッシュされたメタデータ 3 3 5 に対応するサンプリング状態 3 2 5 を記憶する。メタデータ 3 3 0 に関するサンプリング状態 3 2 5 を、キャッシュされたメタデータ 3 3 5 へのアクセス要求に応じて変更することができる。例えば、サンプリング状態 3 4 0 及びメタデータ 3 3 5 がメタデータキャッシュ 3 3 0 に追加されている場合に、サンプリング状態 3 2 5 の値（又は符号化）が「サンプリングされていない」場合には、メタデータ 3 3 5 に対する最初のヒットに応じて、サンプリング状態 3 2 5 を「サンプリングされた」値（又は符号化）に変更する。サンプリング状態 3 2 5 は、対応するテクスチャブロック 3 1 0 のテクスチャデータが G P U メモリ 3 0 5 に常駐していないままである場合に、後続のヘッドに応じて「サンプリングされた」値（又は符号化）を保持する。いくつかの実施形態では、「サン

40

50

プリングされた」値を含むサンプリング状態と、テクスチャブロック 310 のテクスチャデータが G P U メモリ 305 に常駐していないことを示す常駐状態とに応じて、テクスチャブロック 310 のテクスチャデータが投入される。したがって、投入されたテクスチャブロック 310 のテクスチャデータは、後続のアクセス要求に利用可能である。サンプリング状態は「サンプリングされていない」値に設定され、常駐状態は、テクスチャブロック 310 のテクスチャデータの投入に応じて、常駐に設定される。

#### 【0024】

いくつかの実施形態のメモリ構造 300 は、頻繁にアクセスされるテクスチャブロック 345（明確さのために符号によって 1 つのみを示す）を記憶する L2 キャッシュ 340 等のキャッシュを含む。キャッシュされたテクスチャブロック 345 は、キャッシュ置換ポリシーに従って、L2 キャッシュ 340 に追加されるか、L2 キャッシュ 340 からエビクトされる。例えば、L2 キャッシュ 340 によって、最近最も使用されていない（least-recently-used）キャッシュ置換ポリシーを実装することができる。図示した実施形態では、クロスハッチングで示すように有効なテクスチャデータが投入され、これにより、L2 キャッシュ 340 は、キャッシュヒットに応じて有効なテクスチャデータを返すことができる。

10

#### 【0025】

シェーダ 350 等のシェーダは、パイプライン 355 を介して対応するテクスチャブロックにアクセスすると共に、メタデータにアクセスすることができる。例えば、シェーダ 350 は、対応するテクスチャブロックにアクセスすると同時にメタデータにアクセスすることができる。図示した実施形態では、シェーダ 350 は、テクスチャ要求をパイプライン 355 に送り、パイプラインは、このテクスチャ要求を 1 つ以上の同時テクスチャデータ及びメタデータ要求に変換する。例えば、パイプライン 355 は、タイルのメタデータアドレスと、タイル内のテクスチャデータのデータアドレスとの両方を含む要求を生成することができる。パイプライン 355 のいくつかの実施形態は、L2 キャッシュ 340 及び G P U メモリ 305 にも記憶される、頻繁にアクセスされるテクスチャデータのコピーを記憶する内部データキャッシュを実装する。したがって、パイプライン 355 は、その内部データキャッシュに記憶されたテクスチャデータを返すことによって、いくつかのテクスチャデータ要求に応答することができる。

20

#### 【0026】

パイプライン 355 は、テクスチャデータの要求を L2 キャッシュ 340 に送るように構成されている。この要求が L2 キャッシュ 340 においてヒットした場合に、要求されたテクスチャデータが L2 キャッシュ 340 からパイプライン 355 に返される。この要求が L2 キャッシュ 340 においてミスした場合に、この要求が G P U メモリ 305 に転送される。テクスチャデータの常駐状態のクエリがメタデータキャッシュ 330 に送られる。このクエリがメタデータキャッシュ 330 においてヒットした場合に、メタデータキャッシュ 330 は、メタデータ 330 をパイプライン 355 に返し、このパイプラインは、メタデータ 330 を使用して、常住状態（例えば、常駐又は非常駐等）と、テクスチャデータのサンプリング状態 325（例えば、以前にサンプリングされたか否か等）とを判別する。本明細書で説明するように、常住状態又はサンプリング状態を、クエリに応じて変更することができる。このクエリがメタデータキャッシュ 330 においてミスした場合に、クエリが G P U メモリ 305 に転送される。

30

40

#### 【0027】

G P U メモリ 305 は、テクスチャブロック 310 の常駐状態についてのクエリを受信したことに応じて、符号化されたメタデータ 320 をパイプライン 355 に返し、このパイプラインは、符号化されたメタデータ 320 を使用して、テクスチャブロック 310 のテクスチャデータが G P U メモリ 305 に常駐しているか否かを判別する。G P U メモリ 305 は、テクスチャデータが常駐している場合に、テクスチャブロック 310 内のテクスチャデータに対する要求に応じて、要求されたテクスチャデータをパイプライン 355 に返す。

50

## 【 0 0 2 8 】

パイプライン 3 5 5 は、要求されたテクスチャデータ及び要求されたメタデータを、メタデータキャッシュ 3 3 0 又は GPU メモリ 3 0 5 から受信する。パイプライン 3 5 5 は、この情報を変換し、変換された情報をシェーダ 3 5 0 に返すことができる。例えば、パイプライン 3 5 5 は、メタデータを使用して、要求されたテクスチャデータが GPU メモリ 3 0 5 に常駐しているかどうかを判別する。常駐している場合、パイプライン 3 5 5 は、要求されたテクスチャデータをシェーダ 3 5 0 に返す。要求されたテクスチャデータが GPU メモリ 3 0 5 に常駐していないことをメタデータが示す場合、パイプライン 3 5 5 は、要求されたテクスチャデータが GPU メモリ 3 0 5 に常駐していないことを示す信号を返す。要求されたテクスチャデータが返された場合、シェーダ 3 5 0 は、要求されたテクスチャデータを用いて処理を進める。要求されたテクスチャデータが GPU メモリ 3 0 5 に常駐していないことを示す信号をシェーダ 3 5 0 が受信した場合、シェーダ 3 5 0 は、この信号を受信したことに応じて、有効なテクスチャデータをテクスチャブロック 3 1 0 に投入する命令を発行することができる。その後、シェーダ 3 5 0 は、テクスチャデータの要求、及び、テクスチャブロック 3 1 0 に対するメタデータのクエリを再度送ることができる。

10

## 【 0 0 2 9 】

図 4 は、いくつかの実施形態による、テクスチャデータを含むテクスチャブロックの常駐状態に基づいてテクスチャデータを選択的に返す方法 4 0 0 のフロー図である。方法は、図 1 に示す処理システム 1 0 0、図 2 に示すグラフィックスパイプライン 2 0 0、及び、図 3 に示すメモリ構造 3 0 0 のいくつかの実施形態において実装される。

20

## 【 0 0 3 0 】

ブロック 4 0 5 において、シェーダは、メモリに記憶されるテクスチャブロックのメタデータ及びテクスチャデータに同時にアクセスする要求を送る。本明細書で説明するように、テクスチャデータは、シェーダが要求を送る前に必ずしも投入されない。したがって、メタデータは、テクスチャデータの常駐状態を符号化して、テクスチャデータがメモリに常駐しているか否かを示す。したがって、メタデータアクセスは、テクスチャデータをキャッシュ又はメモリから取得するのに使用されるテクスチャデータにアクセスする要求と同時に、テクスチャデータの常駐状態を決定するために使用される。また、メタデータは、テクスチャデータの圧縮率を決定するために使用される。

30

## 【 0 0 3 1 】

判別ブロック 4 1 0 において、テクスチャブロックのテクスチャデータの常駐状態が決定される。例えば、符号化されたメタデータが常駐テクスチャデータに対応する値を含む場合に、テクスチャデータの常駐状態は、「常駐」と判別される。符号化されたメタデータが、テクスチャデータがメモリに常駐していないことを示す値を含む場合に、テクスチャデータの常駐状態は、「非常駐」と判別される。テクスチャデータがメモリに常駐している場合、方法はブロック 4 1 5 に進む。テクスチャデータがメモリに常駐していない場合、方法はブロック 4 2 0 に進む。

## 【 0 0 3 2 】

ブロック 4 1 5 において、要求されたテクスチャデータがシェーダに返される。本明細書で説明するように、常駐テクスチャデータを、図 3 に示す GPU メモリ 3 0 5 等のメモリ又は L 2 キャッシュ 3 4 0 等の対応するキャッシュからシェーダに返すことができる。

40

## 【 0 0 3 3 】

ブロック 4 2 0 において、要求されたテクスチャデータがメモリに常駐していないことを示す信号が、シェーダに返される。シェーダは、この信号を受信したことに応じて、テクスチャブロックに要求されたテクスチャデータを投入するための命令を発行する（ブロック 4 2 5）。次に、方法 4 0 0 はブロック 4 0 5 に進み、シェーダは、テクスチャブロック内のメタデータ及びテクスチャデータにアクセスするために同時要求を再度送る。テクスチャデータは、この時点でメモリに常駐する必要がある。しかしながら、テクスチャデータが未だメモリに常駐していない場合に、方法 4 0 0 は、シェーダが、要求されたテ

50

クスチャデータをメモリから正常に取得するまで、繰り返すことができる。

【 0 0 3 4 】

図 5 は、いくつかの実施形態による、GPUメモリに関連するメタデータキャッシュを操作する方法 5 0 0 のフロー図である。方法 5 0 0 は、図 3 に示すメタデータキャッシュ 3 3 0 及び GPUメモリ 3 0 5 のいくつかの実施形態において実施される。

【 0 0 3 5 】

ブロック 5 0 5 において、テクスチャブロックのメタデータへのアクセス要求がメタデータキャッシュに発行される。例えば、この要求は、タイルのメタデータアドレスを含むことができる。判別ブロック 5 1 0 において、メタデータキャッシュは、例えば、メタデータアドレスの一部をメタデータキャッシュ内のタグアレイと比較することによって、この要求がメタデータキャッシュ内でヒットするかどうかを判別する。この要求がメタデータキャッシュ内でヒットした場合、方法は、判別ブロック 5 1 5 に進む。この要求がメタデータキャッシュ内でミスした場合、方法は、ブロック 5 2 0 に進む。

10

【 0 0 3 6 】

ブロック 5 2 0 において、メタデータに対する要求が GPUメモリに転送され、要求されたメタデータが GPUメモリからアクセスされる。本明細書で説明するように、要求されたメタデータを使用して、対応するテクスチャデータの圧縮率及び常駐状態を決定することができる。ブロック 5 2 5 において、GPUメモリから取得されたメタデータは、キャッシュ置換ポリシーに従って、メタデータキャッシュにキャッシュされる。例えば、最近最も使用されていないキャッシュエントリをメタデータキャッシュからエビクトし、取得されたメタデータに置き換えることができる。

20

【 0 0 3 7 】

判別ブロック 5 1 5 において、メタデータを使用して、要求されたテクスチャデータが GPUメモリに常駐しているかどうかを判別する。例えば、図 3 に示すパイプライン 3 5 5 等のパイプラインは、要求されたテクスチャデータが常駐しているか、常駐していないかを判別するように、メタデータを変換することができる。テクスチャデータがメモリに常駐していない場合に、方法 5 0 0 は、ブロック 5 3 0 に進む。テクスチャデータがメモリに常駐している場合に、方法 5 0 0 は、ブロック 5 3 5 に進む。

【 0 0 3 8 】

ブロック 5 3 0 において、キャッシュされたメタデータのサンプリング状態が「サンプリング」に変更され、以前にキャッシュされたメタデータが他のアクセス要求によってサンプリングされたことを示す。いくつかの実施形態では、キャッシュされたメタデータのサンプリング状態を使用して、タスクリストの重複を排除することができる。次に、方法 5 0 0 は、ブロック 5 3 0 からブロック 5 3 5 に進む。ブロック 5 3 5 において、キャッシュされたメタデータ値が返される。例えば、キャッシュされたメタデータ値をパイプラインに返すことができる。パイプラインは、キャッシュされたメタデータ値を変換し、要求されたテクスチャデータ、又は、要求されたテクスチャデータがメモリに常駐していないことを示す信号の何れかを、シェーダに提供する。

30

【 0 0 3 9 】

図 6 は、いくつかの実施形態による、GPUメモリ及び関連するキャッシュにテクスチャデータを投入する方法 6 0 0 のフロー図である。方法 6 0 0 は、図 3 に示すメモリ構造 3 0 0 のいくつかの実施形態において実施される。例えば、方法 6 0 0 は、図 3 に示すテクスチャブロック 3 1 0 に非常駐テクスチャデータを投入するためのシェーダ 3 5 0 からの要求に応じて実施されてもよい。

40

【 0 0 4 0 】

ブロック 6 0 5 において、GPUメモリ内のテクスチャブロックのテクスチャデータが投入される。例えば、図 1 に示す記憶媒体 1 4 5 等のディスクからテクスチャデータを取得することによって、テクスチャデータを投入することができる。また、プロシージャに従ってテクスチャデータを生成することによって、テクスチャデータを投入することができる。例えば、CPU 1 3 0 は、プログラムコードのコピー 1 2 5 内の命令を実行して、

50

GPUメモリ内のテクスチャブロックのテクスチャデータを、プロシージャに従って生成することができる。

【0041】

ブロック610において、投入されたテクスチャブロックのメタデータは、テクスチャデータがGPUメモリに常駐していることを示すように変更される。例えば、テクスチャデータの圧縮率及びテクスチャデータの常駐状態を符号化するメタデータの値を変更して、テクスチャデータの圧縮率及び「常駐」という常駐状態を示すことができる。テクスチャブロックのメタデータが以前にメタデータキャッシュにキャッシュされていた場合、メタデータキャッシュのエントリが変更され（ブロック615）、テクスチャブロックの新たな常駐状態が示される。例えば、キャッシュコヒーレンシプロトコルを使用して、符号化されたメタデータのキャッシュされた値を変更することができる。

10

【0042】

図7は、いくつかの実施形態による、テクスチャブロックに関連するメタデータ面700のブロック図である。メタデータ面700は、タイルの常駐状態と、テクスチャの対応する部分（タイル等）に使用される圧縮率と、を示すエントリ705（明確さのために符号によって1つのみが示されている）を含む。したがって、メタデータ面700は、図3に示す対応するテクスチャブロック310のメタデータ320のいくつかの実施形態を表すことができる。メモリに常駐しているタイルは、クロスハッチングされたエントリ710（明確さのために符号によって1つのみが示されている）によって示されており、以前にサンプリングされた常駐タイル715は、「S」という文字によって示されている。本明細書で説明するように、ビットの異なるサブセットによって常駐状態及びサンプリング状態を示すことができ、又は、これらを同じビットのセットに符号化することができる。

20

【0043】

タイルの常駐状態又はサンプリング状態は、メタデータ面700のサンプリング部分によって決定されてもよい。例えば、シェーダは、サンプリングフットプリント720内のエントリ705、710をサンプリングして、メタデータ面700に関連するテクスチャ内の対応するタイルの常駐状態又はサンプリング状態を決定することができる。いくつかの実施形態では、シェーダは、タイルのテクスチャデータを要求することなく、サンプリングフットプリント720内のタイルの常駐状態をチェックする。タイルの常駐状態のチェックを、タイルからテクスチャデータへのアクセスと共に実行することができる。例えば、シェーダは、テクスチャ内の非常駐タイルを示すフィードバックを使用して、別のパス間で処理されるタイルのワークリストを生成することができる。ワークリスト内の非常駐タイルには、別のパスの間に有効なデータを投入することができる。シェーダは、テクスチャデータのサンプリングをタイル上で実行し、有効なデータがタイルに投入されていることによって、メタデータ面700への追加のアクセスを必要としない。いくつかの実施形態では、シェーダは、タイルのサンプリング状態を変更するチェック動作を実行する。例えば、サンプリングフットプリント720内のタイルに対してチェック動作を実行する場合、サンプリングフットプリント720内のメタデータの一部のエントリのサンプリング状態が、チェック動作に応じて「サンプリング」に変更される。別のパスでは、シェーダは、メタデータ面700を読み出し、「非常駐」及び「サンプリング」であるエントリを識別する。次に、有効なデータが、非常駐のサンプリングされたタイルに投入される。このアプローチは、1つ以上の以前のパスの間のタイルのサンプリングアクセス数にかかわらず、シェーダがメタデータ検査パスの間にのみサンプリング状態にアクセスするので、重複排除の問題に対処する。

30

【0044】

図8は、いくつかの実施形態による、メタデータ面をサンプリングすることによって非常駐タイルのワークリストを生成する方法800のフロー図である。方法800は、図7に示すメタデータ面700のいくつかの実施形態を使用して実施され、図1に示す処理システム100及び図2に示すグラフィックスパイプライン200のいくつかの実施形態において実施されてもよい。

40

50

## 【 0 0 4 5 】

ブロック 8 0 5 において、第 1 パスでは、シェーダは、メタデータ面のサンプリングフットプリント内でメタデータをサンプリングすることによって、常駐チェックを実行する。例えば、シェーダは、テクスチャの対応する部分又はタイルのメタデータエントリにおいて符号化された常駐状態をチェックすることができる。本明細書で説明するように、常駐状態は、GPUメモリが対応する部分又はタイルのテクスチャデータを含むかどうかを示す。テクスチャデータは、常駐チェックに応じてシェーダに返されない。

## 【 0 0 4 6 】

ブロック 8 1 0 において、シェーダは、テクスチャデータを投入する必要がある非常駐タイルを含むワークリストを構築する。例えば、シェーダは、メタデータにおいて符号化された常駐状態に基づいて、非常駐タイルを識別することができる。次に、シェーダは、テクスチャデータが投入されていないタイルの識別子又はアドレスを含むワークリストを生成することができる。このワークリストを、サンプリングフットプリント内のメタデータエントリを通過した後のアクセスのために記憶することができる。

10

## 【 0 0 4 7 】

ブロック 8 1 5 において、シェーダは、ワークリスト内の情報を使用して、非常駐タイルに入力する。例えば、シェーダは、ワークリストを取得した後に、ワークリスト内の情報に基づいてテクスチャデータを取得し、又は、プロシージャに従って生成することができる。

## 【 0 0 4 8 】

ブロック 8 2 0 において、第 2 パスでは、シェーダは、テクスチャデータサンプリングを実行して、テクスチャ内のタイルからテクスチャデータを取得することができる。このパスの間にシェーダがテクスチャデータを取得できるように、テクスチャデータを GPUメモリで利用可能にする必要がある。したがって、メタデータをサンプリングすることによって実行される常駐チェックと共に、テクスチャデータサンプリングが実行される。

20

## 【 0 0 4 9 】

図 9 は、いくつかの実施形態による、テクスチャデータを、非常駐の、以前にサンプリングされたタイルに投入する方法 9 0 0 のフロー図である。方法 9 0 0 は、図 7 に示すメタデータ面 7 0 0 のいくつかの実施形態を使用して実施され、図 1 に示す処理システム 1 0 0 及び図 2 に示すグラフィックスパイプライン 2 0 0 のいくつかの実施形態において実施されてもよい。

30

## 【 0 0 5 0 】

ブロック 9 0 5 において、メタデータ面を通過する第 1 パスでは、シェーダは、常駐チェックを実行して、1 つ以上のタイルのサンプリング状態を変更する。シェーダのいくつかの実施形態は、メタデータ面のサンプリングフットプリント内でメタデータをサンプリングすることによって、常駐チェックを実行することができる。以前にサンプリングされていない非常駐タイルのサンプリング状態は、例えば、メタデータ面内のエントリのビット値又は符号化を変更することによって、「サンプリングされていない」から「サンプリング」に変更する。しかしながら、ブロック 9 0 5 において実行される常駐チェックは、如何なるフィードバックもシェーダに返さない。本明細書で説明するように、常駐状態は、GPUメモリが対応する部分又はタイルのテクスチャデータを含むかどうかを示す。

40

## 【 0 0 5 1 】

ブロック 9 1 0 において、第 2 パスでは、シェーダは、メタデータ面内のエントリを読み出し、エントリを検査して、非常駐のサンプリングされたタイルを識別する。ブロック 9 1 5 において、シェーダは、非常駐のサンプリングされたタイルに入力する。例えば、シェーダは、非常駐のサンプリングされたタイルのテクスチャデータを取得し、又は、プロシージャに従って生成することができる。ブロック 9 2 0 において、シェーダは、GPUメモリに記憶されたタイル上でテクスチャデータサンプリングを実行して、テクスチャデータを取得する。したがって、シェーダは、タイルの常駐状態のチェックと共に、テク

50

スチャデータサンプリングを実行する。

【 0 0 5 2 】

いくつかの実施形態では、上述した装置及び技術は、1つ以上の集積回路（ＩＣ）デバイス（集積回路パッケージ又はマイクロチップとも呼ばれる）を備えるシステム（例えば、図１～図６を参照して上述したグラフィックス処理システム等）において実施される。これらのＩＣデバイスの設計及び製造には、通常、電子設計自動化（ＥＤＡ）及びコンピュータ支援設計（ＣＡＤ）ソフトウェアツールが使用される。これらの設計ツールは、通常、1つ以上のソフトウェアプログラムとして表される。1つ以上のソフトウェアプログラムは、回路を製造するための製造システムを設計又は適合するための処理の少なくとも一部を実行するように1つ以上のＩＣデバイスの回路を表すコードで動作するようにコンピュータシステムを操作する、コンピュータシステムによって実行可能なコードを含む。このコードは、命令、データ、又は、命令及びデータの組み合わせを含むことができる。設計ツール又は製造ツールを表すソフトウェア命令は、通常、コンピューティングシステムがアクセス可能なコンピュータ可読記憶媒体に記憶される。同様に、ＩＣデバイスの設計又は製造の1つ以上のフェーズを表すコードは、同じコンピュータ可読記憶媒体又は異なるコンピュータ可読記憶媒体に記憶されてもよいし、同じコンピュータ可読記憶媒体又は異なるコンピュータ可読記憶媒体からアクセスされてもよい。

10

【 0 0 5 3 】

コンピュータ可読記憶媒体は、命令及び／又はデータをコンピュータシステムに提供するために、使用中にコンピュータシステムによってアクセス可能な任意の非一時的な記憶媒体又は非一時的な記憶媒体の組み合わせを含む。かかる記憶媒体には、限定されないが、光媒体（例えば、コンパクトディスク（ＣＤ）、デジタル多用途ディスク（ＤＶＤ）、ブルーレイ（登録商標）ディスク）、磁気媒体（例えば、フロッピー（登録商標）ディスク、磁気テープ、磁気ハードドライブ）、揮発性メモリ（例えば、ランダムアクセスメモリ（ＲＡＭ）、キャッシュ）、不揮発性メモリ（例えば、読み出し専用メモリ（ＲＯＭ）、フラッシュメモリ）、又は、微小電気機械システム（ＭＥＭＳ）ベースの記憶媒体が含まれ得る。コンピュータ可読記憶媒体は、コンピュータシステムに内蔵されてもよいし（例えば、システムＲＡＭ又はＲＯＭ）、コンピュータシステムに固定的に取り付けられてもよいし（例えば、磁気ハードドライブ）、コンピュータシステムに着脱可能に取り付けられてもよいし（例えば、光学ディスク又はユニバーサルシリアルバス（ＵＳＢ）ベースのフラッシュメモリ）、有線又は無線のネットワークを介してコンピュータシステムに接続されてもよい（例えば、ネットワークアクセス可能なストレージ（ＮＡＳ））。

20

30

【 0 0 5 4 】

いくつかの実施形態では、上記の技術のいくつかの態様は、ソフトウェアを実行する処理システムの1つ以上のプロセッサによって実装されてもよい。ソフトウェアは、非一時的なコンピュータ可読記憶媒体に記憶され、又は、非一時的なコンピュータ可読記憶媒体上で有形に具現化された実行可能命令の1つ以上のセットを含む。ソフトウェアは、1つ以上のプロセッサによって実行されると、上記の技術の1つ以上の態様を実行するように1つ以上のプロセッサを操作する命令及び特定のデータを含むことができる。非一時的なコンピュータ可読記憶媒体は、例えば、磁気若しくは光ディスク記憶デバイス、例えばフラッシュメモリ等のソリッドステート記憶デバイス、キャッシュ、ランダムアクセスメモリ（ＲＡＭ）、又は、他の不揮発性メモリデバイス等を含むことができる。非一時的なコンピュータ可読記憶媒体に記憶された実行可能命令は、ソースコード、アセンブリ言語コード、オブジェクトコード、又は、1つ以上のプロセッサによって解釈若しくは実行可能な他の命令フォーマットであってもよい。

40

【 0 0 5 5 】

上述したものに加えて、概要説明において説明した全てのアクティビティ又は要素が必要とされているわけではなく、特定のアクティビティ又はデバイスの一部が必要とされない場合があり、1つ以上のさらなるアクティビティが実行される場合があり、1つ以上のさらなる要素が含まれる場合があることに留意されたい。さらに、アクティビティが列挙

50



された順序は、必ずしもそれらが実行される順序ではない。また、概念は、特定の実施形態を参照して説明された。しかしながら、当業者であれば、特許請求の範囲に記載されているような本発明の範囲から逸脱することなく、様々な変更及び変形を行うことができるのを理解するであろう。したがって、明細書及び図面は、限定的な意味ではなく例示的な意味で考慮されるべきであり、これらの変更形態の全ては、本発明の範囲内に含まれることが意図される。

【 0 0 5 6 】

利益、他の利点及び問題に対する解決手段を、特定の実施形態に関して上述した。しかし、利益、利点、問題に対する解決手段、及び、何かしらの利益、利点若しくは解決手段が発生又は顕在化する可能性のある特徴は、何れか若しくは全ての請求項に重要な、必須の、又は、不可欠な特徴と解釈されない。さらに、開示された発明は、本明細書の教示の利益を有する当業者には明らかな方法であって、異なっているが同様の方法で修正され実施され得ることから、上述した特定の実施形態は例示にすぎない。添付の特許請求の範囲に記載されている以外に本明細書に示されている構成又は設計の詳細については限定がない。したがって、上述した特定の実施形態は、変更又は修正されてもよく、かかる変更形態の全ては、開示された発明の範囲内にあると考えられることが明らかである。したがって、ここで要求される保護は、添付の特許請求の範囲に記載されている。

10

20

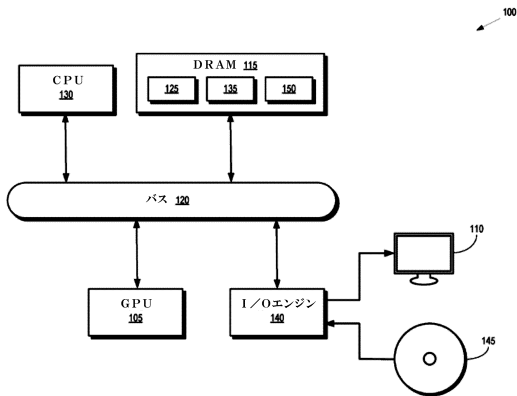
30

40

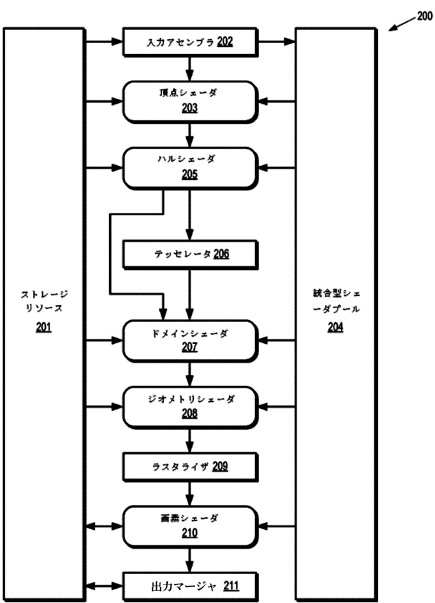
50

【図面】

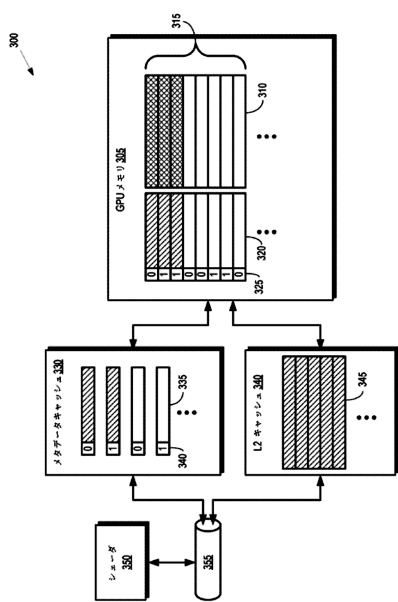
【図 1】



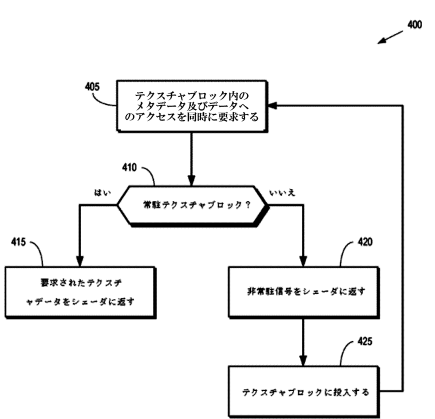
【図 2】



【図 3】



【図 4】



10

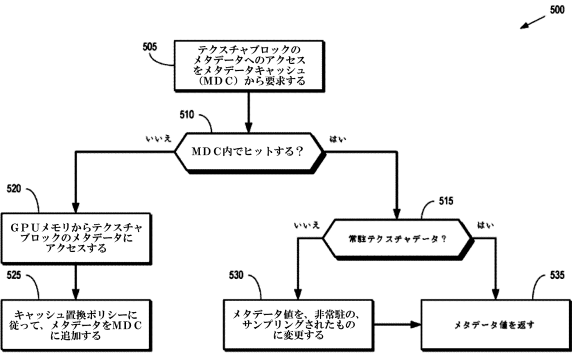
20

30

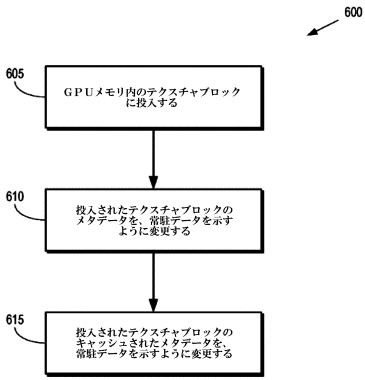
40

50

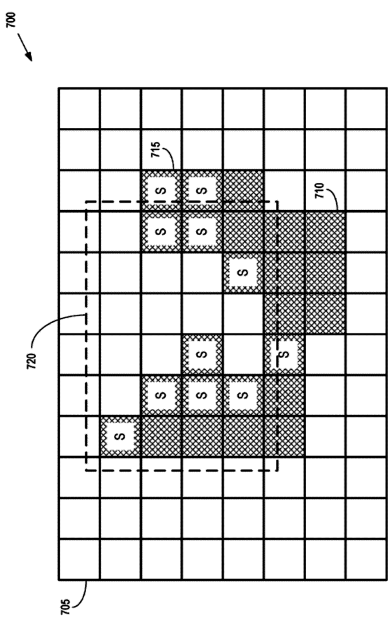
【図 5】



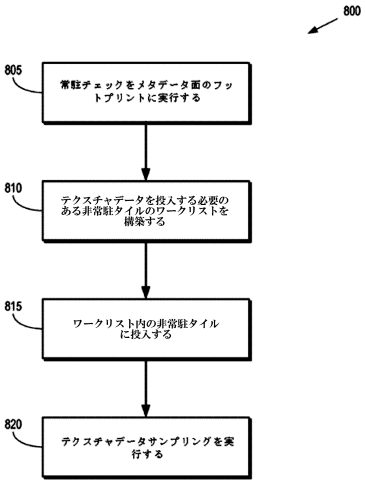
【図 6】



【図 7】



【図 8】



10

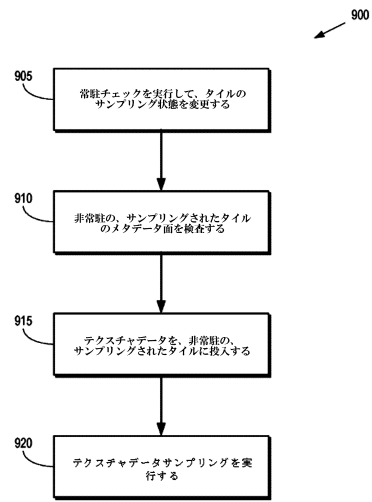
20

30

40

50

【図 9】



10

20

30

40

50

フロントページの続き

アメリカ合衆国 9 5 0 5 4 カリフォルニア州、サンタ クララ、オーガスティン ドライブ 2 4 8 5  
(72)発明者 スカイラー ジェイ . サレハ  
アメリカ合衆国 9 5 0 5 4 カリフォルニア州、サンタ クララ、オーガスティン ドライブ 2 4 8 5  
(72)発明者 ルイジン ウ  
アメリカ合衆国 9 5 0 5 4 カリフォルニア州、サンタ クララ、オーガスティン ドライブ 2 4 8 5  
(72)発明者 サーガル シャンカル バンダーレー  
アメリカ合衆国 9 5 0 5 4 カリフォルニア州、サンタ クララ、オーガスティン ドライブ 2 4 8 5  
審査官 橋爪 正樹  
(56)参考文献 米国特許出願公開第 2 0 1 5 / 0 3 7 9 6 8 4 ( U S , A 1 )  
特開平 1 1 - 2 3 2 4 7 0 ( J P , A )  
(58)調査した分野 (Int.Cl. , D B 名)  
G 0 6 T 1 5 / 0 0 - 1 9 / 2 0