

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2012-146323

(P2012-146323A)

(43) 公開日 平成24年8月2日(2012. 8. 2)

(51) Int.Cl.	F I	テーマコード (参考)
G06T 15/00 (2011.01)	G06T 15/00 I O O A	5 B O 5 7
G06T 1/20 (2006.01)	G06T 1/20 B	5 B O 8 0

審査請求 有 請求項の数 14 O L (全 18 頁)

(21) 出願番号	特願2012-58179 (P2012-58179)	(71) 出願人	501176037
(22) 出願日	平成24年3月15日 (2012. 3. 15)		イマジネーション テクノロジーズ リミテッド
(62) 分割の表示	特願2010-535457 (P2010-535457) の分割		イギリス ハートフォードシャー ダブリン ユーディー4 8エルゼット キングス ラングリー ホーム パーク エステイト (番地なし)
原出願日	平成20年12月1日 (2008. 12. 1)	(74) 代理人	100092093
(31) 優先権主張番号	0723536.9		弁理士 辻居 幸一
(32) 優先日	平成19年11月30日 (2007. 11. 30)	(74) 代理人	100082005
(33) 優先権主張国	英国 (GB)		弁理士 熊倉 禎男
		(74) 代理人	100067013
			弁理士 大塚 文昭
		(74) 代理人	100086771
			弁理士 西島 孝喜

最終頁に続く

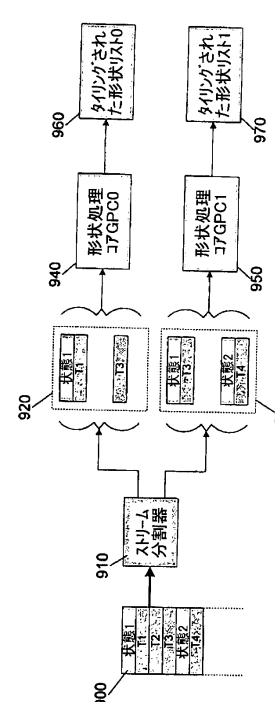
(54) 【発明の名称】 タイルベース・レンダリング・システムにおけるマルチコアの形状処理

(57) 【要約】

【課題】多数のタイルベースの並列コアにわたるタイリング処理能力の向上。

【解決手段】多数の独立型タイルベース・グラフィック・コアを組み合わせるための方法及び装置が提供される。入ってくる形状ストリームは、複数のストリームに分割され、それぞれのタイルベースのグラフィックス処理コアに送られる。それぞれが別個のタイリングされた形状リストを生成する。これらは、マスター・タイリング・ユニットに組み合わせることができ、或いは代替的に、マーカーをタイリングされた形状リストに挿入することもでき、このマーカーがラスター化段階で用いられ、異なる形状処理コアからのタイリング・リスト間で切り替えを行なう。

【選択図】図9



「ハイブライド・インタリープ・マーカー」を用いた多数のコアにわたる形状処理の分割

【特許請求の範囲】**【請求項 1】**

三次元グラフィックス・レンダリング・システムにおいて、形状処理及びタイリングを行なうための方法であって、

レンダリングされるシーンについてのグラフィックス・プリミティブ・データのストリームを提供するステップであって、各々のプリミティブは、そのプリミティブを構成する複数の三角形を定めるデータを含む、ステップと、

複数の形状処理ユニット間でプリミティブ・データを分割するステップと、

各々の形状処理ユニットにより処理される各三角形について、三角形を、その形状処理ユニットと関連した 1 組のタイリングされた形状リストにおけるそれぞれのタイルリストに挿入するステップと、

各タイルについて、マーカーを、各々のグラフィックス処理ユニットにより処理される形状の各ブロックについてのその形状処理ユニットと関連した前記タイリングされた形状リストに挿入するステップと、

各々の形状処理ユニットと関連した前記タイリングされた形状リストから、レンダリングされる前記シーンについてのタイリング・データを生成するステップと、

前記タイリングされた形状リスト内の前記マーカーを用いて、異なるグラフィックス処理ユニットからのタイリングされた形状リスト間でいつ切り換えるかを示すステップと、を含むことを特徴とする方法。

【請求項 2】

前記プリミティブ・データを分割するステップは、ラウンドロビン・ベースで分割するステップを含むことを特徴とする、請求項 1 に記載の方法。

【請求項 3】

前記プリミティブ・データを分割するステップは、実質的に類似した量のプリミティブ・データを各々の形状処理ユニットに分配するよう配置されることを特徴とする、請求項 1 又は請求項 2 に記載の方法。

【請求項 4】

前記プリミティブ・データを分割するステップは、各々のグラフィックス処理ユニットにかかる処理負荷を監視し、それらの処理負荷に応じてグラフィックス処理ユニット間でデータを分割するステップを含むことを特徴とする、請求項 1 に記載の方法。

【請求項 5】

前記プリミティブ・データを分割するステップと、それを前記グラフィックス処理ユニットに送るステップとの間にプリミティブ・データをバッファに入れるステップを含むことを特徴とする、請求項 1 から請求項 4 までのいずれかに記載の方法。

【請求項 6】

前記タイリング・データを生成するステップの前に、タイル・リファレンス・リスト及び前記タイリングされた形状リストからのデータをバッファに入れるステップを含むことを特徴とする、請求項 1 から請求項 5 までのいずれかに記載の方法。

【請求項 7】

前記タイリングされた形状リスト内の前記マーカーを用いるステップは、前記タイリングされた形状リストの各々から領域ヘッダを読み取るステップと、前記シーンのレンダリングに用いるために、タイルごとに順番に、各々のグラフィックス処理ユニット・リストにより生成された前記タイリングされた形状リストの前記開始点を指し示すポインタ・アレイに、タイリング・リスト・ポインタを書き込むステップとを含むことを特徴とする、請求項 1 から請求項 6 までのいずれかに記載の方法。

【請求項 8】

三次元グラフィックス・レンダリング・システムにおいて、形状処理及びタイリングを行なうためのシステムであって、

レンダリングされるシーンについてのグラフィックス・プリミティブ・データのストリームを提供するための手段であって、各々のプリミティブは、そのプリミティブを構成す

10

20

30

40

50

る複数の三角形を定めるデータを含む、手段と、

複数の形状処理ユニット間でプリミティブ・データを分割するための手段と、

処理される各三角形について、三角形を、その形状処理ユニットと関連した１組のタイリングされた形状リストにおけるタイリングされた形状リストに挿入するように配置された各々の形状処理ユニットと、

そのグラフィックス処理ユニットにより処理される形状の各ブロックについて、各タイルについてのマーカーを、その形状処理ユニットと関連した前記タイリングされた形状リストに挿入するための手段と、

各々の形状処理ユニットと関連した前記タイリングされた形状からレンダリングされる前記シーンについてのタイリング・データを生成し、前記タイリングされた形状リストにおける前記マーカーを用いて異なるグラフィックス処理コアからのタイリングされた形状リスト間でいつ切り替えるかを示すための手段と、

を含むことを特徴とするシステム。

【請求項 9】

前記プリミティブ・データを分割するための手段は、ラウンドロビン・ベースでこれを行なうことを特徴とする、請求項 8 に記載のシステム。

【請求項 10】

前記プリミティブ・データを分割するための手段は、実質的に類似した量のプリミティブ・データを各々の形状処理ユニットに分配するよう配置されることを特徴とする、請求項 8 又は請求項 9 に記載のシステム。

【請求項 11】

前記プリミティブ・データを分割するための手段は、各々のグラフィックス処理ユニットにかかる処理負荷を監視するための手段と、それらの処理負荷に応じてグラフィックス処理ユニット間でデータを分割するための手段とを含むことを特徴とする、請求項 8 に記載のシステム。

【請求項 12】

前記プリミティブ・データを分割するための手段と前記グラフィックス処理ユニットとの間でプリミティブ・データをバッファに入れるための手段を含むことを特徴とする、請求項 8 から請求項 11 までのいずれかに記載のシステム。

【請求項 13】

タイル・リファレンス・リスト及び前記タイリングされた形状からのデータを、前記タイリング・データを生成するための手段に提供する前にバッファに入れるための手段を含むことを特徴とする、請求項 8 から請求項 11 までのいずれかに記載のシステム。

【請求項 14】

前記タイリングされた形状リスト内の前記マーカーを用いるための手段は、前記タイリングされた形状リストの各々から領域ヘッダを読み取るための手段と、各々のグラフィックス処理ユニットにより生成された前記タイリングされた形状リストの前記開始点を指し示すポインタ・アレイに、タイリング・リスト・ポインタを書き込むための手段と、前記シーンのレンダリングに用いるために、タイルごとに順番に前記タイリングされた形状リストをフェッチするための手段とを含むことを特徴とする、請求項 1 から請求項 13 までのいずれかに記載のシステム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、三次元コンピュータ・グラフィックス・レンダリング・システムに関し、具体的には、形状処理 (geometry processing) 性能を向上させるために、多数の独立型タイルベース・グラフィックス・コアを組み合わせることに関連した方法及び装置に関する。

【背景技術】

【0002】

10

20

30

40

50

例えば基本的な手持ち式用途のものから高度な専用グラフィック・コンピュータまでといった、多くの異なる性能レベルのコンピュータ・グラフィックス処理コアを提供することが望ましいが、現代のコンピュータ・グラフィックスの複雑さにより、これをタイムリーに、或いは費用対効果が大い方法で行なうことは困難である。従って、新しいコア全体を開発しないで、性能を向上させることができるように、多数の独立型コアを組み合わせる方法を有することが望ましい。

【0003】

タイルベースのレンダリング・システムは、公知である。このシステムは、画像を複数の矩形のブロック又はタイルに細分する。図1は、タイルベースのレンダリング・システムの例を示す。プリミティブ/コマンド・フェッチ・ユニット101が、メモリからコマンド及びプリミティブ・データを取り出し、これを形状処理ユニット102に送る。この形状処理ユニット102は、周知の方法を用いて、プリミティブ及びコマンド・データを画面空間に変換する。次に、このデータは、画面空間形状からのオブジェクト・データを、1組の定められた矩形領域すなわちタイルの各々についてのオブジェクト・リストに挿入するタイリング・ユニット103に与えられる。各タイルについてのオブジェクト・リストは、そのタイル内に全体的に又は部分的に存在するプリミティブ(primitive、基本要素)を含む。幾つかのオブジェクト・リストは、その中にデータを有していないこともあるが、リストは、画面上のあらゆるタイルについて存在する。これらのオブジェクト・リストは、オブジェクト・リストをタイルごとに隠面消去ユニット(hidden surface removal unit、HSR)106に与え、最終シーンに寄与しない表面を除去する(通常、それら表面は別の表面により隠されているため)タイル・パラメータ・フェッチ・ユニット105によってフェッチされる。HSRユニットは、タイル内の各プリミティブを処理し、可視ピクセルについてのデータだけを、テスト・シェーディング・ユニット(TSU)108に送る。TSUは、HSRからデータを取得し、それを用いてテクスチャをフェッチし、周知の技術を用いて可視オブジェクト内の各ピクセルにシェーディングを適用する。次にTSUは、テクスチャリングされ、シェーディングされたデータを、アルファ・テスト/フォギング/アルファ・ブレンディング・ユニット110に提供する。これは、同じく周知の技術を用いて、様々な程度の透明性/不透明性を表面に適用することができる。アルファ・ブレンディングは、オンチップ・タイル・バッファ112を用いて行なわれ、これにより、この操作のために外部メモリにアクセスする必要性がなくなる。各タイリングが終了すると、ピクセル処理ユニット114は、結果として得られたデータを、表示の準備が完了したレンダリングされたシーン・バッファに書き込む前に、パッキング及びアンチエイリアシング・フィルタリングのような何らかの必要なバックエンド処理を行なう。

【0004】

特許文献1において、形状処理及びタイリング操作を、多数のラスタライズコアを供給する別個のプロセッサに分割することによって、タイルベースのレンダリング環境内のラスタライズ性能をスケーリングする方法が記載される。この方法は、形状処理をスケーリングする問題、特に、多数のタイルベースの並列コアにわたるタイリング処理能力の問題を考慮に入れていない。

【0005】

3Dハードウェア・デバイスは、好ましくはプリミティブの順序を、それらが供給アプリケーションによってサブミットされた順序に関して保存しなければならないことが、一般に知られている。例えば、示されるように、図2は、4つの三角形T1(200)、T2(210)、T3(220)及びT4(230)を示し、これらは、T1、T2、T3、T4の順序でアプリケーションにより存在し、タイル0(240)、タイル1(250)、タイル2(260)及びタイル3(270)に重なる。タイルリストにおける三角形のオリジナルの順序を保存するために、三角形は、各タイルリストにおいては以下のように参照される。

10

20

30

40

タイル 0	タイル 1	タイル 2	タイル 3
T 1	T 2	T 3	T 3
T 2	T 3		T 4
T 3			

形状プロセッサ及びタイリング・プロセッサにわたる負荷を均等に分配するために、ラウンドロビン・ベースで又は個々のプロセッサにかかる負荷に基づいて、入力データをプロセッサにわたって分割する必要がある。しかしながら、各プロセッサは、オブジェクトのタイルリストを局所的に生成するので、オブジェクトがタイルに挿入される順序の保存には、プロセッサが、タイル毎のオブジェクト・リストに書き込む順序を制御することが必要である。この制御は、通常、存在する G P C (グラフィックス処理コア) の各々の間の通信を必要とし、このことは、存在するコアの数をスケールリングするときに、それらの設計を変更する必要があることを意味する。

10

【先行技術文献】

【特許文献】

【0006】

【特許文献 1】英国特許第 G B 2 3 4 3 5 9 8 号

【発明の概要】

【課題を解決するための手段】

【0007】

20

本発明の好ましい実施形態は、タイルベースのレンダリング・システムが、形状処理及びタイリング操作を直線状にスケールリングするのを可能にする方法及び装置を提供する。このことは、コア内でかなりの量の入ってくる形状を処理し、局所的にタイリングすること、及び、各コアからの結果として得られる領域リストをオリジナルの入力形状の順序に対応する順序で互いにリンクさせることを可能にする階層リスト構造を使用することによって達成される。これに加えて、用いられる機構により、それらのコアの各々に対する修正をほとんど又は全く必要とすることなく、多数のコアを並行して用いることが可能になる。

【0008】

好ましくは、本発明の実施形態は、多数の独立したタイルベースのグラフィック・コアを組み合わせるための方法及び装置を提供し、そこでは、入ってくる形状ストリームが、タイルベース・グラフィックス処理コアごとに 1 つの、複数の形状ストリームに分割され、次に、各コア及びこれが処理する各三角形についての別個のタイリングされた形状リストが、マスター・タイリング・ユニットを用いて組み合わせられ、このマスター・タイリング・ユニットは、形状処理コアからデータを取得して各タイルについてマスター・タイルリストを生成し、入力された形状の順序を保存するか、或いは、ラスター化の際に、用いられるタイリングされた形状リスト内でマーカー使用し、ラスター化コアがリスト間で切り替わるようにする。

30

【0009】

ここで、添付図面を参照して、本発明の好ましい実施形態を一例として詳細に説明する。

40

【図面の簡単な説明】

【0010】

【図 1】 上述したような従来技術のタイルベースのレンダリング・システムを示す。

【図 2】 上述したような例示的な組の 4 つ三角形が 4 つのタイルに重なることを示す。

【図 3】 多数のコアにわたって制御ストリームを分割することを示す。

【図 4】 本発明の 1 つの実施形態においてタイル・リファレンス・リストのために提案されたデータ構造を示す。

【図 5】 本発明の 1 つの実施形態における、提案された階層タイルリスト・データ構造を示す。

50

【図 6】本発明の 1 つの実施形態において、制御ストリームを多数のコアにわたってより粗い粒度で分割することを示す。

【図 7】本発明を具体化するシステムの例示的な実装を示す。

【図 8】図 7 の例示的なシステムへの修正を示す。

【図 9】パイプ・インターリーブ・マーカーを用いて、制御ストリームを多重コアにわたって分割することを示す。

【図 10】パイプ・インターリーブ・マーカーを用いる、本発明の 1 つの実施形態における提案されたタイルリスト・データ構造を示す。

【図 11】パイプ・インターリーブ・マーカーを用いて、互いにリンクされた多数のタイルリストを処理するためにラスタ化コアにより用いられるアルゴリズムを示す。

【図 12】パイプ・インターリーブ・マーカーを用いる例示的な形状処理システムを示す。

【図 13】PIM ベースの形状システムにより生成されるタイルリストを処理することができる、タイルベースのラスタ化システムのフロントエンドの例を示す。

【発明を実施するための形態】

【0011】

図 3 は、形状リストを組み合わせるためにマスター・タイリング・ユニットを用いる提案されたシステムの単純化されたバージョンを示す。入ってくる形状ストリーム 300 を多数のコアにわたって処理するために、(この例においては) 2 つの形状処理コア (GPC) 340 及び 350 にわたって処理するためにデータ・ストリームを分割するストリーム分割器 310 によって、初めに、入ってくる形状ストリーム 300 を、コア毎に 1 つの別個のストリームに分割する必要がある。この例においては、入ってくるストリームは、単純なラウンドロビン・ベースで、2 つのコアにわたって分配されるが、各コアにかかる処理負荷に基づいて、ストリームを処理コアにわたって分割することも可能である。ストリーム分割器は、制御状態及びプリミティブを各々の形状処理コアに分配することに留意することが重要である。制御状態は、形状データをどのように処理するかについて、ハードウェアに指示する情報を含む。例えば、マトリクスを用いて、形状を周知の方法で変換することができ、或いは、テクスチャリングをどのように形状に適用するかの詳細を含ませることができる。例えば、GPC 0 320 についての制御ストリームは、三角形 T1 - T3 と関連した制御状態 1 を含み、GPC 1 330 についての制御状態は、T1 - T3 についての状態 1 及び T4 についての状態 4 を含む。

【0012】

GPC 340 及び 350 の各々は、それが処理する各三角形について、別個のタイリングされた形状リスト 360 及び 370 を生成する。各三角形ごとに更新された各タイルについて、GPC は、タイル・リファレンス・リスト (TRL) 365 及び 375 のそれぞれのものに、リファレンスを挿入する。TRL 及び GPC 毎のタイリングされた形状リストは、図 4 に示すデータ構造を形成し、マスター・タイリング・ユニット 380 を生成するために、マスター・タイリング・ユニット 380 により使用される。

【0013】

GPC 0 の TRL 400 は、そのコアにより処理される三角形 T1 及び T3 についてのリファレンスを含み、例えば、T1 はタイル 0、0 のみの中に存在するので、リファレンス及び対応するタイリングされた形状リスト 410 へのポインタは、TRL 内に含まれ、これに、4 つのタイル全てにおける T3 についてのリファレンスが続く。同様に、GPC 1 の TRL 420 は、GPC 1 についての対応するタイリングされた形状リスト 430 において、T2 及び T4 に重なる各タイルについてのリファレンスを含む。タイリングされた三角形のリストは、430 において、各三角形が示された後に「End」マーカーを含むことに留意すべきである。

【0014】

図 3 のマスター・タイリング・ユニット (MTU) 380 は、(この例においては) プリミティブが GPC にわたって分配される同じラウンドロビン順序で TRL を読み取り、

10

20

30

40

50

次に移る前に、各 T R L から 1 つの三角形についてのタイル・リファレンスを取得する。M T U 3 8 0 は、タイル・リファレンスを取得し、マスター・タイルリスト 3 9 0 を生成し、このリストは、図 5 に示すデータ構造を有する。各々がトップレベルのタイルリスト 5 1 0 内の対応するタイルリストを指し示す、通常のタイルベース・レンダリング・システムの領域ヘッダ 5 0 0 と同様に、これらのリストは、三角形のオリジナルの提示順序を保存し、各々が「T e r m i n a t e (終了)」マスターを有することに留意すべきである。トップレベルのタイルリストは、G P C により生成され上述されたような、各タイル 5 2 0 及び 5 3 0 内の参照された三角形リストへのリンクを含む。

【 0 0 1 5 】

上述のように、各 G P C のタイルリスト内の各三角形の後には、「E n d」マーカーが続く。これらのマーカーは、ラスター化ハードウェアにより、G P C タイルリストから上位のタイルリストに戻るよう、これに指示するために用いられる。マーカーは、単一の三角形の代わりに、各 G P C において三角形のグループを処理できるように使用される。このことは、マーカーが、上位のタイルリストと関連したメモリの量を最小にし、かつ、幾つかの三角形における頂点の処理が他のものより時間がかかる場合に、より大きい G P C の分離を可能にするために重要である。

【 0 0 1 6 】

図 6 は、1 0 0 0 個の三角形のブロックが各 G P C に押し下げられる、多数の G P C にわたって、アプリケーションから入ってくるプリミティブ・ストリームを分割することを示す。入ってくるデータ・ストリーム 6 0 0 は、各々がそれぞれ 4 個、5 0 個、2 0 0 0 個及び 1 5 0 0 個の三角形を含む 4 つのプリミティブ、すなわち p r i m 1、2、3 及び 4 を含む。示されるように、2 つの G P C (6 5 0 及び 6 6 0) にわたる処理のために、ストリーム分割器 6 1 0 は、ストリームを 4 つのブロックに分割し、ブロック 6 2 0 及び 6 3 0 は G P C 0 に渡され、ブロック 6 4 0 及び 6 5 0 は G P C 1 に渡される。P r i m 1 及び P r i m 2 は両方とも、P r i m 3 及び P r i m 4 の第 1 の部分と共に G P C 0 に送られる。P r i m 3 及び P r i m 4 の残りの部分は、G P C 1 に送られる。分割の目的は、2 つの G P C 間の負荷を平衡させようとするものである。ブロック 6 2 0 及び 6 3 0 において、P r i m 3 からのデータも、2 つのブロック間で分割され、その両方とも G P C 0 によって処理される。これは、類似したブロック・サイズを生成するためである。T R L 及びトップレベルのデータ構造は、単一の三角形を指し示すのではなく、タイル毎のリファレンスが、各タイル内の各ブロックからの三角形のグループを指し示すことを除いて、変更されない。

【 0 0 1 7 】

図 7 は、2 つの形状処理及びタイリング・コアを用いるシステムの例示的な実装を示す。プリミティブ及びコマンド・フェッチ・ユニット 7 0 0 が、入ってくる制御ストリームを読み取り、それを、上述のように 2 つ (又はそれ以上) のコアにわたる処理のためにストリームを分割するストリーム分割器ユニット 7 0 5 に渡す。分割器は、フェッチされるプリミティブへのポインタを、「ローカル」プリミティブ・フェッチ・ユニット 7 1 5 及び 7 1 6 への入力における別個のコア、具体的には F I F O 7 1 2 及び 7 1 4 に渡す。F I F O は、各バッチを処理するのに各コアが要する時間のために、ストリーム分割処理の分離を助けなければならない。ローカル・プリミティブ・フェッチ・ユニットは、F I F O 7 1 2 及び 7 1 4 からポインタを読み取り、次に、メモリから実際の形状データを読み取り、それを形状処理ユニット 7 2 0 及び 7 2 1 に渡し、形状処理ユニット 7 2 0 及び 7 2 1 は、形状を処理すると、それをローカル・タイリング・ユニット 7 2 5 及び 7 2 6 に渡す。タイリング・ユニットは、処理された形状をタイリングし、ローカル・タイルリスト 7 3 0 及び 7 3 1 を生成し、これらリストについての T R L をタイル・リファレンス F I F O 7 4 0 及び 7 4 1 に渡し、タイル・リファレンス F I F O 7 4 0 及び 7 4 1 は、マスター・タイリング・ユニット (M T U) により消費されるのを待ちながら、既述した T R L をバッファに入れる。これらの F I F O を、外部メモリ内又はチップ上に含ませ、G P C とマスター・タイリング・ユニットとの間のバッファリング量の柔軟性を大きくでき

10

20

30

40

50

ることに留意すべきである。F I F O / バッファの使用により、G P C を M T U の操作から分離し、M T U がマスター・タイルリストの生成に相当量の時間を費やしている場合のストールを最小にすることができる。M T U は、F I F O からの T R L データを使用して、マスター領域リスト 7 6 0 を生成し、これとローカル・タイルリストと協働して、上述のようなデータ構造を形成する。

【 0 0 1 8 】

単一のラウンドロビン方式の使用により、1つのG P C が他のG P C よりも著しく時間がかかるために、分割ストリームF I F O 7 1 2 及び7 1 4 のいずれかが一杯になる場合に、ストリーム分割器がストールすることを意味することが留意されるであろう。このように、処理時間の不均衡が大きい場合には、G P C パイプラインのいずれもアイドル状態になるのを回避するため、これらのF I F O を法外に大きくする必要がある場合がある。図8は、そのコアがどれだけビジーであるかに基づいて、分割器がプリミティブを各コアに送る、すなわちその処理負荷が監視される、代替的な実施形態を示す。システムの操作は、例えば入力バッファリングの占有量 (f u l l n e s s) など、それらがどれだけビジーであるかを示す、形状処理ユニット8 2 0 及び8 2 1 からの情報が、ストリーム分割器8 0 5 に与えられる点を除いて、上述されたものと同一である。ストリーム分割器は、この情報を用いて、プリミティブのグループを、最も負荷の高くないG P C に指向させる。プリミティブがコアにサブミットされる順序は今や非決定的 (n o n d e t e r m i n i s t i c) であるため、ストリーム分割器は、M T U が T R L F I F O から T R L を正しい順序で引くことができるように、M T U のためにコア・リファレンス・シーケンスを生成する必要がある。リファレンス・シーケンスは、ストリーム分割器によりサービス・オーダーF I F O 8 7 0 に書き込まれ、サービス・オーダーF I F O 8 7 0 は、M T U は、どのT R O F I F O を次に読み取るかを決定するために、順番に読み込む。

【 0 0 1 9 】

図9は、マスター・タイリング・ユニットの代わりに、「パイプ・インターリーブ・マーカ」を用いて、多数のコアにわたって形状を処理するのを可能にするシステムを示す。マスター・タイリング・ユニット・ベースのシステムと同様に、上述のように、入ってくる形状ストリーム・データは、ストリーム分割器9 1 0 によって分割され、G P C 9 4 0、9 5 0 に分配される。各G P C は、それぞれのタイリングされた形状リスト9 6 0 及び9 7 0 を生成する。図10は、生成された、タイリングされた形状リストの構造を示す。各G P C は、通常のタイルベースのレンダリング・システムのように、各コアに通される形状についてのタイリングされた形状リストを指し示すそれぞれの領域ヘッダ1 0 0 0 及び1 0 2 0 を生成する。各形状ブロックの終わりに、G P C は、ラスタ化プロセスの際に単一のコアにより正しい順序でリストをトラバースするのを可能にするために用いられる「パイプ・インターリーブ・マーカ」(P I M) 1 0 4 0 を挿入する。

【 0 0 2 0 】

図11のフローチャートは、リストをトラバースするために、ラスタ化によりP I M マーカがどのように使用されるかを示す。1 1 0 0 において、各タイルの処理の開始時に、各コアが生成する領域ヘッダの内容が、コアリスト・ポインタ・アレイにロードされる。これにより、アレイ内の各エントリが、処理される領域の各コアのために生成された領域リストへのポインタを含むようになる。1 1 0 5 において、プリミティブ・データの第1のブロックが、アレイへのインデックスを0に設定することにより、第1のG P C すなわちG P C 0 によって処理されたと仮定して、領域リストの処理が開始される。次に、1 1 1 0 において、これがゼロであるかどうかを確認するために、ポイント値が試験され、もしゼロであれば、それは、リストが空であるか、又は、既に処理が完了したことを意味し、1 1 1 5 において、アレイ・インデックスがインクリメントされ、1 1 1 0 において行なわれた試験が繰り返される。このプロセスは、形状を含むリストが見つかるまで繰り返され、その時点で、1 1 2 0 において、ポイント・データが、アレイ・インデックスにより索引付けされたポイントを用いてタイリングされた形状リストからフェッチされる。1 1 2 5 において、これがP I M であるかどうかを判断するために、フェッチされたデ

ータが試験され、P I Mである場合には、次に、現在のリスト・ポインタが、タイリングされた形状リストの次のデータを指し示すように更新され、1 1 3 0においてコアリスト・ポインタ・アレイに書き戻される。次に、1 1 3 5において、アレイ・インデックスは、P I M内で指定された値に設定され、処理は1 1 1 0に戻る。1 1 2 5における試験によりP I Mが検知されなかった場合、1 1 4 0において、これがE n d マーカーであるかどうかを確認するために、フェッチされたデータが試験され、E n d が検知された場合、次に現在のタイリングの処理が完了し、ハードウェアは、次のタイリングの処理に移る。データがE n d マーカーではない場合、これは形状又は状態リファレンスであり、必要に応じて1 1 4 5で処理される。次に、1 1 5 0において、リスト・ポインタが更新され、処理は1 1 2 0に戻り、タイリングされた形状リスト内の次のエントリをフェッチする。

10

【0 0 2 1】

図1 2は、2つの形状処理及びタイリング・コアを用いるP I Mベースのシステムの例示的な実装を示す。プリミティブ及びコマンド・フェッチ・ユニット1 2 0 0が、入ってくる制御ストリームを読み取り、これを、上述のような2つ（又はそれ以上）のコアにわたる処理のためにストリームを分割するストリーム分割器ユニット1 2 0 5に渡す。分割器は、フェッチされるプリミティブへのポインタを、「ローカルの」プリミティブ・フェッチ・ユニット1 2 1 5及び1 2 1 6への入力における別個のコア、具体的にはF I F O 1 2 1 2及び1 2 1 4に渡す。F I F Oは、各バッチを処理するのに各コアが要する時間のために、ストリーム分割処理の分離を助けなければならない。ローカル・プリミティブ・フェッチ・ユニットは、メモリから実際の形状データを読み取り、それを形状処理ユニット1 2 2 0及び1 2 2 1に渡し、形状処理ユニット1 2 2 0及び1 2 2 1は、形状を処理すると、それをローカル・タイリング・ユニット1 2 2 5及び1 2 2 6に渡す。タイリング・ユニットは、処理された形状をタイリングし、コア毎のタイルリスト1 2 3 0及び1 2 3 1を生成する。

20

【0 0 2 2】

図1 3は、P I Mを用いて、多数のG P Cにより生成されたタイルリストをトラバースすることができるラスタ化コアのフロントエンドを示す。領域ヘッダ・フェッチ・ユニット1 3 1 0が、各G P Cにより生成された画面空間タイリングされた形状リスト1 3 0 0から領域ヘッダを読み取り、図1 1について述べられたように結果として得られたリスト・ポインタをコアリスト・ポインタ・アレイ1 3 2 0に書き込む。タイリングされた形状リスト・フェッチ・ユニット1 3 3 0は、次に、図1 1について説明されたようなタイル毎の制御リストをフェッチして処理し、処理されることになる結果として得られる形状を、隠面消去ユニット1 3 4 0に渡し、この時点から、全ての処理は、通常のタイルベースのレンダリング・システムについて説明されたものと同じである。

30

【0 0 2 3】

上記の開示事項に加えて、原特許出願2 0 1 0 - 5 3 5 4 5 7号の出願時の特許請求の範囲の記載を以下に開示する。

〔請求項1〕

三次元グラフィックス・レンダリング・システムにおいて形状処理及びタイリングを行なうための方法であって、

40

レンダリングされるシーンについてのグラフィックス・プリミティブ・データのストリームを提供するステップであって、各々のプリミティブは、そのプリミティブを構成する複数の三角形を定めるデータを含む、ステップと、

複数の形状処理ユニット間でプリミティブ・データを分割するステップと、

各々の形状処理ユニットにより処理される各三角形について、三角形を、その形状処理ユニットに関連した1組のタイリングされた形状リストにおけるそれぞれのタイルリストに挿入するステップと、

三角形が挿入される各タイルについて、そのタイルへのリファレンスを、その形状処理ユニットに関連したタイル・リファレンス・リストに挿入するステップと、

データが前記形状処理ユニットに分配された順序で、各々の形状処理ユニットと関連し

50

た前記タイル・リファレンス・リストからデータを読み取るステップと、

前記タイル・リファレンス・リスト及び各々の形状処理ユニットと関連した前記タイリングされた形状から、レンダリングされる前記シーンについてのタイリング・データを生成するステップと、

を含むことを特徴とする方法。

〔請求項 2〕

前記プリミティブ・データを分割するステップは、ラウンドロビン・ベースで分割するステップを含むことを特徴とする、請求項 1 に記載の方法。

〔請求項 3〕

前記プリミティブ・データを分割するステップは、実質的に類似した量のプリミティブ・データを各々の形状処理ユニットに分配するよう配置されることを特徴とする、請求項 1 又は請求項 2 に記載の方法。

10

〔請求項 4〕

前記プリミティブ・データを分割するステップは、各々のグラフィック処理ユニットにかかる処理負荷を監視し、それらの処理負荷に応じてグラフィック処理ユニット間でデータを分割するステップを含むことを特徴とする、請求項 1 に記載の方法。

〔請求項 5〕

前記プリミティブ・データを分割するステップと、それを前記グラフィックス処理ユニットに送るステップとの間にプリミティブ・データをバッファに入れるステップを含むことを特徴とする、前記請求項のいずれかに記載の方法。

20

〔請求項 6〕

前記タイリング・データを生成するステップの前に、タイル・リファレンス・リスト及び前記タイリングされた形状リストからのデータをバッファに入れるステップを含むことを特徴とする、前記請求項のいずれかに記載の方法。

〔請求項 7〕

前記タイリング・データを生成するステップは、前記シーンのレンダリングに用いるためのマスター・タイルリストを生成するステップを含むことを特徴とする、前記請求項のいずれかに記載の方法。

〔請求項 8〕

三次元グラフィックス・レンダリング・システムにおいて形状処理及びタイリングを行なうためのシステムであって、

30

レンダリングされるシーンについてのグラフィックス・プリミティブ・データのストリームを提供するための手段であって、各々のプリミティブは、そのプリミティブを構成する複数の三角形を定めるデータを含む、手段と、

複数の形状処理ユニット間でプリミティブ・データを分割するための手段と、

処理される各三角形について、三角形を、その形状処理ユニットと関連した 1 組のタイリングされた形状リストにおけるタイリングされた形状リストに挿入するように構成された形状処理ユニットと、

三角形が挿入される各タイルについて、そのタイルへのリファレンスをその形状処理ユニットと関連したタイル・リファレンス・リストに挿入するための手段と、

40

データが前記形状処理ユニットに分配された順序で、各々の形状処理ユニットと関連した前記タイル・リファレンス・リストからデータを読み取るための手段と、

前記タイル・リファレンス・リスト及び各々の形状処理ユニットと関連した前記タイリングされた形状から、レンダリングされる前記シーンについてのタイリング・データ・リストを生成するための手段と、

を含むことを特徴とするシステム。

〔請求項 9〕

前記プリミティブ・データを分割するための手段は、ラウンドロビン・ベースでこれを行なうことを特徴とする、請求項 8 に記載のシステム。

〔請求項 10〕

50

前記プリミティブ・データを分割するための手段は、実質的に類似した量のプリミティブ・データを各々の形状処理ユニットに分配するよう配置されることを特徴とする、請求項 8 又は請求項 9 に記載のシステム。

〔請求項 11〕

前記プリミティブ・データを分割するための手段は、各々のグラフィックス処理ユニットにかかる処理負荷を監視するための手段と、それらの処理負荷に応じてグラフィックス処理ユニット間でデータを分割するための手段とを含むことを特徴とする、請求項 8 に記載のシステム。

〔請求項 12〕

前記プリミティブ・データを分割するための手段と前記グラフィックス処理ユニットとの間でプリミティブ・データをバッファに入れるための手段を含むことを特徴とする、請求項 8 から請求項 11 までのいずれかに記載のシステム。

〔請求項 13〕

マスター・タイルリストを生成するための前記手段に提供する前に、タイル・リファレンス・リスト及び前記タイリングされた形状からのデータをバッファに入れるための手段を含むことを特徴とする、請求項 8 から請求項 12 までのいずれかに記載のシステム。

〔請求項 14〕

前記タイリング・データを生成するための手段は、前記タイリング・データが挿入されるマスター・タイルリストを含むことを特徴とする、請求項 8 から請求項 13 までのいずれかに記載のシステム。

〔請求項 15〕

三次元グラフィックス・レンダリング・システムにおいて、形状処理及びタイリングを行なうための方法であって、

レンダリングされるシーンについてのグラフィックス・プリミティブ・データのストリームを提供するステップであって、各々のプリミティブは、そのプリミティブを構成する複数の三角形を定めるデータを含む、ステップと、

複数の形状処理ユニット間でプリミティブ・データを分割するステップと、

各々の形状処理ユニットにより処理される各三角形について、三角形を、その形状処理ユニットに関連した 1 組のタイリングされた形状リストにおけるそれぞれのタイルリストに挿入するステップと、

各タイルについて、マーカーを、各々のグラフィックス処理ユニットにより処理される形状の各ブロックについてのその形状処理ユニットに関連した前記タイリングされた形状リストに挿入するステップと、

各々の形状処理ユニットに関連した前記タイリングされた形状リストから、レンダリングされる前記シーンについてのタイリング・データを生成するステップと、

前記タイリングされた形状リスト内の前記マーカーを用いて、異なるグラフィックス処理ユニットからのタイリングされた形状リスト間でいつ切り換えるかを示すステップと、を含むことを特徴とする方法。

〔請求項 16〕

前記プリミティブ・データを分割するステップは、ラウンドロビン・ベースで分割するステップを含むことを特徴とする、請求項 15 に記載の方法。

〔請求項 17〕

前記プリミティブ・データを分割するステップは、実質的に類似した量のプリミティブ・データを各々の形状処理ユニットに分配するよう配置されることを特徴とする、請求項 15 又は請求項 16 に記載の方法。

〔請求項 18〕

前記プリミティブ・データを分割するステップは、各々のグラフィックス処理ユニットにかかる処理負荷を監視し、それらの処理負荷に応じてグラフィックス処理ユニット間でデータを分割するステップを含むことを特徴とする、請求項 15 に記載の方法。

〔請求項 19〕

前記プリミティブ・データを分割するステップと、それを前記グラフィックス処理ユニットに送るステップとの間にプリミティブ・データをバッファに入れるステップを含むことを特徴とする、請求項 15 から請求項 18 までのいずれかに記載の方法。

〔請求項 20〕

前記タイリング・データを生成するステップの前に、タイル・リファレンス・リスト及び前記タイリングされた形状リストからのデータをバッファに入れるステップを含むことを特徴とする、請求項 15 から請求項 19 までのいずれかに記載の方法。

〔請求項 21〕

前記タイリングされた形状リスト内の前記マーカを用いるステップは、前記タイリングされた形状リストの各々から領域ヘッダを読み取るステップと、前記シーンのレンダリングに用いるために、タイルごとに順番に、各々のグラフィックス処理ユニット・リストにより生成された前記タイリングされた形状リストの前記開始点を指し示すポインタ・アレイに、タイリング・リスト・ポインタを書き込むステップとを含むことを特徴とする、請求項 15 から請求項 20 までのいずれかに記載の方法。

〔請求項 22〕

三次元グラフィックス・レンダリング・システムにおいて、形状処理及びタイリングを行なうためのシステムであって、

レンダリングされるシーンについてのグラフィックス・プリミティブ・データのストリームを提供するための手段であって、各々のプリミティブは、そのプリミティブを構成する複数の三角形を定めるデータを含む、手段と、

複数の形状処理ユニット間でプリミティブ・データを分割するための手段と、

処理される各三角形について、三角形を、その形状処理ユニットと関連した 1 組のタイリングされた形状リストにおけるタイリングされた形状リストに挿入するように配置された各々の形状処理ユニットと、

そのグラフィックス処理ユニットにより処理される形状の各ブロックについて、各タイルについてのマーカを、その形状処理ユニットと関連した前記タイリングされた形状リストに挿入するための手段と、

各々の形状処理ユニットと関連した前記タイリングされた形状からレンダリングされる前記シーンについてのタイリング・データを生成し、前記タイリングされた形状リストにおける前記マーカを用いて異なるグラフィックス処理コアからのタイリングされた形状リスト間でいつ切り替えるかを示すための手段と、

を含むことを特徴とするシステム。

〔請求項 23〕

前記プリミティブ・データを分割するための手段は、ラウンドロビン・ベースでこれを行なうことを特徴とする、請求項 22 に記載のシステム。

〔請求項 24〕

前記プリミティブ・データを分割するための手段は、実質的に類似した量のプリミティブ・データを各々の形状処理ユニットに分配するよう配置されることを特徴とする、請求項 22 又は請求項 23 に記載のシステム。

〔請求項 25〕

前記プリミティブ・データを分割するための手段は、各々のグラフィックス処理ユニットにかかる処理負荷を監視するための手段と、それらの処理負荷に応じてグラフィックス処理ユニット間でデータを分割するための手段とを含むことを特徴とする、請求項 22 に記載のシステム。

〔請求項 26〕

前記プリミティブ・データを分割するための手段と前記グラフィックス処理ユニットとの間でプリミティブ・データをバッファに入れるための手段を含むことを特徴とする、請求項 22 から請求項 25 までのいずれかに記載のシステム。

〔請求項 27〕

タイル・リファレンス・リスト及び前記タイリングされた形状からのデータを、前記タ

10

20

30

40

50

イリング・データを生成するための手段に提供する前にバッファに入れるための手段を含むことを特徴とする、請求項 22 から請求項 25 までのいずれかに記載のシステム。

〔請求項 28〕

前記タイリングされた形状リスト内の前記マーカを用いるための手段は、前記タイリングされた形状リストの各々から領域ヘッダを読み取るための手段と、各々のグラフィックス処理ユニットにより生成された前記タイリングされた形状リストの前記開始点を指し示すポインタ・アレイに、タイリング・リスト・ポインタを書き込むための手段と、前記シーンのレンダリングに用いるために、タイルごとに順番に前記タイリングされた形状リストをフェッチするための手段とを含むことを特徴とする、請求項 11 から請求項 27 までのいずれかに記載のシステム。

10

【符号の説明】

【0024】

300：形状ストリーム

310、610、805、910：ストリーム分割器

320、330、650、660、940、950：GPC

340、350：形状処理コア（GPC）

360、370、410、430、960、970、1010、1030：タイリングされた形状リスト

365、375、400、420：タイル・リファレンス・リスト（TRL）

380：マスター・タイリング・ユニット（MTU）

390：マスター・タイルリスト

500、1000、1020：領域ヘッダ

520、530：タイル

510：トップレベルのタイルリスト

620、630、640、650：ブロック

700、1200：プリミティブ及びコマンド・フェッチ・ユニット

705、1205：ストリーム分割器ユニット

712、714、1212、1214：FIFO

715、716、1215、1216：「ローカル」プリミティブ・フェッチ・ユニット

720、721、820、821、1220、1221：形状処理ユニット

725、726、1225、1226：ローカル・タイリング・ユニット

730、731：ローカル・タイルリスト

740、741：タイル・リファレンスFIFO

760：マスター領域リスト

870：サービス・オーダーFIFO

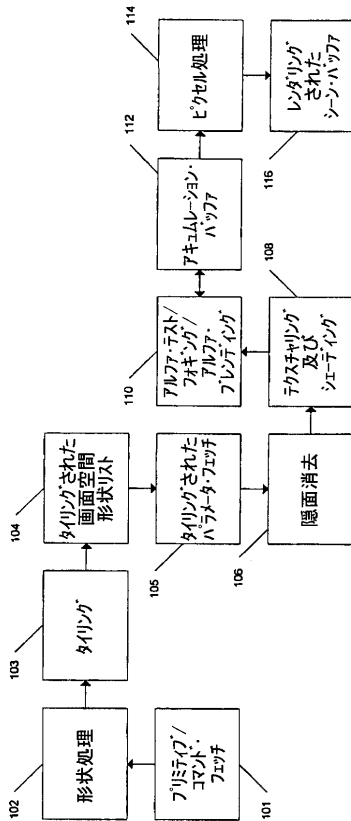
1040：「パイプ・インターリーブ・マーカ」（PIM）

1230、1231：コア毎のタイルリスト

20

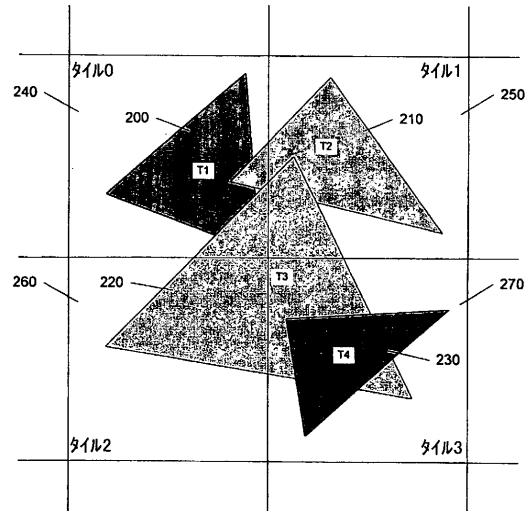
30

【図 1】



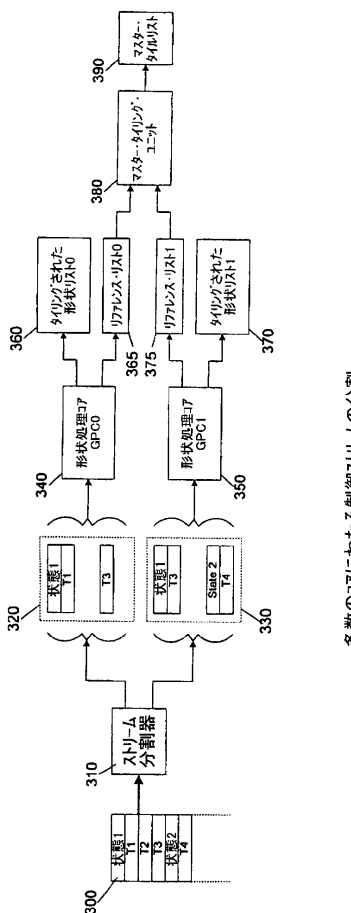
例示的なタイルベースのレンダリングシステム

【図 2】



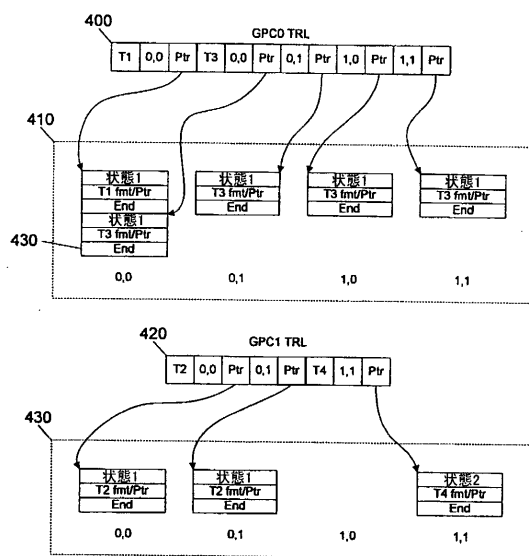
タイルの重複及びタイル内の三角形の順序の例

【図 3】



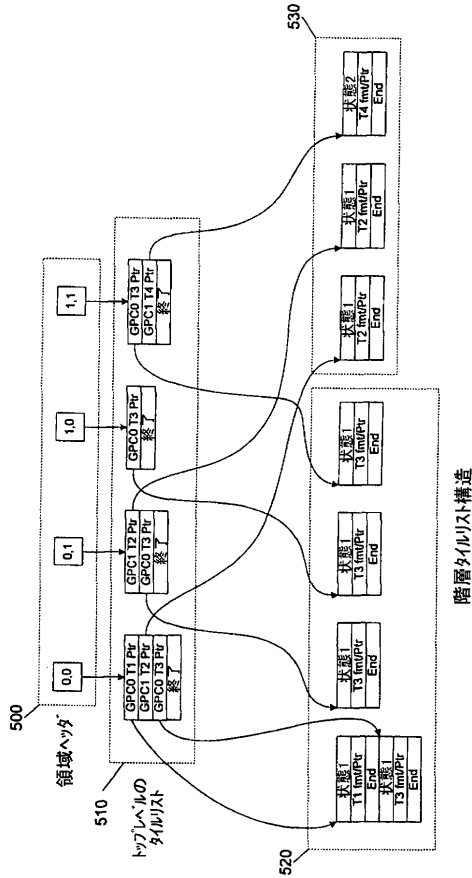
多数のコアにわたる制御ストリームの分割

【図 4】

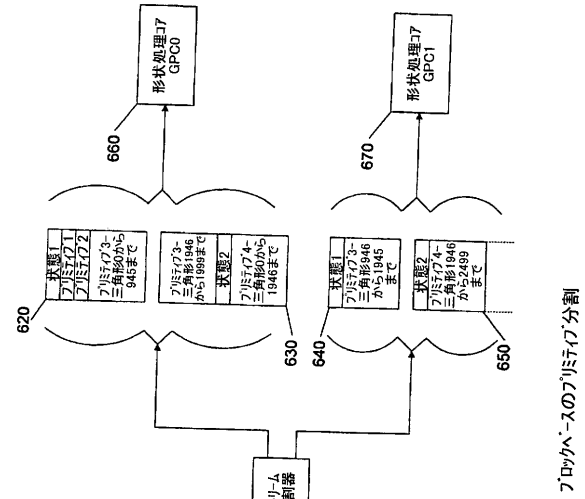


タイル・リファレンス・リストのデータ構造

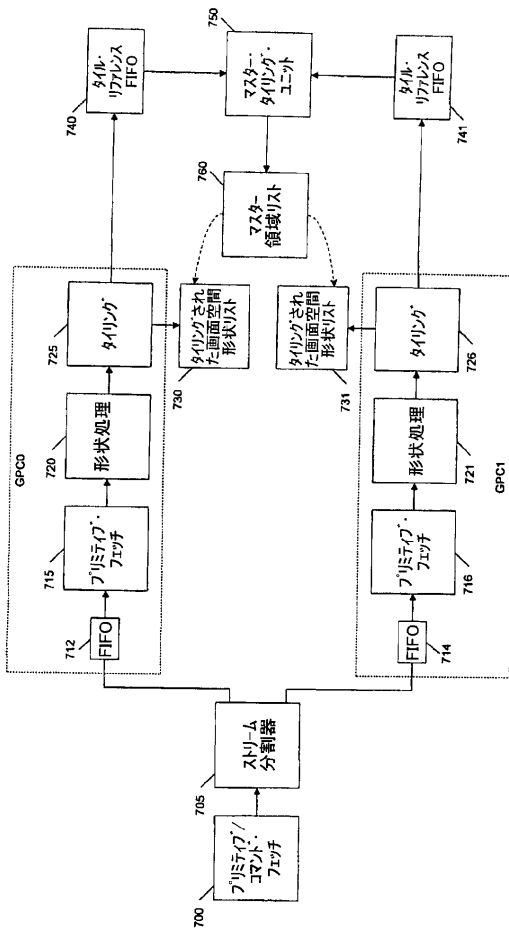
【図 5】



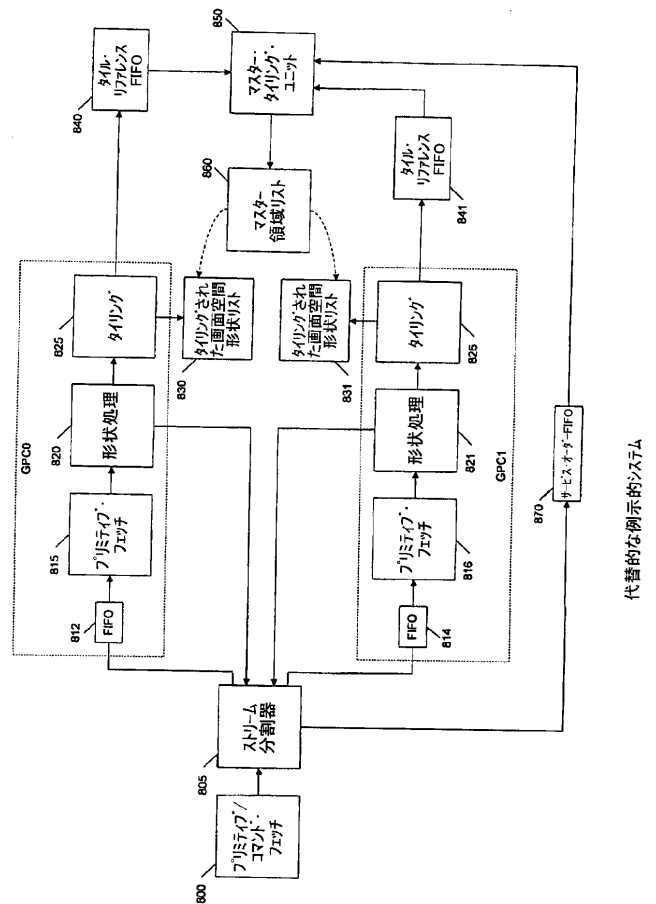
【図 6】



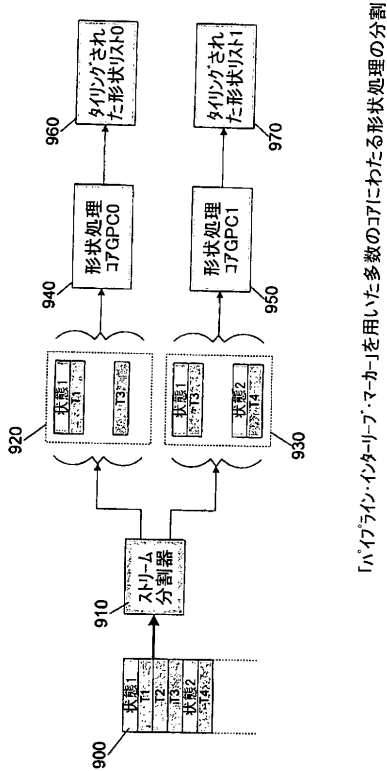
【図 7】



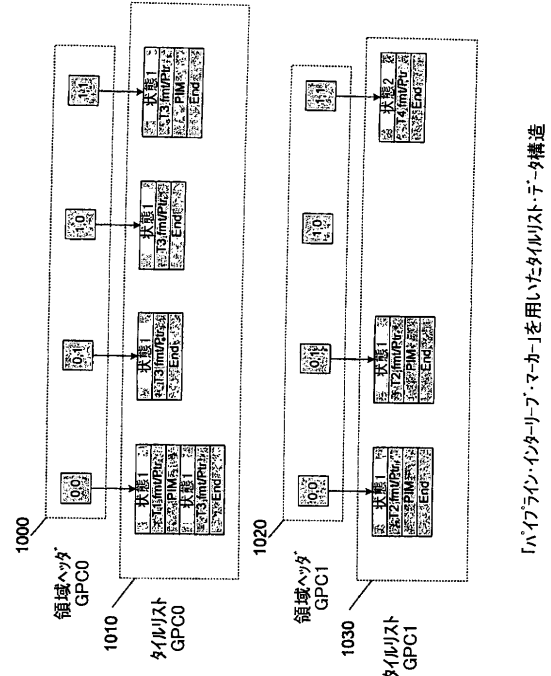
【図 8】



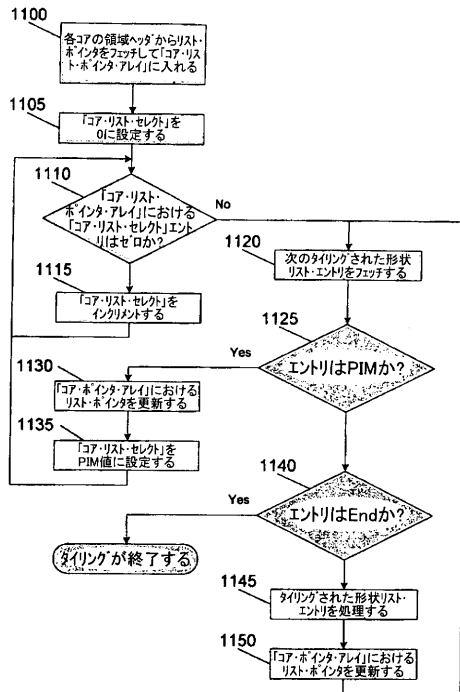
【図 9】



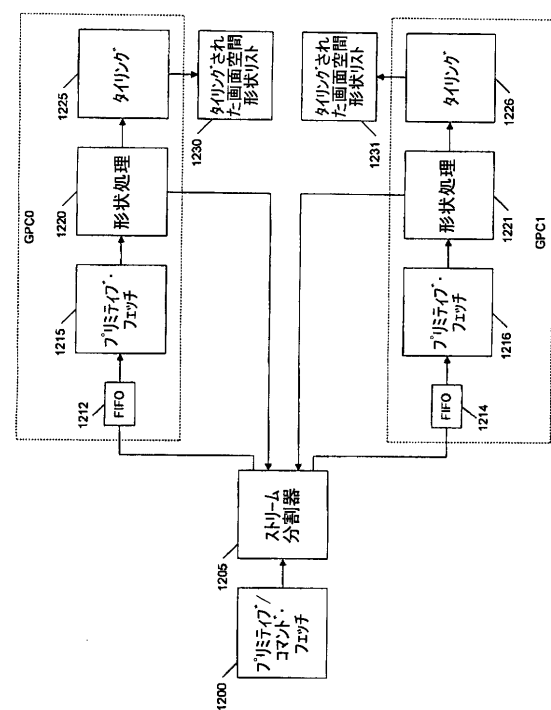
【図 10】



【図 11】



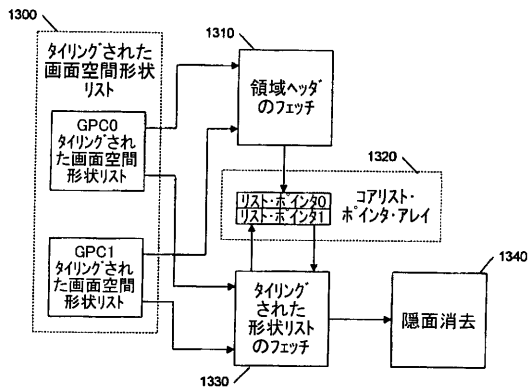
【図 12】



制御リスト及び「ハイライン・インターリーブ・マーカー」のラスタ化処理

ハイライン・インターリーブ・マーカーを用いた例示的な形状処理システム

【図 13】



PIMベースの形状システムにより生成されたタイルリストを
処理することができるタイルベースのラスタライズシステムのフロントエンドの例

フロントページの続き

(74)代理人 100109070

弁理士 須田 洋之

(74)代理人 100109335

弁理士 上杉 浩

(72)発明者 ホーソン ジョン ウィリアム

イギリス エイエル2 2 ジェイディー ハートフォードシャー セント オルバンズ パーク
ストリート ピルグリム クローズ 2

F ターム(参考) 5B057 AA20 CA13 CA17 CB13 CB16 CC01 CF10 CH04

5B080 AA14 BA00 CA01 CA03 CA04 CA05 GA00