

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5295246号
(P5295246)

(45) 発行日 平成25年9月18日 (2013.9.18)

(24) 登録日 平成25年6月21日 (2013.6.21)

(51) Int.Cl. F I
G O 6 T 15/00 (2011.01) G O 6 T 15/00 I O O A

請求項の数 30 (全 37 頁)

(21) 出願番号	特願2010-524560 (P2010-524560)	(73) 特許権者	501176037
(86) (22) 出願日	平成20年9月1日 (2008.9.1)		イマジネーション テクノロジーズ リミテッド
(65) 公表番号	特表2010-539579 (P2010-539579A)		イギリス ハートフォードシャー ダブリエューディー4 8エルゼット キングス
(43) 公表日	平成22年12月16日 (2010.12.16)		ラングリー ホーム パーク エステイト (番地なし)
(86) 国際出願番号	PCT/GB2008/002937	(74) 代理人	100092093
(87) 国際公開番号	W02009/034294		弁理士 辻居 幸一
(87) 国際公開日	平成21年3月19日 (2009.3.19)	(74) 代理人	100082005
審査請求日	平成23年8月16日 (2011.8.16)		弁理士 熊倉 禎男
(31) 優先権主張番号	0717781.9	(74) 代理人	100067013
(32) 優先日	平成19年9月12日 (2007.9.12)		弁理士 大塚 文昭
(33) 優先権主張国	英国 (GB)	(74) 代理人	100086771
			弁理士 西島 孝喜

最終頁に続く

(54) 【発明の名称】 3次元コンピュータ画像を生成する方法及びシステム

(57) 【特許請求の範囲】

【請求項1】

少なくとも1つの形状を各々が含む複数のオブジェクトを生成される画像が含む3次元シーンのコンピュータ画像を生成する方法であって、

a) プロセッサが、画像を複数の矩形区域に細分割し、かつ各矩形区域を複数のより小さい区域に細分割する段階と、

b) プロセッサが、前記画像内のオブジェクトの少なくとも一部に対して、該オブジェクト内の各形状に関する面データ及び頂点データと、該面データと該頂点データの間のポインタとを含むオブジェクトデータをメモリに記憶させる段階と、

c) プロセッサが、前記画像内の各小区域に対して前記小区域内の各オブジェクトに関する前記オブジェクトデータにリンクする制御データを導出する段階であって、前記制御データが、前記小区域内のオブジェクトの表示と、前記小区域内のオブジェクトに関する前記メモリに記憶されたオブジェクトデータへのポインタとを含む、段階と、

d) プロセッサが、前記制御データ及び前記オブジェクトデータから各矩形区域に関する深度データを導出する段階と、

e) プロセッサが、表示に向けて前記深度データから画像データ及びシェーディングデータを導出する段階と、

を含み、

前記段階c)が、

i) 段階b)で記憶された前記オブジェクトデータから2つ又はそれよりも多くのオ

10

20

プロジェクトに関するオブジェクトデータを同時に抽出する段階、

i i) 前記画像内の前記小区域のうちのどれが、段階 i) でオブジェクトデータが抽出された前記オブジェクトのうちの1つ又は全てを含むかを計算する段階、

i i i) 段階 i i) で計算された前記小区域の各々に対して、該小区域に含まれる前記オブジェクト又はその各々に関するオブジェクトデータにリンクする制御データセットを導出する段階、及び

i v) 段階 b) での全てのオブジェクトに対する制御データが導出されるまで段階 i) から i i i) を繰り返す段階、
を含むことを特徴とする方法。

【請求項 2】

段階 i i i) で導出された各制御データセットに対する最終メモリ位置において、ポインタが、前記小区域に関する次の制御データセットに対して設けられ、該最終メモリ位置は、キャッシュに記憶されることを特徴とする請求項 1 に記載の方法。

【請求項 3】

段階 b) は、オブジェクトデータを前記画像内の全てのオブジェクトに対してメモリに記憶させる段階を含む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】

段階 b) は、オブジェクトデータを前記画像内の所定数のオブジェクトに対してメモリに記憶させる段階を含み、

方法が、段階 e) の前に、前記画像内の全ての前記オブジェクトに対してオブジェクトデータを記憶するまで段階 b)、段階 c)、及び段階 d) を繰り返す段階を更に含む、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 5】

段階 i) は、段階 b) で記憶された前記オブジェクトデータから 2 つのオブジェクトに関するオブジェクトデータを抽出する段階を含むことを特徴とする請求項 1 から請求項 4 のいずれか 1 項に記載の方法。

【請求項 6】

少なくとも 1 つの形状を各々が含む複数のオブジェクトを生成される画像を含む 3 次元シーンのコンピュータ画像を生成するための装置であって、

a) 画像を複数の矩形区域に細分割し、かつ各矩形区域を複数のより小さい区域に細分割するための手段と、

b) 前記画像内のオブジェクトの少なくとも一部に対して、該オブジェクト内の各形状に関する面データ及び頂点データと、該面データと該頂点データの間のポインタとを含むオブジェクトデータをメモリに記憶させるための手段と、

c) 前記画像内の各小区域に対して制御データを導出するための導出手段であって、前記制御データが、前記小区域内の各オブジェクトに関する前記オブジェクトデータにリンクし且つ前記小区域内のオブジェクトの表示と前記小区域内のオブジェクトに関する前記メモリに記憶されたオブジェクトデータへのポインタとを含む、導出手段と、

d) 前記制御データ及び前記オブジェクトデータから各矩形区域に関する深度データを導出するための手段と、

e) 表示に向けて前記深度データから画像データ及びシェーディングデータを導出するための手段と、

を含み、

前記制御データ導出手段が、

i) 前記記憶させるための手段によって記憶された前記オブジェクトデータから 2 つ又はそれよりも多くのオブジェクトに関するオブジェクトデータを同時に抽出し、

i i) 前記画像内の前記小区域のうちのどれが、オブジェクトデータが抽出された前記オブジェクトのうちの 1 つ又は全てを含むかを計算し、

i i i) i i) で計算された前記小区域の各々に対して、該小区域に含まれる前記オブジェクト又はその各々に関するオブジェクトデータにリンクする制御データセットを導

10

20

30

40

50

出し、かつ

i v) b) での全てのオブジェクトに対する制御データが導出されるまで i) から i i) を繰り返す、
ように構成されていることを特徴とする装置。

【請求項 7】

生成される画像が複数のオブジェクトを含む 3 次元コンピュータ画像を生成する方法であって、

a) プロセッサが、画像を複数の個別区域に細分割する段階と、
 b) プロセッサが、前記画像内のオブジェクトの少なくとも一部に対して、オブジェクトデータをメモリに記憶させる段階と、
 c) プロセッサが、一度に 2 つの個別区域に対して、該 2 つの個別区域内の前記オブジェクトデータにリンクする制御データを導出する段階であって、前記制御データが、前記 2 つの個別区域内のオブジェクトを列挙し且つ前記 2 つの個別区域内の各オブジェクトについてそれぞれのオブジェクトデータへのリンクを含む、段階と、

d) プロセッサが、

i) 前記制御データ及び前記オブジェクトデータから第 1 の個別区域に関する深度データを導出し、かつ表示に向けて該深度データから該第 1 の個別区域に関する画像データ及びシェーディングデータを導出し、かつ

i i) 前記制御データ及び前記オブジェクトデータから第 2 の個別区域に関する深度データを導出し、かつ表示に向けて該深度データから該第 2 の個別区域に関する画像データ及びシェーディングデータを導出する、

ことを並行して行うことにより、前記 2 つの個別区域に 3 次元コンピュータ画像を生成する段階と、

e) 画像が全ての前記個別区域に対して生成されるまで段階 c) 及び段階 d) を繰り返す段階と、

を含むことを特徴とする方法。

【請求項 8】

各オブジェクトが、複数の頂点を有する複数の形状を含み、段階 c) で導出された前記制御データは、各オブジェクトに対して、該オブジェクトのどの形状が前記個別区域のどれに収まるかに関する指示、及び該オブジェクトのどの頂点が該個別区域のどれに収まるかに関する指示を含むことを特徴とする請求項 7 に記載の方法。

【請求項 9】

前記制御データは、ヘッダ部分及びペイロード部分を含むことを特徴とする請求項 8 に記載の方法。

【請求項 10】

前記ヘッダ部分は、前記オブジェクトのどの頂点が前記個別区域のどれに収まるかに関する前記指示を含むことを特徴とする請求項 9 に記載の方法。

【請求項 11】

前記オブジェクトの前記形状の一部が前記個別区域の各々に収まるか、又は該オブジェクトの該形状の全てが該個別区域の 1 つのみに収まる場合には、前記ペイロードは、該オブジェクトのどの形状が該個別区域のどれに収まるかに関する前記指示を含むことを特徴とする請求項 9 又は請求項 10 に記載の方法。

【請求項 12】

生成される画像が複数のオブジェクトを含む 3 次元コンピュータ画像を生成するための装置であって、

a) 画像を複数の個別区域に細分割するための手段と、

b) 前記画像内のオブジェクトの少なくとも一部に対して、オブジェクトデータをメモリに記憶させるための手段と、

c) 一度に 2 つの個別区域に対して、該 2 つの個別区域内の前記オブジェクトデータにリンクする制御データを導出するための導出手段であって、前記制御データが、前記 2 つ

10

20

30

40

50

の個別区域内のオブジェクトを列挙し且つ前記2つの個別区域内の各オブジェクトについてそれぞれのオブジェクトデータへのリンクを含む、導出手段と、

d) 前記2つの個別区域に3次元コンピュータ画像を生成するための画像発生器と、
を含む、

前記画像発生器は、第1の画像発生器と、それと並列の第2の画像発生器とを含み、

前記第1の画像発生器は、前記制御データ及び前記オブジェクトデータから第1の個別区域に関する深度データを導出し、かつ表示に向けて該深度データから該第1の個別区域に関する画像データ及びシェーディングデータを導出するように構成され、

前記第2の画像発生器は、前記制御データ及び前記オブジェクトデータから第2の個別区域に関する深度データを導出し、かつ表示に向けて該深度データから該第2の個別区域に関する画像データ及びシェーディングデータを導出するように構成される、

ことを特徴とする装置。

【請求項13】

生成される画像が複数のオブジェクトを含む3次元コンピュータ画像を生成する方法であって、

a) プロセッサが、画像を複数の個別区域に細分割する段階と、

b) プロセッサが、前記画像内のオブジェクトの少なくとも一部に対して、第1のエンジンによって、メモリページに分割されているメモリにオブジェクトデータを記憶させる段階と、

c) プロセッサが、段落b)でオブジェクトデータが記憶された各オブジェクトに対して、該オブジェクトデータが1つのメモリページ上に記憶された場合に、前記第1のエンジンによって、このメモリページアドレスを第2のエンジンに送り、又は該オブジェクトデータが2つのメモリページ上に記憶された場合に、該第1のエンジンによって、これらのメモリページアドレスを該第2のエンジンに送る段階と、

d) プロセッサが、前記画像内の各個別区域に対して、前記第2のエンジンによって、より小さな前記区域内の各オブジェクトに関する前記オブジェクトデータにリンクする制御データを導出する段階であって、前記制御データが、前記個別区域内のオブジェクトを列挙し且つ前記個別区域内の各オブジェクトについてそれぞれのオブジェクトデータへのリンクを含み、さらに、前記制御データが、段落b)で記憶された該オブジェクトデータが1つのメモリページ上に記憶された場合に前記1つのメモリページアドレスを含み、段落b)で記憶された該オブジェクトデータが2つのメモリページ上に記憶された場合に前記2つのメモリページアドレスを含む、段階と、

e) プロセッサが、前記制御データ及び前記オブジェクトデータから各個別区域に関する深度データを導出する段階と、

f) プロセッサが、表示に向けて前記深度データから画像データ及びシェーディングデータを導出する段階と、

を含むことを特徴とする方法。

【請求項14】

メモリマネージャによって、前記メモリのページを前記画像内の前記オブジェクトに割り当てる段階を更に含むことを特徴とする請求項13に記載の方法。

【請求項15】

前記メモリマネージャは、2ページのメモリを各オブジェクトに自動的に割り当てることを特徴とする請求項14に記載の方法。

【請求項16】

前記深度データを導出する段階e)は、前記個別区域内の各オブジェクトに関するオブジェクトデータにリンクする前記制御データが、1つ又は2つのいずれのメモリページアドレスを含むか否かを判断する段階を含むことを特徴とする請求項13から請求項15のいずれか1項に記載の方法。

【請求項17】

生成される画像が複数のオブジェクトを含む3次元コンピュータ画像を生成するための

10

20

30

40

50

装置であって、

- a) 画像を複数の個別区域に細分割するための手段と、
- b) 前記画像内のオブジェクトの少なくとも一部に対してオブジェクトデータをメモリに記憶させるための第1のエンジンと、
- c) 前記個別区域内の各オブジェクトに関する前記オブジェクトデータにリンクする制御データを導出するための第2のエンジンであって、前記制御データが、前記個別区域内のオブジェクトを列挙し且つ前記個別区域内の各オブジェクトについてそれぞれのオブジェクトデータへのリンクを含む、第2のエンジンと、
- d) 前記制御データ及び前記オブジェクトデータから各個別区域に関する深度データを導出するための手段と、
- f) 表示に向けて前記深度データから画像データ及びシェーディングデータを導出するための手段と、

10

を含み、

前記メモリは、メモリページに分割され、

前記第1のエンジンは、オブジェクトデータが記憶された各オブジェクトに対して、i) 該オブジェクトデータが1つのメモリページ上に記憶された場合に、このメモリページアドレスを前記第2のエンジンに送り、ii) 該オブジェクトデータが2つのメモリページ上に記憶された場合に、これらのメモリページアドレスを該第2のエンジンに送るよう構成され、かつ

前記制御データは、前記第1のエンジンによって記憶された前記オブジェクトデータが1つのメモリページ上に記憶された場合に、前記1つのメモリページアドレスを含み、又は該第1のエンジンによって記憶された該オブジェクトデータが2つのメモリページ上に記憶された場合に、前記2つのメモリページアドレスを含む、

20

ことを特徴とする装置。

【請求項18】

前記メモリのページを前記画像内の前記オブジェクトに割り当てるためのメモリマネージャを更に含むことを特徴とする請求項17に記載の装置。

【請求項19】

前記メモリマネージャは、2ページのメモリを各オブジェクトに自動的に割り当てるよう構成されることを特徴とする請求項17に記載の装置。

30

【請求項20】

前記深度データを導出するための手段は、前記個別区域内の各オブジェクトに関するオブジェクトデータにリンクする前記制御データが、1つ又は2つのいずれのメモリページアドレスを含むか否かを判断するように構成されることを特徴とする請求項17から請求項19のいずれか1項に記載の装置。

【請求項21】

生成される画像が複数のオブジェクトを含む3次元コンピュータ画像を生成する方法であって、

- a) プロセッサが、画像を複数の個別区域に細分割する段階と、
- b) プロセッサが、メモリページに分割され、かつルックアップテーブルを含むメモリを準備する段階と、

40

c) プロセッサが、メモリマネージャによって、前記ルックアップテーブル内の2つ又はそれよりも多くの連続部分を各個別区域に割り当て、該ルックアップテーブル内の各割り当てられた部分が、その個別区域に関するオブジェクトデータを記憶するのに利用可能なメモリページを指し示す段階と、

d) プロセッサが、前記画像内のオブジェクトの少なくとも一部に対して、第1のエンジンによって、各個別区域に関して、オブジェクトデータを、段階c)でその個別区域に割り当てられた前記メモリページに記憶させる段階と、

e) プロセッサが、各個別区域に対して、前記第1のエンジンによって、前記ルックアップテーブル内の前記2つ又はそれよりも多くの割り当て部分のメモリアドレスを第2の

50

エンジンに送る段階と、

f) プロセッサが、前記画像内の各個別区域に対して、前記第1のエンジンから送られた前記メモリアドレスを有する前記第2のエンジンによって、前記個別区域内の各オブジェクトに関する前記オブジェクトデータにリンクする制御データを導出する段階であって、前記制御データが、前記個別区域内のオブジェクトを列挙し且つ前記個別区域内の各オブジェクトについてそれぞれのオブジェクトデータへのリンクを含む、段階と、

g) プロセッサが、前記画像内の各個別区域に対して、前記第1のエンジンから送られた前記メモリアドレスによって、前記制御データ及び前記オブジェクトデータから深度データを導出する段階と、

h) プロセッサが、表示に向けて前記深度データから画像データ及びシェーディングデータを導出する段階と、
を含むことを特徴とする方法。

10

【請求項22】

生成される画像が複数のオブジェクトを含む3次元コンピュータ画像を生成するための装置であって、

a) 画像を複数の個別区域に細分割するための手段と、

b) メモリページに分割され、かつルックアップテーブルを含むメモリと、

c) 前記ルックアップテーブル内の2つ又はそれよりも多くの連続部分を各個別区域に割り当て、該ルックアップテーブル内の各割り当てられた部分が、その個別区域に関するオブジェクトデータを記憶するのに利用可能なメモリページを指し示すメモリマネージャと、

20

d) 前記画像内のオブジェクトの少なくとも一部に対して、かつ各個別区域に関して、オブジェクトデータを前記メモリマネージャによってその個別区域に割り当てされた前記メモリページに記憶するように構成された第1のエンジンと、

e) 前記ルックアップテーブル内の前記2つ又はそれよりも多くの割り当てられた部分の各個別区域に対するメモリアドレスを前記第1のエンジンから受け取るように構成され、かつ該メモリアドレスによって、その個別区域内の各オブジェクトに関する前記オブジェクトデータにリンクする制御データを導出するように構成された第2のエンジンであって、前記制御データが、前記個別区域内のオブジェクトを列挙し且つ前記個別区域内の各オブジェクトについてそれぞれのオブジェクトデータへのリンクを含む、第2のエンジンと、

30

g) 前記メモリアドレスによって、前記画像内の各個別区域に関して前記制御データ及び前記オブジェクトデータから深度データを導出するための手段と、

h) 表示に向けて前記深度データから画像データ及びシェーディングデータを導出するための手段と、
を含むことを特徴とする装置。

【請求項23】

生成される画像が複数のオブジェクトを含む3次元コンピュータ画像を生成する方法であって、

a) プロセッサが、画像を複数の個別区域に細分割する段階と、

40

b) プロセッサが、前記画像内のオブジェクトの少なくとも一部に対して、オブジェクトデータを第1のメモリに記憶させる段階と、

c) プロセッサが、前記オブジェクトデータから各個別区域に関する深度データを導出する段階と、

d) プロセッサが、リンク部分として割り当てられた部分を各ページが含む複数のメモリページを含む、前記深度データを記憶するための第2のメモリを準備する段階と、

e) プロセッサが、各個別区域に関する前記深度データを前記第2のメモリに記憶させる段階であって、

前記個別区域に関する前記深度データを前記リンク部分を除く1つのメモリページ上に記憶できる場合に、前記個別区域に関する深度データを該メモリページに書き込むと共

50

に該深度データが該メモリページのみに含まれることを該メモリページの前記リンク部分に示すことによって、又は、

前記個別区域に関する前記深度データを記憶するのに1つよりも多いメモリページを必要とする場合に、前記個別区域に関する深度データの一部を該個別区域に対する第1のメモリページに書き込み、該第1のメモリページの前記リンク部分に前記個別区域に対する別のメモリページへのポインタを示し、そして前記個別区域に関する別の深度データを前記別のメモリページに書き込むことによって、記憶させる段階と、

f) プロセッサが、別のオブジェクトデータを前記第1のメモリ内にロードして、既存の内容の少なくとも一部を置換する段階と、

g) プロセッサが、各個別区域に関して前記記憶された深度データを、この個別区域に対する各メモリページの前記リンク部分を読み取ること、及び、

前記深度データが前記メモリページのみに含まれることを前記リンク部分を示す場合に、該メモリページのみから前記深度データを読み取ること、又は、

前記リンク部分が別のメモリページへのポインタを含む場合に、前記メモリページ及び別のメモリページから前記深度データを読み取ること、によって取り出す段階と、

h) プロセッサが、前記別のオブジェクトデータから更新された深度データを導出し、該更新された深度データを記憶して以前に記憶された深度データを置換する段階と、

i) 前記メモリ内にロードされる更に別のオブジェクトデータがなくなるまで段階 e)、f)、g)、及び h) を繰り返す段階と、

j) プロセッサが、表示に向けて前記深度データから画像データ及びシェーディングデータを導出する段階と、

を含むことを特徴とする方法。

【請求項 24】

前記取り出す段階は、各メモリページの深度データを読み取るまで各メモリページのリンク部分を読み取る段階を含むことを特徴とする請求項 23 に記載の方法。

【請求項 25】

前記個別区域に関する前記深度データを前記リンク部分を除く1つのメモリページ上に記憶できる場合に、前記メモリページの前記リンク部分は0に等しいビットのみを含むことを特徴とする請求項 23 に記載の方法。

【請求項 26】

前記個別区域に関する前記深度データを記憶するのに1つよりも多いメモリページを必要とする場合に、前記第1のメモリページの前記リンク部分の前記ポインタは、31番目のビットが1に等しい32ビット値を含むことを特徴とする請求項 23 に記載の方法。

【請求項 27】

生成される画像が複数のオブジェクトを含む3次元コンピュータ画像を生成するための装置であって、

a) 画像を複数の個別区域に細分割するための手段と、

b) 前記画像内のオブジェクトの少なくとも一部に関するオブジェクトデータを記憶するための第1のメモリと、

c) 前記オブジェクトデータから各個別区域に関する深度データを導出するための手段と、

d) リンク部分として割り当てられた部分を各ページが含む複数のメモリページを含む、前記深度データを記憶するための第2のメモリと、

e) 各個別区域に関する前記深度データを前記第2のメモリに記憶させるための手段であって、

前記個別区域に関する前記深度データを前記リンク部分を除く1つのメモリページ上に記憶できる場合に、前記個別区域に関する深度データを該メモリページに書き込むと共に該深度データが該メモリページのみに含まれることを該メモリページの前記リンク部分

10

20

30

40

50

に示すことによって、又は、

前記個別区域に関する前記深度データを記憶するのに1つよりも多いメモリページを必要とする場合に、前記個別区域に関する深度データの一部を該個別区域に対する第1のメモリページに書き込み、該第1のメモリページの前記リンク部分に前記個別区域に対する別のメモリページへのポインタを示し、そして前記個別区域に関する別の深度データを前記別のメモリページに書き込むことによって、記憶させる手段と、

f) 別のオブジェクトデータを前記第1のメモリ内にロードして、既存の内容の少なくとも一部を置換するための手段と、

g) 各個別区域に関して前記記憶された深度データを、この個別区域に対する各メモリページの前記リンク部分を読み取ること、及び、

前記深度データが前記メモリページのみに含まれることを前記リンク部分が示す場合に、該メモリページのみから前記深度データを読み取ること、又は、

前記リンク部分が別のメモリページへのポインタを含む場合に、前記メモリページ及び別のメモリページから前記深度データを読み取ること、

によって取り出すための手段と、

h) 前記別のオブジェクトデータから更新された深度データを導出し、該更新された深度データを記憶して以前に記憶された深度データを置換するための手段と、

i) 表示に向けて前記深度データから画像データ及びシェーディングデータを導出するための手段と、

を含むことを特徴とする装置。

【請求項28】

前記取り出す手段は、各メモリページの深度データを読み取るまで各メモリページのリンク部分を読み取るように構成されていることを特徴とする請求項27に記載の装置。

【請求項29】

前記個別区域に関する前記深度データを前記リンク部分を除く1つのメモリページ上に記憶できる場合に、前記メモリページの前記リンク部分は0に等しいビットのみを含むことを特徴とする請求項27に記載の装置。

【請求項30】

前記個別区域に関する前記深度データを記憶するのに1つよりも多いメモリページを必要とする場合に、前記第1のメモリページの前記リンク部分の前記ポインタは、31番目のビットが1に等しい32ビット値を含むことを特徴とする請求項27に記載の装置。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、3次元コンピュータ画像を生成する方法及び装置に関する。

【背景技術】

【0002】

本出願人の英国特許第2281682号では、3Dシーン内の各プリミティブオブジェクトが無数の面の集合で定義されるような光景として見られる多角形に対する3Dレンダリングシステムが説明されている。画像が表示されることになる画面の各基本区域(例えば、ピクセル)は、それを通して光線を視点から3Dシーン内へ投影させる。次に、投影光線と各面の交点の位置が判断される。これらの交点から、いずれかの交差面がこの基本区域で可視であるか否かを次に判断することが可能である。この基本区域は、次に、判断の結果に応じて表示に向けてシェーディングされる。

【0003】

このシステムは、各々が面との交点計算を実行することができるいくつかのセルを含むパイプライン型プロセッサに実施することができる。従って、多数の面交点を同時に計算することができる。各セルには、交点検査を実施する面を定義する1組の係数がロードされる。

この構成に対する改善が、本出願人の英国特許第2298111号に説明されている。

10

20

30

40

50

この文献では、画像は、部分領域又はタイルに分割され、タイルは、逐次処理することができる。可変タイルサイズを用い、完全なプリミティブオブジェクトの周囲に境界ボックスを投影し、それによって境界ボックス内に収まるタイルのみが処理を必要とすることが提案されている。これは、適切なタイルサイズを選択するために可視画面上のプリミティブオブジェクトの配分を判断することによって行われる。タイル内の各オブジェクトは、面及び頂点を有するいくつかの三角形として記憶される。他の形状も可能である。次に、様々なプリミティブオブジェクトを定義する面が、表示リストとして公知のリストに記憶され、それによって各タイルに対して同一の面を記憶する必要性が回避されるが、これは、多くの面から成る1つのオブジェクトがいくつかのタイルに出現することができるからである。表示リスト内のプリミティブオブジェクトを識別するオブジェクト
10
ポイントも同様に記憶される。タイル毎に1つのオブジェクトポイントリストが存在する。次に、タイルは、各タイル内の全てのプリミティブオブジェクトが処理されるまで、上述の光線投射技術を用いて逐次レンダリングすることができる。これは、特定のタイル内で可視でないことが既知であるオブジェクトをレンダリングする必要がないので有用な方法である。

【0004】

このシステムに対する更なる改善は、本出願人の国際特許出願番号PCT/GB99/03707（公開番号WO2000/028483）及びPCT/GB2004/001076（公開番号WO2004/086309）、並びに本出願人の英国特許第2343603号に提案されており、特定のプリミティブオブジェクトを表示する必要のない境界
20
ボックス内のあらゆるタイルが、レンダリングの前に廃棄される。

【0005】

図1は、上述の既存のシステムに用いられる種類のプロセッサ101を示している。基本的に、3つの構成要素が存在する。タイルアクセラレータユニット(TA)103は、タイル分割作動を実施し、すなわち、適切なタイルサイズを選択して可視画面をタイルに分割し、タイル情報、すなわち、各タイルに関する3Dオブジェクトデータを表示リストメモリ105へと供給する。画像合成プロセッサ(ISP)107は、表示リストメモリ内の3Dオブジェクトデータを用いて、上述の光線/面交点検査を実施する。この検査により、可視画面の各基本区域に関する深度データが生成される。この後、ISP107から導出された画像データは、可視であると判断された面にテクスチャリング及びシェーディングデータを適用して画像及びシェーディングデータをフレームバッファメモリ111
30
に出力するテクスチャリング及びシェーディングプロセッサ(TSP)109に供給される。すなわち、ディスプレイの各基本区域の外観が、3D画像を表すように判断される。

【0006】

上述のシステムでは、レンダリングされるシーンの複雑さが増大する時に問題が発生する場合がある。複雑なシーンは、各タイルに対してより多くの3Dオブジェクトデータを表示リストメモリに記憶することを必要とし、これは、記憶要件が増大することを意味する。表示リストメモリが空間を使い切ってしまった場合、単純に、シーンの一部は、レンダリングされない場合があり、この種類の画像破損は、益々容認できなくなっている。

【0007】

この問題を解決するために、本出願人の国際特許出願PCT/GB01/02536（公開番号WO2001/095257）は、部分レンダリングの考えを提案している。タイルのレンダリングが完了する前に、システム(ISP及びTSP)の状態がメモリに記憶され、この状態は、レンダリングを完了するために後の時点で再ロードされる。この処理を「z/フレームバッファロード及び記憶」と呼ぶ。
40

【0008】

画面は、各タイルが画面の矩形領域から成るマクロタイルと呼ぶいくつかの領域に分割される。次に、表示リスト内のメモリが、ブロックに分割され、これらは、空き記憶領域リストに列挙される。空き記憶領域からのブロックは、次に、必要に応じてマクロタイルに割り当てられる。タイル分割作動は、各マクロタイルに関連付けられたデータを各プロ
50

ックに記憶する。(T A によって実施されるタイル分割作動は、表示リストメモリを埋め、従って、時によっては「メモリ割り当て」と呼ばれる。) 表示リストメモリが埋め尽くされるか又は何らかの所定の閾値に達すると、システムは、マクロタイルを選択し、z / フレームバッファロードを実行し、マクロタイルの内容をそれをz / フレームバッファ記憶作動を用いて保存する前にレンダリングする。すなわち、マクロタイルに関する深度データが、それまでに表示リスト内にロードされたデータに従って記憶される。そのようなレンダリングの完了時に、システムは、このマクロタイルに関連付けられたあらゆるメモリブロックを空にして、それによって更に別の記憶に向けてこれらのメモリブロックを利用可能にする。(レンダリング処理は、表示リストメモリ空間を解放するので、それは、「メモリ割り当て解除」として公知である。) 従って、各タイルに対するシーンは、いくつかのタイル分割作動とそれに続く部分レンダリングによって構成される。各部分レンダリングは、記憶された深度データを更新する。これは、メモリ消費に対して上限が課せられ、システムによって消費されるメモリ帯域幅も最小にされることを意味する。

10

【 0 0 0 9 】

部分レンダリングシステムに用いられる種類のプロセッサの一例を図2に示す。このプロセッサは、図1の改訂バージョンであることが分る。z バッファメモリ209は、z 圧縮 / 解凍ユニット211を通じてISP207にリンクされる。これは、システムが複雑なシーンをレンダリングし、かつ表示リストメモリ205が、特定のタイルに対して処理する必要がある全ての面を収容するほど十分大きくない時に作動を始める。表示リストには、それが実質的に一杯になるまで(又は、所定の閾値に達するまで)全てのタイルに対してT A 203によってデータがロードされることになる。しかし、これは、初期データの一部を表すのみであろう。画像は、ISP207によって一度に1つのタイルでレンダリングされる。各タイルに関する出力データは、テクスチャデータを用いてタイルをテクスチャリングするT S P 2 1 3に供給される。それと同時に、画像データは、未完成であるから、ISP207からの結果(すなわち、深度データ)は、一時的な記憶に向けて圧縮 / 解凍ユニット211を通じてバッファメモリ209に記憶される。残りのタイルのレンダリングは、次に、未完了の画像データを用いて、全てのタイルがレンダリングされてフレームバッファメモリ215及びz バッファメモリ209に記憶されるまで継続される。

20

【 0 0 1 0 】

次に、表示リストの最初の部分が廃棄され、更に別の画像データがその中に読み込まれる。ISP207によって各タイルに対して処理が順次実行される時に、z バッファメモリ209からのデータの関連の部分は、それを表示リストメモリ205からの新しい画像データと組み合わせることができるよう、z 圧縮 / 解凍ユニット211を通じてロードされる。次に、各タイルに関する新しい深度データがT S P 2 1 3に供給され、それは、データをフレームバッファ215へ供給する前にデータをテクスチャデータと組み合わせる。

30

【 0 0 1 1 】

この処理は、全ての画像データがレンダリングされ終わるまで、シーン内の全てのタイルに対して継続される。すなわち、z バッファメモリは、一時的記憶領域を埋めて、それによって特に複雑なシーンをレンダリングするのに必要になると考えられるものよりも小さい表示リストメモリを用いることを可能にすることが分る。圧縮 / 解凍ユニット211は、任意的であるが、それは、より小さいz バッファメモリを用いることを可能にする。更に、繰延ピクセル処理が有効にされて、あらゆる冗長なピクセル作動を排除する。

40

【 0 0 1 2 】

従って、国際特許出願番号P C T / G B 0 1 / 0 2 5 3 6 に解説されているように、表示リストメモリが埋め尽くされるか又はある一定の閾値に達した状態で、システムは、一部の表示リストメモリを解放するために、レンダリングするマクロタイルを選択する。この出願では、レンダリングするマクロタイルの選択は、いくつかの要因に依存し、例えば、最も多くのメモリを空き記憶領域へと解放して戻すことになるマクロタイルを選択する

50

ことができる。

【 0 0 1 3 】

このシステムに対する更なる改善が、本出願人の英国特許出願第 0 6 1 9 3 2 7 . 0 号に示されている。この構成に用いられるプロセッサの概略的な例を図 3 に示す。システム 3 0 1 は、図 2 のものと同様であり、T A 3 0 3 , I S P 3 1 1 , T S P 3 1 3、及び「フレームバッファ」3 1 5 を含むことが分る。しかし、この場合、「表示リストメモリ」3 0 5 及び「Z バッファメモリ」3 0 7 は、両方共に「パラメータメモリ」3 0 9 と呼ぶメモリの単一ヒープの一部である。パラメータメモリ内での「表示リスト」と「Z バッファ」の間の割り当ては、必要に応じてのものであり、それによってメモリ使用がより効率的になる。図 3 は、図 2 におけるような z 圧縮 / 解凍ユニットを示していないが、そのようなユニットを含めることができると考えられる。

10

【 0 0 1 4 】

更に、本出願人の英国特許出願第 0 6 1 9 3 2 7 . 0 号では、表示リストメモリは、ブロックを必要とする各マクロタイルのためのブロック、及び同じく全体リストを含む。オブジェクトデータは、1 つよりも多くのマクロタイルを横断することが可能である。この場合、オブジェクトデータは、1 つよりも多くのマクロタイルに存在するオブジェクトデータのみを収容する全体リストに割り当てられる。従って、表示リストは、マクロタイルに全体リストを加えたものへとグループ分けされる。マクロタイル内の全てのオブジェクト及び制御ストリームデータは、所定のマクロタイル内に属するタイルによってのみアドレス指定されることになり、1 つよりも多くのマクロタイルに属するあらゆるオブジェクトデータは、全体リストに存在し、これらのデータは、あらゆる制御ストリームからアドレス指定することができる。これは、同じくより効率的なメモリ使用及びアクセスに役立つ。

20

【 0 0 1 5 】

図 4 は、英国特許第 0 6 1 9 3 2 7 . 0 号に説明されているシステムの特定の例を示している。この構成は、「動的パラメータマネージャ」D P M 4 1 1 , I S P 4 0 7、「マクロタイル分割エンジン」4 0 3、「タイル分割エンジン」T E 4 0 5、及びメモリ 4 0 9 を含む。一般的には、M T E 4 0 3 は、各マクロタイルに対してプリミティブオブジェクトデータを生成し、これらのデータをメモリ 4 0 9 の適切な部分に入力する。T E 4 0 5 は、M T E 4 0 3 からのプリミティブオブジェクトデータを用いて各タイルに関する制御ストリームデータを生成し、これらのデータをメモリ 4 0 9 に入力する。従って、M T E と T E は、各マクロタイル内の各タイルに対する記憶域が適切なオブジェクトデータ記憶域を指すように、一緒にタイルアクセラレータとして作用する。I S P 4 0 7 は、オブジェクトデータを用いて深度データを導出し、これらのデータをメモリ 4 0 9 の z バッファ部分に記憶する。

30

【 0 0 1 6 】

次に、図 4 の作動のより詳細な説明を図 5 への参照も行いながら提供することにする。図 5 は、表示されるいくつかの三角形形状を示している。各三角形形状は、プリミティブ又は単純に形状として公知である。画面は、3 0 個のタイルに分割される。プリミティブは、2 つのプリミティブオブジェクトに分割される。第 1 のプリミティブオブジェクトは、頂点 5 0 1 , 5 0 3 , 5 0 5 , 5 0 7 , 及び 5 0 9 を有する。第 2 のプリミティブオブジェクトは、頂点 5 1 1 , 5 1 3 , 5 1 5 , 5 1 7 , 及び 5 1 9 を有する。すなわち、各プリミティブオブジェクトは、いくつかのプリミティブを含む。

40

【 0 0 1 7 】

第 1 段階では、頂点及びプリミティブのデータが入力される (4 0 1 において)。図 5 における三角形は、頂点 5 0 1 から 5 0 9 及び頂点 5 1 1 から 5 1 9 を有する 2 つのプリミティブオブジェクトへとグループ分けされる。プリミティブオブジェクトの各々は、M T E 4 0 3 により、メモリ空間を適切に割り当てる D P M 4 1 1 を用いてメモリ内に書き込まれ、メモリアドレスは、次のタイル分割段階へと渡される。

【 0 0 1 8 】

50

第2段階では、MTEは、頂点501から509及び頂点511から519を画面空間内へと不可視であるあらゆる頂点（この事例では存在しない）を画面外として除去しながら処理する。既に解説したように、プリミティブ及び頂点データをフェッチする上でメモリ帯域幅の影響を最小にするために、プリミティブは、マクロタイル内のプリミティブオブジェクトへとグループ分けされる。一般的に、各マクロタイルは、画面サイズの4分の1又は16分の1である。プリミティブオブジェクトがマクロタイルと交差する場合には、これらのプリミティブオブジェクトのデータは、全体リストに一度だけ書き込むことができる。従って、403では、プリミティブオブジェクト（最大数の頂点を有する）は、マクロタイルに関連付けられたメモリ409の一部に書き込まれる。各マクロタイルに関連付けられたメモリページは、要求に応じてDPM411によって割り当てられる。頂点及びプリミティブの数は、プリミティブオブジェクトのメモリアドレス（これらは、頂点ポインタとして公知である）と共にTE405に送られて、プリミティブオブジェクトが可視であるタイルの制御ストリームに適用される。

【0019】

第3段階では、TE405は、制御データを各タイルに書き込む。MTE403において書き込まれるプリミティブオブジェクトからの各プリミティブは、プリミティブオブジェクトの境界ボックスの内側にある各タイルと照合される。タイル内でいずれか可視のプリミティブが存在する場合には、制御データは、そのタイルに関連付けられた制御ストリームに書き込まれる。従って、図5における頂点501から509を有する第1のプリミティブでは、このプリミティブオブジェクトに関連付けられた制御データは、C1及びC6のようなタイルではなく、C2、C3、C4、及びC5（陰付き灰色の）のようなタイルのうちの一部にのみ書き込まれる。同様に、図5における頂点511から519を有する第2のプリミティブオブジェクトでは、制御データは、B4及びB5のようなタイルのうちの一部に対して書き込まれ、B6のような他のものには書き込まれない。この段階で書き込まれる制御データは、プリミティブオブジェクト内の頂点及びプリミティブの数に関するプリミティブオブジェクトヘッダ、及び書き込まれるプリミティブオブジェクトのメモリアドレスへの頂点ポインタを含む。DPM411による制御ストリームでは、別々のメモリ空間が、各タイルに割り当てられる。

【0020】

最終的に、第4段階では、ISP407によって3D画像処理が実施される。3D処理は、画面内の各タイルに対して実施される。タイル分割のためのデータ構造の概要を示している図6を参照すると、3D処理は、領域アレイ601からの各タイルに対して実施される。各タイルに対する制御データは、図6におけるいくつかの制御ストリーム0からMを含む。各制御ストリームは、プリミティブオブジェクトヘッダ及び頂点ポインタを含み、各タイルに対する制御データ内の制御ストリーム数は、各タイル内に収まるプリミティブオブジェクトの数に依存する。ISPは、各タイル603の制御ストリームを通して全探索を行い、制御データ605内の頂点ポインタによってメモリアドレスから頂点及びプリミティブのデータを読み取る。従って、図5を再度参照すると、タイルの一部は、いかなるプリミティブも含まず、B2のような一部のタイルは、1つのプリミティブオブジェクトだけからのプリミティブを含み、C2及びC3のような他のタイルは、両方のプリミティブオブジェクトからのプリミティブを含む。隠れ面除去、テクスチャリング、及びシェーディングのような画像処理作動は、タイル内で有効なプリミティブオブジェクトからの全てのプリミティブに対して実施される。

【0021】

上述の公知の構成は、多くの利点を有するが、一部の欠点を確かに有する。タイル分割システムは、制御データに関するメモリ書込みが1つの制御ストリームから別のものへとジャンプするように、各タイル内のプリミティブオブジェクトを処理する。これは、メモリバーストサイズを突破し、メモリアクセスの効率を低下させる。

また、大きいプリミティブ形状が画面上のタイルの多くを覆う時には、制御データは、多数回繰り返される。特定のメモリブロック内の頂点及びプリミティブのデータは、3D

10

20

30

40

50

画像処理中に多くのタイルに対して多数回読み取られる可能性がある。この繰返しによってメモリ帯域幅が拡大する。一方、所定のタイル内で、プリミティブオブジェクトからの頂点及びプリミティブのうちの一部だけしか可視ではなかった場合には、プリミティブオブジェクトデータ全体をフェッチするのは効率的ではない。

【0022】

マクロタイルに対して頂点及びプリミティブのデータに向けて用いられるメモリページは、要求に応じて割り当てられ、従って、必ずしも連続したページであるわけではない。特定のプリミティブオブジェクトからのデータがページ境界と交差するのを防ぐために、現在のページ上にプリミティブオブジェクト全体に対する十分な空間が存在しない場合には、空き記憶領域からの新しいページが割り当てられる必要がある。この割り当てによって浪費が生じる。また、パイプラインシステムでは、プリミティブオブジェクトの正確なサイズは、オブジェクト全体が実際に書き込まれるまでは既知ではなく、従って、開始時点で最大サイズを仮定する必要がある。この仮定によって更に別の浪費が生じる。

10

【0023】

図2を参照して冒頭で解説した部分レンダリングでは、圧縮/解凍ユニット211が、zデータを圧縮してそれをメモリ209内に書き込み始める時に、タイルに関する圧縮データのサイズが既知ではない点で類似の問題が存在する。タイルに関する圧縮データがメモリページと交差するのを回避するために、圧縮/解凍ユニット211は、タイルの開始時点で現在のメモリページ内に十分な空間が存在するか否かを検査する必要がある。この検査は、圧縮zデータに対して最大サイズを仮定する必要がある、それによって浪費が生

20

【先行技術文献】

【特許文献】

【0024】

【特許文献1】英国特許第2281682号

【特許文献2】英国特許第2298111号

【特許文献3】PCT/GB99/03707(WO2000/028483)

【特許文献4】PCT/GB2004/001076(WO2004/086309)

【特許文献5】英国特許第2343603号

【特許文献6】PCT/GB01/02536(WO2001/095257)

30

【特許文献7】英国特許出願第0619327.0号

【特許文献8】GB060619327.0

【発明の概要】

【発明が解決しようとする課題】

【0025】

本発明者は、上述の公知のシステムに様々な改善を加えることができると考えられることを見出した。

【課題を解決するための手段】

【0026】

本発明の第1の態様により、生成される画像が少なくとも1つの形状を各々含む複数のオブジェクトを含む3次元コンピュータ画像を生成する方法を提供し、本方法は、a)画像を複数の矩形区域に細分割し、各矩形区域を複数の小区域に細分割する段階と、b)画像内のオブジェクトの少なくとも一部に対して、オブジェクト内の各形状に関する面データ及び頂点データと、面データと頂点データの間のポイントとを含むオブジェクトデータをメモリに記憶する段階と、c)画像内の各小区域に対して、i)段階b)で記憶されたオブジェクトデータから、2つ又はそれよりも多くのオブジェクトに関するオブジェクトデータを抽出する段階、ii)画像内の小区域のうちのどれが、段階i)でオブジェクトデータが抽出されたオブジェクトのうちの1つ又は全てを含むかを計算する段階、iii)段階ii)で計算された小区域の各々に対して、小区域内に含まれるオブジェクト又はその各々に関するオブジェクトデータにリンクする制御データセットを導出する段階、及

40

50

び i v) b) における全てのオブジェクトに対して制御データが導出されるまで段階 i) から i i i) を繰り返す段階によって小区域内の各オブジェクトに関するオブジェクトデータにリンクする制御データを導出する段階と、 d) 制御データ及びオブジェクトデータから各矩形区域に関する深度データを導出する段階と、 e) 表示に向けて深度データから画像データ及びシェーディングデータを導出する段階とを含む。

【 0 0 2 7 】

制御データは、2つ又はそれよりも多くのオブジェクトから一度に導出されるので、メモリアクセスが改善される。段階 i i i) で導出される各制御データセットは、段階 i) でオブジェクトデータが抽出された小区域内に含まれる全てのオブジェクトのオブジェクトデータへのリンクを含む。

10

好ましくは、段階 i i i) で導出された各制御データセットにおける最終メモリ位置に、小区域に関する次の制御データセットへのポインタが設けられ、最終メモリ位置は、キャッシュに記憶される。すなわち、制御データは、段階 i) でオブジェクトデータが抽出された小区域内に含まれる全てのオブジェクトのオブジェクトデータへのリンクを含むことになるので、この小区域内に収まる1つよりも多くのオブジェクトが存在したとしても、キャッシュは、一度アクセスするだけでよい。

【 0 0 2 8 】

一実施形態では、段階 b) は、画像の全てのオブジェクトに対してオブジェクトデータをメモリに記憶する段階を含む。これは、記憶される全てのオブジェクトデータに対して十分なメモリ空間が存在する場合のことである。

20

【 0 0 2 9 】

代替実施形態では、段階 b) は、画像内の所定数のオブジェクトに対してオブジェクトデータをメモリに記憶する段階を含み、本方法は、画像内の全てのオブジェクトに対してオブジェクトデータを記憶し終わるまで、段階 e) の前に段階 b)、段階 c)、及び段階 d) を繰り返す段階を更に含む。所定数は、利用可能メモリサイズによって判断することができる。

1つの好ましい実施形態では、段階 i) は、段階 b) で記憶されたオブジェクトデータから2つのオブジェクトに関するオブジェクトデータを抽出する段階を含む。

【 0 0 3 0 】

本発明の第1の態様により、生成される画像が少なくとも1つの形状を各々含む複数のオブジェクトを含む3次元コンピュータ画像を生成するための装置も提供し、装置は、 a) 画像を複数の矩形区域に細分割し、各矩形区域を複数の小区域に細分割するための手段と、 b) 画像内のオブジェクトの少なくとも一部に対して、オブジェクト内の各形状に関する面データ及び頂点データと、面データと頂点データの間のポインタとを含むオブジェクトデータをメモリに記憶するための手段と、 c) 画像内の各小区域に対して、小区域内の各オブジェクトに関するオブジェクトデータにリンクする制御データを導出するために、 i) 記憶するための手段によって記憶されたオブジェクトデータから、2つ又はそれよりも多くのオブジェクトに関するオブジェクトデータを抽出し、 i i) 画像内の小区域のうちのどれが、オブジェクトデータが抽出されたオブジェクトのうちの1つ又は全てを含むかを計算し、 i i i) i i) で計算された小区域の各々に対して、小区域内に含まれるオブジェクト又はその各々に関するオブジェクトデータにリンクする制御データセットを導出し、かつ i v) b) における全てのオブジェクトに対して制御データが導出されるまで i) から i i i) を繰り返すように構成された導出手段と、 d) 制御データ及びオブジェクトデータから各矩形区域に関する深度データを導出するための手段と、 e) 表示に向けて深度データから画像データ及びシェーディングデータを導出するための手段とを含む。

30

40

【 0 0 3 1 】

本発明の第2の態様により、生成される画像が複数のオブジェクトを含む3次元コンピュータ画像を生成する方法を提供し、本方法は、 a) 画像を複数の個別区域に細分割する段階と、 b) 画像内のオブジェクトの少なくとも一部に対して、オブジェクトデータをメ

50

メモリに記憶する段階と、c) 一度に2つの個別区域に対して、2つの個別区域内のオブジェクトデータにリンクする制御データを導出する段階と、d) i) 制御データ及びオブジェクトデータから第1の個別区域に関する深度データを導出し、更に、表示に向けて深度データから第1の個別区域に関する画像データ及びシェーディングデータを導出し、かつ ii) 制御データ及びオブジェクトデータから第2の個別区域に関する深度データを導出し、更に、表示に向けて深度データから第2の個別区域に関する画像データ及びシェーディングデータを導出することを並行して行うことにより、2つの個別区域内に3次元コンピュータ画像を生成する段階と、e) 全ての個別区域に対して画像が生成されるまで段階c) 及び段階d) を繰り返す段階とを含む。

【0032】

d) において2つの個別区域が並行して処理されるので、処理は、より迅速に達成される。

各個別区域は、好ましくは、矩形である。2つの個別区域は、好ましくは、1つが左に、1つが右に隣接する矩形の個別区域である。

1つの好ましい実施形態では、各オブジェクトは、複数の頂点を有する複数の形状を含み、段階c) で導出された制御データは、各オブジェクトに対して、オブジェクトのどの形状が個別区域のうちのどれに収まるかに関する指示、及びオブジェクトのどの頂点が個別区域のうちのどれに収まるかに関する指示を含む。

形状及び頂点が何処に収まるかというこの指示は、深度データが導出される時に、処理されている個別区域内に実際に収まらない形状又は頂点に関するデータを取得するいかなる不要なデータフェッチも存在しないことを意味する。

【0033】

一実施形態では、制御データは、ヘッダ部分及びペイロード部分を含む。好ましくは、ヘッダ部分は、オブジェクトのどの頂点が個別区域のうちのどれに収まるかに関する指示を含む。

好ましくは、オブジェクトの形状のうちの一部が、個別区域の各々に収まるか、又はオブジェクトの形状のうち全てが、個別区域のうち1つのみに収まる場合には、ペイロードは、オブジェクトのどの形状が個別区域のうちのどれに収まるかに関する指示を含む。

【0034】

本発明の第2の態様により、生成される画像が複数のオブジェクトを含む3次元コンピュータ画像を生成するための装置も提供し、装置は、a) 画像を複数の個別区域に細分割するための手段と、b) 画像内のオブジェクトの少なくとも一部に対して、オブジェクトデータをメモリに記憶するための手段と、c) 一度に2つの個別区域に対して、2つの個別区域内のオブジェクトデータにリンクする制御データを導出するための導出手段と、d) 2つの個別区域内に3次元コンピュータ画像を生成するための第1の画像発生器を第2の画像発生器と並列に備える画像発生器とを含み、第1の画像発生器は、第1の個別区域に対して、制御データ及びオブジェクトデータから深度データを導出し、更に、表示に向けて深度データから第1の個別区域に関する画像データ及びシェーディングデータを導出するように構成され、第2の画像発生器は、第2の個別区域に対して、制御データ及びオブジェクトデータから深度データを導出し、更に、表示に向けて深度データから第2の個別区域に関する画像データ及びシェーディングデータを導出するように構成される。

【0035】

本発明の第3の態様により、生成される画像が複数のオブジェクトを含む3次元コンピュータ画像を生成する方法を提供し、本方法は、a) 画像を複数の個別区域に細分割する段階と、b) 画像内のオブジェクトの少なくとも一部に対して、第1のエンジンが、メモリページに分割されているメモリにオブジェクトデータを記憶する段階と、c) b) でオブジェクトデータが記憶された各オブジェクトに対して、オブジェクトデータが1つのメモリページ上に記憶された場合に、第1のエンジンが、このメモリページアドレスを第2のエンジンに送る段階、又はオブジェクトデータが2つのメモリページ上に記憶された場

10

20

30

40

50

合に、第1のエンジンが、これらのメモリページアドレスを第2のエンジンに送る段階と、d)画像内の各個別区域に対して、第2のエンジンが、b)で記憶されたオブジェクトデータが1つのメモリページ上に記憶された場合に1つのメモリページアドレスを含み、b)で記憶されたオブジェクトデータが2つのメモリページ上に記憶された場合に2つのメモリページアドレスを含む、小区域内の各オブジェクトに関するオブジェクトデータにリンクする制御データを導出する段階と、e)各個別区域に対して、制御データ及びオブジェクトデータから深度データを導出する段階と、f)表示に向けて深度データから画像データ及びシェーディングデータを導出する段階とを含む。

【0036】

オブジェクトが2つのメモリページに記憶されることは、データが2つのページにわたって延びる(データが複写されるのではなく)ことを意味する。b)において記憶されるオブジェクトデータは、単一のメモリページ上、又は2つのメモリページにわたって記憶することができるのでメモリ浪費が低減する。2つのメモリページアドレスが制御データ内に含まれるので、画像が処理される時に正しいメモリページを得ることができる。

【0037】

本方法は、メモリマネージャが、メモリのページを画像内のオブジェクトに割り当てる段階を更に含む。この場合、メモリマネージャは、2つのメモリページを各オブジェクトに自動的に割り当てることができる。実際に、オブジェクトデータが1つのメモリページしか必要としない場合には、第2のメモリページは、単純に次のオブジェクトデータセットに向けて用いられる。

好ましくは、深度データを導出する段階e)は、個別区域内の各オブジェクトに関するオブジェクトデータにリンクする制御データが1つ又は2つのいずれのメモリページアドレスを含むか否かを判断する段階を含む。

【0038】

本発明の第3の態様により、生成される画像が複数のオブジェクトを含む3次元コンピュータ画像を生成するための装置も提供し、装置は、a)画像を複数の個別区域に細分割するための手段と、b)画像内のオブジェクトの少なくとも一部に対して、オブジェクトデータをメモリに記憶するための第1のエンジンと、c)小区域内の各オブジェクトに関するオブジェクトデータにリンクする制御データを導出するための第2のエンジンと、d)各個別区域に対して、制御データ及びオブジェクトデータから深度データを導出するための手段と、f)表示に向けて深度データから画像データ及びシェーディングデータを導出するための手段とを含み、メモリは、メモリページに分割され、第1のエンジンは、オブジェクトデータが記憶された各オブジェクトに対して、i)オブジェクトデータが1つのメモリページ上に記憶された場合に、このメモリページアドレスを第2のエンジンに送り、かつii)オブジェクトデータが2つのメモリページ上に記憶された場合に、これらのメモリページアドレスを第2のエンジンに送り、制御データは、第1のエンジンによって記憶されたオブジェクトデータが1つのメモリページ上に記憶された場合に、1つのメモリページアドレスを含み、第1のエンジンによって記憶されたオブジェクトデータが2つのメモリページ上に記憶された場合に、2つのメモリページアドレスを含む。

【0039】

装置は、メモリのページを画像内のオブジェクトに割り当てるためのメモリマネージャを更に含むことができる。メモリマネージャは、2つのメモリページを各オブジェクトに自動的に割り当てるように構成することができる。

好ましくは、深度データを導出するための手段は、個別区域内の各オブジェクトに関するオブジェクトデータにリンクする制御データが1つ又は2つのいずれのメモリページアドレスを含むか否かを判断するように構成される。

【0040】

本発明の第3の態様により、生成される画像が複数のオブジェクトを含む3次元コンピュータ画像を生成する方法も提供し、本方法は、a)画像を複数の個別区域に細分割する段階と、b)メモリページに分割され、かつルックアップテーブルを含むメモリを準備す

10

20

30

40

50

る段階と、c)メモリマネージャが、ルックアップテーブル内の2つ又はそれよりも多くの連続部分を各個別区域に割り当て、ルックアップテーブル内の各割り当てられた部分が、その個別区域に関するオブジェクトデータを記憶するために利用可能なメモリページを指し示す段階と、d)画像内のオブジェクトの少なくとも一部に対して、第1のエンジンが、各個別区域に関するオブジェクトデータを段階c)でその個別区域に割り当てられたメモリページに記憶する段階と、e)各個別区域に対して、第1のエンジンが、ルックアップテーブル内の2つ又はそれよりも多くの割り当て部分のメモリアドレスを第2のエンジンに送る段階と、f)画像内の各個別区域に対して、第2のエンジンが、第1のエンジンから送られたメモリアドレスを用いて、小区域内の各オブジェクトに関するオブジェクトデータにリンクする制御データを導出する段階と、g)画像内の各個別区域に対して、第1のエンジンから送られたメモリアドレスを用いて、制御データ及びオブジェクトデータから深度データを導出する段階と、h)表示に向けて深度データから画像データ及びシェーディングデータを導出する段階とを含む。

10

【0041】

各個別区域に割り当てられた連続部分を有するルックアップテーブルを用いることにより、この個別区域に関するオブジェクトデータが記憶される実際のメモリページは、連続のものである必要はない。これは、メモリページが必ずしも連続するわけではない場合であっても、オブジェクトデータが2つのメモリページに及ぶことができることを意味する。それによってメモリ使用量が改善され、浪費が低減する。本明細書では、順序又は一連における次のものを意味するのに「連続」という用語を用いており、すなわち、ルックアップテーブル内の2つ又はそれよりも多い連続部分は、ルックアップテーブル内で一方が他方に続く隣接部分である。

20

【0042】

本発明の第3の態様により、生成される画像が複数のオブジェクトを含む3次元コンピュータ画像を生成するための装置も提供し、装置は、a)画像を複数の個別区域に細分割するための手段と、b)メモリページに分割され、かつルックアップテーブルを含むメモリと、c)ルックアップテーブル内の各割り当てられた部分が、その個別区域に関するオブジェクトデータを記憶するのに利用可能なメモリページを指し示し、ルックアップテーブル内の2つ又はそれよりも多くの連続部分を各個別区域に割り当てるためのメモリマネージャと、d)各個別区域に対して、画像内のオブジェクトの少なくとも一部に関するオブジェクトデータをメモリマネージャによってその個別区域に割り当てされたメモリページに記憶するように構成された第1のエンジンと、e)各個別区域に対して、ルックアップテーブル内の2つ又はそれよりも多くの割り当てられた部分のメモリアドレスを第1のエンジンから受け取るように構成され、更に、メモリアドレスを用いて、その個別区域内の各オブジェクトに関するオブジェクトデータにリンクする制御データを導出するように構成された第2のエンジンと、g)メモリアドレスを用いて、画像内の各個別区域に関する制御データ及びオブジェクトデータから深度データを導出するための手段と、h)表示に向けて深度データから画像データ及びシェーディングデータを導出するための手段とを含む。

30

【0043】

本発明の第4の態様により、生成される画像が複数のオブジェクトを含む3次元コンピュータ画像を生成する方法を提供し、本方法は、a)画像を複数の個別区域に細分割する段階と、b)画像内のオブジェクトの少なくとも一部に対して、オブジェクトデータを第1のメモリに記憶する段階と、c)オブジェクトデータから各個別区域に関する深度データを導出する段階と、d)各々がリンク部分として割り当てられた部分を含む複数のメモリページを含む、深度データを記憶するための第2のメモリを準備する段階と、e)個別区域に関する深度データが1つのメモリページ上に記憶された場合に、そうであることをこのメモリページのリンク部分内に示し、個別区域に関する深度データが1つよりも多くのメモリページ上に記憶された場合に、この個別区域に対する各メモリページのリンク部分を用いて、この個別区域に対する次のメモリページを指し示す、各個別区域に関する深

40

50

度データを第2のメモリに記憶する段階と、f)更に別のオブジェクトデータを第1のメモリ内にロードして、既存の内容のうち少なくとも一部を置換する段階と、g)各個別区域に対して、この個別区域に対する上述の又は各メモリページのリンク部分を読み取ることにより、記憶された深度データを取り出す段階と、h)新しいオブジェクトデータ及び記憶された深度データから最新の深度データを導出し、最新の深度データを記憶して過去に記憶された深度データを置換する段階と、i)メモリ内にロードされる更なるオブジェクトデータがなくなるまで段階e)、段階f)、段階g)、及び段階h)を繰り返す段階と、j)表示に向けて深度データから画像データ及びシェーディングデータを導出する段階とを含む。

【0044】

本発明の第4の態様により、生成される画像が複数のオブジェクトを含む3次元コンピュータ画像を生成するための装置も提供し、装置は、a)画像を複数の個別区域に細分割するための手段と、b)画像内のオブジェクトの少なくとも一部に関するオブジェクトデータを記憶するための第1のメモリと、c)オブジェクトデータから各個別区域に関する深度データを導出するための手段と、d)各々がリンク部分として割り当てられた部分を含む複数のメモリページを含む、深度データを記憶するための第2のメモリと、e)個別区域に関する深度データが1つのメモリページ上に記憶された場合に、そうであることをこのメモリページのリンク部分内に示し、個別区域に関する深度データが1つよりも多くのメモリページ上に記憶された場合に、この個別区域に対する各メモリページのリンク部分を用いて、この個別区域に対する次のメモリページを指し示す、各個別区域に関する深度データを第2のメモリに記憶するための手段と、f)更に別のオブジェクトデータを第1のメモリ内にロードして、既存の内容のうち少なくとも一部を置換するための手段と、g)各個別区域に対して、この個別区域に対する上述の又は各メモリページのリンク部分を読み取ることにより、記憶された深度データを取り出すための手段と、h)新しいオブジェクトデータ及び記憶された深度データから最新の深度データを導出し、最新の深度データを記憶して過去に記憶された深度データを置換するための手段と、i)表示に向けて深度データから画像データ及びシェーディングデータを導出するための手段とを含む。

本発明の一態様に関連して説明した態様は、本発明の別の態様にも適用可能である。

既存のシステムは、図1から6を参照して既に説明した。

本発明の例示的な実施形態は、図7から図11を参照して以下に説明する。

【図面の簡単な説明】

【0045】

【図1】第1の公知のレンダリング及びテクスチャリングシステムの概略図である。

【図2】図1のシステムに対して改善をもたらす第2の公知のレンダリング及びテクスチャリングシステムの概略図である。

【図3】図2のシステムに対して改善をもたらす第3の公知のレンダリング及びテクスチャリングシステムの概略図である。

【図4】図3のレンダリング及びテクスチャリングシステムの詳細図である。

【図5】タイル、プリミティブ、及びプリミティブオブジェクトを示すシーンの概略図である。

【図6】タイル分割のためのデータ構造の外観図である。

【図7】本発明の第1の態様の実施形態の方法の流れ図である。

【図8】本発明の第2の態様の実施形態による2つの半タイルを用いたタイル分割のためのデータ構造の外観図である。

【図8a】本発明の第2の態様の実施形態におけるプリミティブオブジェクトヘッダ構造の図である。

【図8b】本発明の第2の態様の実施形態における2つの16ビットプリミティブ半マスクが存在する場合のプリミティブワードマスク構造の図である。

【図8c】本発明の第2の態様の実施形態における1つの32ビットプリミティブマスクが存在する場合のプリミティブワードマスク構造の図である。

10

20

30

40

50

【図 8 d】本発明の第 2 の態様の実施形態における 1 つの 3 2 ビットプリミティブマスクが存在する場合の別のプリミティブワードマスク構造の図である。

【図 8 e】本発明の第 2 の態様の実施形態における 2 つの 3 2 ビットプリミティブマスクが存在する場合のプリミティブワードマスク構造の図である。

【図 9】プリミティブオブジェクトに割り当てられた 2 つの頂点ポイントを示す本発明の第 3 の態様の実施形態の概略図である。

【図 10】メモリページ割り当てを連続的にするルックアップテーブルの使用を示す本発明の第 3 の態様の代替実施形態の概略図である。

【図 11】本発明の第 4 の態様の実施形態による圧縮 z パッファにおけるページリンクの概略図である。

10

【発明を実施するための形態】

【0046】

上述のように、本発明の第 1 の態様によると、制御ストリームデータを導出する方法によってメモリ使用量を改善することができる手法が達成される。ここで、第 1 の態様の実施形態を以下に説明する。

図 4、図 5、及び図 6 を参照して上述したように、標準的な構成では、TE 405 は、MTE 403 からプリミティブオブジェクト（各々がいくつかのプリミティブを含む）を取得し、このプリミティブオブジェクトをレンダリングするのに必要な最少タイル数を計算する。例えば、図 5 の第 1 のプリミティブオブジェクトをレンダリングするのに必要なタイルは C 2、C 3、C 4、C 5、B 3、B 4、B 5、及び A 4 である。次に、この最少タイルリストは処理されて、プリミティブオブジェクトのアドレスへの頂点ポイントが、タイルに存在するプリミティブを表すプリミティブオブジェクトヘッダと共に動的割り当てメモリに制御ストリームとして書き込まれる。これを図 6 に示す。

20

【0047】

各タイルは、制御ストリームデータに向けて用いられる独自のメモリブロックを有し、これらのメモリブロックは、要求に応じて DPM 411 によって割り当てられる。特定のタイルに対する制御ストリームに追加される新しいデータが、DPM 411 によって割り当てられた現在のメモリブロックを超えることになる場合には、新しい割り当てが要求され、古い制御ストリームは、ストリームリンクを用いて新しい割り当てにリンクされる。MTE 403 が終了信号によってシーンの終了を示すと、全てのタイルが処理されて、有効領域内のタイルに対する制御システムが終了ワードで終了される。

30

【0048】

大きいプリミティブを含むプリミティブオブジェクトでは、プリミティブオブジェクトが 1 つよりも多くのマクロタイルを覆う時に、全体リストを用いることができるが、制御ストリームデータは、プリミティブが収まる全てのタイルに対して数回書き込まれることを必要とする可能性がある。極端な場合には、制御ストリームデータを画面上の全てのタイルに対して書き込まなければならない場合がある。制御ストリームデータのそのような繰返しは、効率的なメモリ使用をもたらさない。

【0049】

GB 060619327.0 から、TE に追加された小さいキャッシュである「テールポイントキャッシュ (TPC)」を用いてメモリアクセスを改善することができることは公知である。タイル内の制御ストリームの最後のメモリ位置が記憶され、次に、メモリアクセスする必要がある時に、「テールポイントキャッシュ」から読み取られる。それによって TE からの主メモリアクセスが低減する。TPC の付設により、必要な主メモリアクセスの回数は低減するが、1 つのタイルから次のものへの制御ストリーム間の頻繁なアドレス変更は改善されない。制御ストリームの書込みは配分されるので、TPC の付設は、幅広のバスを有するメモリインタフェースに対してはそれ程有効ではない。例えば、メモリインタフェースが 128 ビット幅であった場合には、2 つの 3 ビットワード制御データを各タイル内に書き込む段階は、メモリ書込みの殆どを最大バーストのものにしないことになる (32 ビット × 2 は 128 ビットを大きく下回る)。この場合、128 ビットの

40

50

バースト書込みで有効な64ビットのみを更新するために、バースト書込みの一部をマスクする必要がある(すなわち、未使用64ビットを)。メモリモジュールにおけるこの種のマスクされたバースト書込みには、性能低下が生じる可能性がある。

【0050】

本発明のこの実施形態では、TEは、MTEから一度に2つのプリミティブオブジェクト(1つではなく)を取得する。TEは、特定のタイルに対して2つのプリミティブオブジェクトを同時に処理する。従って、各プリミティブオブジェクトを取得して、このプリミティブオブジェクトが収まるタイルの全てを通じて作業し、次に、別のプリミティブオブジェクトを取得し、以降同様に続ける代わりに、TEは、2つのプリミティブオブジェクトを一度に取得し、それによってメモリ使用量及びアクセスが改善される。

10

【0051】

図5を参照すると、TEは、MTEから両方のプリミティブオブジェクトを取得する。TEは、片方又は両方のプリミティブオブジェクトが収まる各タイル(この事例では陰付きタイル)に関する制御ストリームデータを順次書き込む。片方のプリミティブオブジェクトだけが収まるタイル(例えば、タイルB5)に対しては、TEは、このプリミティブオブジェクトに関する制御ストリームデータを書き込む。制御ストリームの最後では、メモリ位置が、テールポインタキャッシュ(TPC)に記憶される。両方のプリミティブオブジェクトが収まるタイル(例えば、タイルC3)に対しては、TEは、第1のプリミティブオブジェクトに関する制御ストリームデータを生成する。次に、TEは、第2のプリミティブオブジェクトに関する制御ストリームデータを生成する。このタイルにおける両方のプリミティブオブジェクトに対する制御ストリームデータは、メモリと一緒に書き込まれる。別のタイルに対して処理を続ける前に、このタイルの処理にわたる全ての処理の最後において、制御ストリームの最終メモリアドレスがTPCに記憶される。

20

【0052】

この実施形態の一般的な方法を図7の流れ図に示す。タイル分割が2つのプリミティブオブジェクトに対して行われ、各プリミティブオブジェクトは、32個までの形状(プリミティブとも呼ぶ)を収容することができる。

すなわち、2つのプリミティブオブジェクトが収まるタイルでは、このタイルに関する制御ストリームの最後におけるアドレスは、従来のシステムでの2度ではなく、一度だけ読み取られる必要がある。これらのタイルに対しては、制御ストリームデータ書込みは、128ビットよりも小さい2回のマスクされた書込みではなく、128ビットの最大バースト書込みになる。TPCアクセスは、これらのタイルでは半分になる。本発明者は、一部の事例において制御ストリームデータに関する128ビットの最大バースト書込みを0%から25%に増大させることができ、一部の事例においてTPCキャッシュミスの百分率を12%から0.04%に低減することができることを見出した。

30

【0053】

上述の実施形態をTEがMTEから2つのプリミティブオブジェクトを一度に取得する場合で説明した。明らかに、本方法は、TEが一度に2つよりも多くのオブジェクトをMTEから取得するように拡張することができる。毎回得られるプリミティブオブジェクトの数は、レンダリングされるシーンの複雑さ、プリミティブオブジェクト自体の複雑さ(すなわち、これらのプリミティブオブジェクトがいくつの面及び頂点を含むか)、及びTEの処理機能に依存することになる。

40

【0054】

上述のように、本発明の第2の態様によると、2つの等しい3D画像処理パイプが2つのタイルに対して同時に機能することを可能にすることによって3D画像発生器の性能を改善する手法が達成される。これを図8に示す。

図8は、図4の簡略化したものであり、TE、ISP、及びメモリのみを示している。TE801は、制御データをメモリ809に書き込む。3Dパラメータフェッチ803は、頂点及びプリミティブのデータを読み取って画像処理を実施する。この実施形態では、この処理は、2つのサブISP/TSP805及び807によって並行して実施される。

50

この構成では、3D画像処理の性能は、二倍になると予想される。

図8のシステムを有効に作動させるために、TE801は、タイルのうちの2つに対して互いに制御データを書き込む。この書込みを811に略示している。本方法は、タイル分割制御ストリームデータにおける重複を大幅に低減し、従って、良好なメモリ帯域幅の改善を与える。

【0055】

各タイルサイズが16×16ピクセルであった場合には、TE801は、全てのプリミティブオブジェクトを元の16×16タイルにタイル分割する。しかし、TEは、制御ストリームデータ(811に示している)を32×16ピクセルのタイル、すなわち、2つの隣接する16×16タイルで書き込む。すなわち、各制御ストリームにおいて2つの16×16タイルが参照される。この2つのタイルの利点は、両方のタイル内に収まるオブジェクトを一度書き込むだけでよいので、書き込まれる制御ストリームデータの合計量が大きく低減することになる点である。一方、欠点も存在する。制御ストリームデータは、プリミティブオブジェクトの境界ボックスが、両方の16×16タイルと重なるという仮定に基づいている。プリミティブオブジェクト内には、例えば、32個までのプリミティブ及び16個までの頂点が存在する可能性があるが、これらのプリミティブ及び頂点の全てが特定の16×16タイルに収まることにはならない。プリミティブ及び頂点のうちの一部が、2つの16×16タイルのうち片方にしか収まらない場合には、これらの無効なプリミティブ及び頂点によってパラメータデータ参照情報をフェッチして処理することは浪費になる。別の言い方をすると、2つのタイルを一度に取得することにより、最初に小さいタイルを有するという利点が幾分減少する。

【0056】

この実施形態では、この問題は、制御ストリーム内で頂点マスク及びプリミティブマスクを用いて解決される。制御ストリームデータは、各プリミティブオブジェクトに対して、プリミティブオブジェクトヘッダ、及びそのプリミティブオブジェクトに関するオブジェクトデータ(プリミティブ及び頂点のデータを含む)のメモリアドレスを指し示す頂点ポインタを含む。この実施形態に用いられるマスクは、各16×16タイルに対して有効な頂点及びプリミティブ、すなわち、実際にそのタイル内に収まる頂点及びプリミティブを示している。頂点マスクは、頂点数及びプリミティブ数のような他の情報と共にプリミティブオブジェクトヘッダ内に含めることができる16ビットのマスクである。頂点マスクは、各32×16タイルに対して有効な頂点を定義する。通常は、16×16タイルの各々に対して1つずつの2つの32ビットプリミティブマスクが必要である。プリミティブマスクは、各16×16タイルに対して有効なプリミティブを定義する。プリミティブマスクは、ヘッダ内に含めることができ、又は制御ストリームに追加ワードとして加えることを必要とする場合がある。この実施形態では、両方の16×16タイルに対して1つの頂点マスクだけしか必要とされず、それに対して各16×16タイルに対して1つのプリミティブマスクを必要とすることに注意されたい。これは、頂点データがバーストで読み取られ、レンダリング処理がプリミティブに基づいており、従って、少数の不要頂点を読み取るのにそれ程コストが掛からないことによる。従って、空間を節約するために、32×16タイルに対して単一の頂点マスクを用いることができる。一方、不要なプリミティブを処理するには、より多くのコストが掛かり、従って、これを回避するために、より大きな空間を必要としますが、各16×16タイルに対してプリミティブマスクが用いられる。

プリミティブが多くのタイルに重なる時にメモリ帯域幅に対する大きな影響を回避するために、制御ストリーム内のマスクのデータサイズを最小に保つ必要がある。この実施形態は、頂点マスク及びプリミティブマスクが含まれる時に制御ストリームのサイズを縮小する符号化手法を用いる。

【0057】

既に解説したように、頂点マスク(及び場合によってはプリミティブマスク)は、プリミティブオブジェクトヘッダに含まれる。32ビットプリミティブオブジェクトヘッダに

10

20

30

40

50

おける符号化手法の詳細を表 1 に示す。

【 0 0 5 8 】

(表 1)

ビット	記号	説明
31 : 21
20 : 19	PF MASK K CONTROL	<p>頂点マスク及びプリミティブマスクが如何に詰め込まれているかを判断する 2 ビットフィールド。</p> <p>このフィールドは以下の符号化を有する。</p> <p>01 - PF MASK CTRL PRIM MASK PRES : 付加的な 1 つ又は 2 つの 32 ビットプリミティブマスクワードが、制御ストリームに存在し、32 ビットワードの数は PF PRM START に依存する。</p> <p>00 - PF MASK CTRL PACKED LEFT : プリミティブマスク及び頂点マスクは、ヘッダワード内に詰め込まれ、プリミティブマスクは、32 × 16 タイルの左 16 × 16 に付属する。</p> <p>11 - PR MASK CTRL PACKED RIGHT : プリミティブマスク及び頂点マスクは、ヘッダワード内に詰め込まれ、プリミティブマスクは、32 × 16 タイルの右 16 × 16 に付属する。</p> <p>10 - PF MASK CTRL FULL MASK : プリミティブオブジェクト内の全てのプリミティブは、タイルの両方の半分において参照される。</p>

10

20

18 : 17	PF_PRM_START	PF_MASK_CONTROLが、PF_MASK_CTRL_PRIM_MASK_PRESに等しくない場合：PF_PRM_STARTは、プリミティブマスク内でのPF_MASK_BYTE0のバイトオフセットを示している。 PF_MASK_CONTROLがPF_MASK_CNTRL_PRIM_MASK_PRESに等しい場合： 00-2つの16ビットプリミティブ半マスク (0xRL) が続く。 01-1つの32ビットプリミティブマスク (0xL) が続く。 10-1つの32ビットプリミティブマスク (0xR) が続く。 11-2つの32ビットプリミティブマスク (0xL、0xR) が続く。	
16	PF_VTM_START	PF_MASK_CONTROLが、PF_MASK_CTRL_PRIM_MASK_PRESと等しくない場合： 1である場合には、PF_MASK_BYTE1が上位8ビットの頂点マスクを含むことを示している。 0である場合には、PF_MASK_BYTE1が下位8ビットの頂点マスクを含むことを示している。 PF_MASK_CONTROLが、PF_MASK_CTRL_PRIM_MASK_PRESと等しくPF_PRM_STARTが00である場合： PF_VTM_STARTは、1である場合には、プリミティブマスクワードの両方がプリミティブ31:16に、それ以外の場合は、15:0に適用されることを示している。	10
15 : 8	PF_MASK_BYTE1	PF_VTM_STARTに依存して、かつPF_MASK_CONTROLが、PF_MASK_CTRL_PRIM_MASK_PRESに等しい場合に、頂点マスクの上位又は下位8ビットのいずれかである。	20
7 : 0	PF_MASK_BYTE0	PF_VTM_STARTに依存して、かつPF_MASK_CONTROLが、PF_MASK_CTRL_PRIM_MASK_PRESに等しい場合に、頂点マスクの下位8ビット又はプリミティブマスクの1バイトのいずれかである。	

10

20

30

【0059】

表1に示しているプリミティブオブジェクトヘッダの構造を図8aに例示している。

PF_MASK_CONTROL :

プリミティブマスク制御フラグ「PF_MASK_CONTROL」に対して2ビット(20:19)が用いられる。このフラグは、制御ストリーム内の何処に頂点マスク及びプリミティブマスクが含まれるかを判断する。この符号化では、制御ストリーム内の追加プリミティブマスクワードは任意的である。

PF_MASK_CONTROL = 01 = PF_MASK_CTRL_PRIM_MASK_PRESである場合には、PF_MASK_CONTROLは、プリミティブマスクワードが制御ストリームに含まれることを意味する。一方、PF_MASK_CONTROL = 01である場合には、プリミティブワードは、制御ストリームに含まれず、ヘッダに含まれる。

40

【0060】

PF_MASK_CONTROL = 01である前者の場合には、1つ又は2つのあらゆる追加プリミティブマスクワードを必要とすることができる。1つ又は2つのいずれの32ビットワードが存在するかは、PF_PRM_STARTに示されている。1つの追加プリミティブマスクワードしか持たない条件は、両方のプリミティブマスクが、下位16ビット又は上位16ビットセットのいずれかしか持たないということである。これは、両方のプリミティブオブジェクトが、左手16x16タイル又は右手16x16タイルのい

50

ずれかのみで有効であり、両方において有効ではないことを意味する。この場合、2つのプリミティブマスクワードが存在する場合は、片方は単純に0になり、従って、プリミティブワードマスクを完全に省略することができる。代替的に、両方のプリミティブオブジェクト内のプリミティブは、 16×16 タイルのうちの左又は右のいずれか片方においてのみ有効である。そうでなければ、2つの付加的なプリミティブマスクワードが必要とされ、これらは、2つの 16×16 タイル内で参照されるプリミティブに関する制御ストリームに追加される。

【0061】

PF_MASK_CONTROL 01である後者の場合を有する条件、すなわち、いずれのプリミティブワードも制御ストリームに含まれない条件は、プリミティブオブジェクト内の全てのプリミティブが現在の 32×16 タイル内で参照されるか（すなわち、プリミティブオブジェクト内の全てのプリミティブが両方の 16×16 タイル内に収まり、従って、プリミティブマスクワードの必要がない）、又はプリミティブ及び頂点の両方のマスクに対して1バイト分の非0ビットのみが存在するということである。この第2の条件は、頂点マスクワードが、1バイト分の非0ビットを有し、非0頂点マスクバイトが、PF_MASK_BYTE1内にあることを意味する。すなわち、頂点1から8が 32×16 タイル内で参照されないか、又は頂点9から16が 32×16 タイル内で参照されないかのいずれかである。（後に解説することになるが、PF_VTM_STARTが1の場合には、PF_VTM_STARTは、PF_MASK_BYTE1が頂点マスクの上位8ビットを含み、すなわち、頂点1から8が 32×16 タイル内で参照されないことを示している。PF_VTM_STARTが0である場合には、PF_VTM_STARTは、PF_MASK_BYTE1が頂点マスクの下位8ビットを含み、すなわち、頂点9から16は、 32×16 タイル内で参照されないことを示している。）PF_MASK_CONTROL 01である場合には、PF_MASK_BYTE0内には、1バイトのプリミティブマスクしか存在せず、従って、32ビットプリミティブマスクのうちの片方が0である必要がある。すなわち、プリミティブは、 16×16 タイルのうちの左又は右のいずれか片方においてのみ有効である。PF_PRM_STARTにおける値は、プリミティブマスク内でのPF_MASK_BYTE0のバイトオフセットを示している。例えば、PF_PRM_START = 1である場合には、32ビットプリミティブマスクは、 $0 \times 0000F700$ のように見えることになり、バイト $10 \times F7$ が、ヘッダにPF_MASK_BYTE0として挿入される。

【0062】

PF_MASK_CONTROL 01である後者の場合には、PF_MASK_CONTROL 01である時に、PF_MASK_BYTE0内に詰め込まれた1バイトプリミティブマスクに対して2つの可能性が存在する。

PF_MASK_CONTROL = 00 (PF_MASK_CTRL_PACKED_LEFT) である場合には、プリミティブマスク及び頂点マスクは、両方共にヘッダに含まれ、PF_MASK_BYTE0内に詰め込まれた1バイトプリミティブマスクは、 32×16 タイルの左半分に付属する。

PF_MASK_CONTROL = 11 (PF_MASK_CTRL_PACKED_RIGHT) である場合には、プリミティブマスク及び頂点マスクは、両方共にヘッダに含まれ、プリミティブマスクは、 32×16 タイルの右半分に付属する。

すなわち、PF_MASK_CONTROLのこれらの2ビットが、00, 11, 又は10である場合には、制御ストリームにいかなる追加プリミティブマスクも存在しない。

【0063】

PF_PRM_START :

プリミティブマスク開始点「PF_PRM_START」に対して2ビット(18:17)が用いられる。PF_PRM_STARTは、プリミティブマスクがヘッダに含まれるか(PF_MASK_CONTROL 01)、又は制御ストリームに追加マスクとして含まれるか(PF_MASK_CONTROL = 01)のいずれであるかに依存して別

10

20

30

40

50

々に用いられる。

PF_MASK_CONTROL = 01 である場合には、この2ビットは、プリミティブマスクがヘッダに含まれる時のプリミティブマスク内でのPF_MASK_BYTE0のバイトオフセットを示している。

PF_MASK_CONTROL = 01 である場合には、この2ビットは、以下の通りに制御ストリームにどの追加マスクが含まれるかを示す上で用いられる。

【0064】

これらの2ビット、すなわち、PF_PRIM_STARTが00である場合には、制御ストリームに2つの16ビットプリミティブマスクが含まれ、すなわち、左右の16×16タイルに対する32ビットプリミティブマスクの両方が、下位16ビットを0として又は上位16ビットを0として、そのいずれかで有する。PF_VTM_START = 0 である場合には、PF_VTM_STARTは、2つの16ビットプリミティブマスクがビット0から15におけるものであり、かつ両方の16×16タイルに対してプリミティブ1から16のみが参照されることを示している。PF_VTM_START = 1 である場合には、PF_VTM_STARTは、2つの16ビットプリミティブマスクがビット16から31におけるものであり、かつ両方の16×16タイルに対してプリミティブ17から32のみが参照されることを示している。

これらの2つの16ビットプリミティブマスクは、32ビットプリミティブマスクワードへと組み合わされる。これを下記の表2に示す。

【0065】

(表2)

表2-PF_PRIM_START=00		
ビット	記号	説明
31:16	PF_PRIM_MASKR	PF_VTM_STARTに依存して、タイルの右手半分に対するマスクの上位又は下位16ビットのいずれかである。
15:0	PF_PRIM_MASKL	PF_VTM_STARTに依存して、タイルの左手半分に対するマスクの上位又は下位16ビットのいずれかである。

【0066】

この場合のプリミティブマスクワードの構造を図8bに示す。ビット0から15は、左手16×16タイルに対する16ビットプリミティブ半マスクのために用いられる。ビット16から31は、右手16×16タイルに対する16ビットプリミティブ半マスクのために用いられる。

2つのビット、すなわち、PF_PRIM_STARTが01である場合には、左手16×16タイルに対して1つの32ビットプリミティブマスクが制御ストリームに含まれる。これを下記の表3に示す。

【0067】

(表3)

表3-PF_PRIM_START=01		
ビット	記号	説明
31:0	PF_PRIM_MASKL	左手のタイルを覆う32ビットプリミティブマスクである。

【0068】

この場合のプリミティブマスクワードの構造を図8cに示す。

2つのビット、すなわち、PF_PRIM_STARTが10である場合には、右手16×16タイルに対して1つの32ビットプリミティブマスクが制御ストリームに含まれる。これを下記の表4に示す。

【 0 0 6 9 】

(表 4)

表 4 - PF__PRIM__START=1 0		
ビット	記号	説明
3 1 : 0	PF__PRIM__ MASKR	左手のタイルを覆う 3 2 ビットプリミティブマスクである。

【 0 0 7 0 】

この場合のプリミティブマスクワードの構造を図 8 d に示す。

最後に、2つのビット、すなわち、PF__PRM__STARTが11である場合には、各16×16タイルに対して1つの2つの32ビットプリミティブマスクが制御ストリームに含まれる。これを下記の表 5 a 及び 5 b に示す。

【 0 0 7 1 】

(表 5 a)

表 5 a - PF__PRIM__START=1 1, ワード 1		
ビット	記号	説明
3 1 : 0	PF__PRIM__ MASKL	各ビットがプリミティブオブジェクト内の1つの三角形にマップされる 3 2 ビットプリミティブマスクである。プリミティブマスクは、各プリミティブオブジェクト内のどの三角形が現在の参照先であるかを識別する(タイルの左手半分)。

【 0 0 7 2 】

(表 5 b)

表 5 b - PF__PRIM__START=1 1, ワード 2		
ビット	記号	説明
3 1 : 0	PF__PRIM__ MASKL	各ビットがプリミティブオブジェクト内の1つの形状にマップされる 3 2 ビットプリミティブマスクである。プリミティブマスクは、各プリミティブオブジェクト内のどの形状が現在の参照先であるかを識別する(タイルの右手半分)。

【 0 0 7 3 】

この場合のプリミティブマスクワードの構造を図 8 e に示す。

【 0 0 7 4 】

PF__VTM__START :

表 1 を再度参照すると、頂点マスク開始点「PF__VTM__START」に対して1ビット(16)が用いられる。PF__VTM__STARTは、プリミティブマスクがヘッダに含まれるか(PF__MASK__CONTROL 01)、又は制御ストリーム内の追加マスクとして含まれるか(PF__MASK__CONTROL = 01)のいずれであるかに依存して別々に用いられる。

PF__MASK__CONTROL 01であり、すなわち、プリミティブマスクがヘッダに含まれる場合には、PF__VTM__STARTのビットが設定され、すなわち、=1である場合に、PF__VTM__STARTは、PF__MASK__BYTE 1(下記に解説することになる)が、16ビット頂点マスクの上位8ビットであるバイト1を反映することを示している。このビットが消去され、すなわち、=0である場合には、PF__VTM__STARTは、PF__MASK__BYTE 1が、16ビット頂点マスクの下位8ビットであるバイト0を含むことを示している。

【 0 0 7 5 】

PF__MASK__CONTROL = 01であり、すなわち、プリミティブマスクが制御ストリームに追加され、かつPF__PRM__START = 00である場合には、PF__V

10

20

30

40

50

TM_STARTが1である場合に、PF_VTM_STARTは、16ビットプリミティブマスクの両方が、プリミティブ31:16に適用されることを示している。上記以外の場合は、これらは、15:0に適用される。

【0076】

PF_MASK_BYTE1:

「PF_MASK_BYTE1」に対して8ビットが用いられる。PF_MASK_BYTE1は、PF_MASK_CONTROL及びPF_VTM_STARTのステータスに依存する。PF_MASK_CONTROL 01、すなわち、プリミティブマスクが、ヘッダ内のPF_MASK_BYTE0内に含まれ、かつPF_VTM_STARTが1に等しい場合には、これらの8ビットは、16ビット頂点マスクの上位8ビットである。一方、PF_MASK_CONTROL 01であり、かつPF_VTM_STARTが0に等しい場合には、これらの8ビットは、16ビット頂点マスクの下位8ビットである。

10

【0077】

PF_MASK_BYTE0:

「PF_MASK_BYTE0」に対して8ビットが用いられる。PF_MASK_BYTE0は、PF_MASK_CONTROL及びPF_PRM_STARTのステータスに依存する。PF_MASK_CONTROL = 01、すなわち、プリミティブマスクが、ヘッダではなく制御ストリームに含まれる場合には、16ビット頂点マスクは、PF_MASK_BYTE0及びPF_MASK_BYTE1内にあり、PF_MASK_BYTE0は、頂点マスクの下位8ビットである。一方、PF_MASK_CONTROL 01である場合には、プリミティブマスクからの1つの非0バイトが、PF_MASK_BYTE0内にある。この場合、この非0バイトプリミティブマスクのバイトオフセットは、PF_PRM_STARTからのものである。

20

【0078】

TEが、32×16タイル内のプリミティブオブジェクトからのプリミティブ及び頂点データを処理する場合には、TEは、オブジェクト内の全ての頂点を検査して、タイル内の有効頂点に対する頂点マスクを作成する。次に、TEは、16×16タイルの各々に収まるオブジェクト内の有効プリミティブに対するプリミティブマスクを生成する。上述の符号化手法は、次に、頂点マスク及びプリミティブマスクを制御ストリームデータ内に可能な限り小さいサイズで書き込む上で用いられる。

30

【0079】

制御ストリームの最後のテールポインタアドレスは、「テールポインタキャッシュ(TPC)」を通じて32×16タイルへのタイル分割の開始時点で読み戻される。テールポインタアドレスは、タイルの全ての処理が完了した後に初めて書き戻される。32×16タイルの両方の16×16タイルが、両方のプリミティブオブジェクト含む場合は(図5に戻り、例えば、タイルC3及びC4を参照されたい)、2つのプリミティブオブジェクトは、2つの16×16タイル内で処理されることになり、TPCが制御ストリームの最後に対するテールポインタで更新される前に、制御ストリームに書き込まれた適切な制御データを有する。それによって制御ストリームデータ書き込みからのメモリアクセスが改善され、TPCの性能が改善される。

40

【0080】

本発明のこの実施形態では、各32×16タイルの制御ストリームが一度に処理される。32×16タイルでは、タイル内で参照される頂点は、頂点マスクを用いてメモリからフェッチされる。このフェッチは、図8のパラメータフェッチ803によって実施される。従って、タイル内で参照されないフェッチ頂点に対していかなるメモリ帯域幅も追加されず、従って、上記に解説した問題は解決される。同様に、16×16タイルの各々に対するプリミティブマスクは、プリミティブデータをフェッチするのに用いられる。このフェッチも、図8の803によって実施される。上述の場合のように、タイル内で参照されないフェッチプリミティブに対していかなるメモリ帯域幅も追加されない。次に、左手の

50

16 × 16 タイルに対して適切なプリミティブ及び頂点が、サブISP / TSP 805 に送られ、右手の 16 × 16 タイルに対して適切なプリミティブ及び頂点が、サブISP / TSP 807 に送られる。

【0081】

本発明者は、32 × 16 ピクセルタイルに基づくタイル分割であるこの実施形態の方法が、16 × 16 ピクセルタイルに基づくタイル分割と比較して、付加的なプリミティブマスクワードを用いた場合でさえも平均で35%の制御ストリームデータ書き込みの低減をもたらすことを見出した。3Dパラメータフェッチにおける合計のデータ読取りに対しても同等のレベルの低減を達成した。

上述のように、本発明の第3の態様によると、メモリ空間を用いるより効率的な手法が達成される。次に、第3の態様の実施形態を以下に説明する。

【0082】

既に解説したように、既存のシステムでは、頂点及びプリミティブデータに対するマクロタイルに用いられるメモリページは、要求に応じて割り当てられ、従って、必ずしも連続するページであるわけではない。特定のプリミティブオブジェクトからのデータがページ境界を超えるのを防ぐために、プリミティブオブジェクト全体に対して現在のページ上に不十分な空間しか存在しない場合には、空き記憶領域から新しいページを割り当てる必要がある。

【0083】

特定のプリミティブオブジェクトに含まれる頂点及びプリミティブの最大数には制限が存在する(例えば、16個の頂点及び32個のプリミティブ)。上述のように、MTEは、プリミティブオブジェクトがメモリのページ境界に架からないことを確実にすべきである。これは、メモリページが、連続する順序で割り当てられることが保証されないからである。頂点をプリミティブオブジェクトに割り当てるパイプラインのフロントエンドは、このプリミティブオブジェクトに残されたいかなる空間量も既知ではないので、上述のことは困難である。一方、メモリを割り当て、実際にプリミティブオブジェクトデータをメモリに入力するパイプラインのバックエンドは、オブジェクト内の頂点数に対していかなる制御も持たない。現在のメモリページにプリミティブオブジェクト内の最大頂点数を保持するのに十分なメモリ空間が存在するか否かの検査が、十分な空間が存在しないことを明らかにした場合には、バックエンドにおいて新しいメモリページの割り当てが実施される。頂点データのサイズは既知であり(X、Y、Z座標に加えて、基本色、オフセット色、テクスチャ座標、テクスチャ座標がいくつあるか等に関する更に別のデータがある場合がある)、プリミティブオブジェクトに必要な最大メモリ空間は、容易に計算することができる。従って、例えば、プリミティブオブジェクト内の最大頂点数が16である場合には、バックエンドは、頂点数が16であることを常に仮定し、フロントエンドは、1つのプリミティブオブジェクト内に実際に16個までの頂点を含むことができることを常に仮定する。一方では、メモリ帯域幅に関して優れた効率を生じる頂点間の最大共有量を得ることができるので、上述のことは有利である。しかし、他方では、例えば、このメモリページ内に15個の頂点に対する空間だけしか残っていない場合には、新しいページをバックエンドで開始する必要があるが、それにも関わらず、フロントエンドで割り当てられる頂点の数は実際には14である可能性がある。しかし、バックエンドは、これを知る由もなく、これに対していかなる制御も持たない。それによって明らかにメモリ浪費が生じる。このメモリ浪費によるコストは、頂点サイズ、すなわち、特定の頂点に関連付けられた追加データ(基本色、オフセット色、テクスチャ座標などのような)と共に変化する。

【0084】

本発明の第3の態様のこの構成は、この問題を解決するための2つの実施形態を提供する。提案する第1の実施形態は、MTEによってプリミティブオブジェクトに対して2つのメモリページを割り当てることを可能にする。すなわち、特定のプリミティブオブジェクトは、2つのメモリページにわたることができる。従って、開始時点で、MTEによってプリミティブオブジェクトがメモリ内に書き込まれる時に、メモリマネージャは、2つ

10

20

30

40

50

のメモリページをそのマクロタイルに割り当てる。(1ページしか実際には必要ではないことになった場合には、第2のページは、単純に次の書込みに用いられる。)全てのプリミティブオブジェクトに対して、頂点及びプリミティブの数が、プリミティブオブジェクトのメモリアドレス(頂点ポインタ)と共にTEに送られて、プリミティブオブジェクトが可視であるタイルの制御ストリームに追加される。MTEがプリミティブオブジェクトデータをメモリ内に連続的に書き込む時には、MTEは、メモリ書込みがページ境界に架かるか否かを検査する。書込みが、1つのメモリページ範囲に収容されることになる場合には、このメモリページに対する頂点ポインタのみがTEに渡されることになる。一方、書込みがページ境界を超え、プリミティブオブジェクトデータが2つのメモリページ上に存在することになる場合には、第1のページ及び第2のページに対する両方の頂点ポインタがTEに送られる。

10

【0085】

MTEによってTEに送られるデータが、2つの頂点ポインタを含む場合には(MTE書込みが、上述のようにページ境界を超えたために)、TEは、このタイルに対する制御ストリーム内のリンクページのアドレスへの追加頂点ポインタを書き込むことになる。これを図9に略示している。

図9では、制御ストリームは、第1のプリミティブオブジェクトに対してプリミティブオブジェクトヘッダを含み、次に、このプリミティブオブジェクトに対する頂点ポインタを含む。従って、このプリミティブオブジェクトに関するオブジェクトデータは、第1のメモリページ上にのみ記憶される。しかし、第2のプリミティブオブジェクトでは、制御ストリームは、プリミティブオブジェクトヘッダ、頂点ポインタ、及び頂点リンクポインタを含む。この頂点リンクポインタは、第2のメモリページへの頂点ポインタであり、頂点ポインタが第1のメモリページを指し示すのに対して、頂点リンクポインタは、第2のメモリページを指し示すことが分る。従って、このプリミティブオブジェクトに関するオブジェクトデータは、第1のメモリページ及び第2のメモリページという2つのメモリページにわたっている。

20

【0086】

このタイルに対して画像処理が実施されている時に、制御ストリームは構文解析され、ISPは、特定のプリミティブオブジェクトに対して2つの頂点ポインタが存在することを把握する。これらの頂点リンクページポインタは、パイプラインの下流へと渡される。メモリアドレスがページ境界を超えた場合には、メモリからプリミティブ及び頂点データをフェッチする上で第2の頂点ポインタからのリンクページアドレスが用いられる。プリミティブオブジェクトサイズの上限、及びメモリページが割り当てられる手法により、MTE書込みは、最大2ページしか必要としないことになる、すなわち、1つ又は2つの頂点ポインタしか存在しないことに注意されたい。

30

【0087】

提案する第2の実施形態は、ページが連続することを保証する追加論理、及び付加的な小さいキャッシュを用いる「動的パラメータ管理(DPM)」を提供する。ページが連続することが保証される場合には、所定のマクロタイルに関する書込みが2ページにわたることを容認することができる。

40

ページを各マクロタイルに割り当てられたままに保つために、MTEが連続書込みを行う時に、DPMは、一部分が各マクロタイルに割り当てられた1つ又は複数のルックアップテーブルページを維持する。MTEに割り当てられたページは、連続メモリページアドレスではなく、ルックアップテーブル内の連続インデックスである。各マクロタイルに対して、ルックアップテーブル内には、データが書き込まれる実際のメモリページへのポインタを与える部分が存在する。ルックアップテーブル内のこれらの部分は連続し、この連続性は、これらの部分が開始時点にDPMによって割り当てられることから得ることができる。

【0088】

各マクロタイル(及び用いられている場合は全体リスト)に対して、MTEへのページ

50

割り当てのために、DPM内でカウンタが維持される。MTEによって書き込まれるデータに対して第1のページがマクロタイルに割り当てられる時に、同様に一部分が、そのマクロタイルに関するルックアップテーブル内に割り当てられる。マクロタイルに割り当てられた第1のページのアドレスは、このマクロタイルに関するルックアップテーブルページ内のこの部分の位置0に書き込まれる。このマクロタイルに対するその後の割り当ては、このマクロタイルに関するルックアップテーブルページ内のこの部分の位置1, 2等に書き込まれることになる。従って、実際の割り当てページが連続でない可能性があったとしても、ルックアップテーブルのこれらの部分は連続であり、従って、マクロタイルに関する書き込みデータは、1つよりも多くのページにわたることができる。上述のルックアップテーブル部分が一杯になった状態で、新しい部分がルックアップテーブル内に割り当てられ、この位置は、更に別のページアドレス割り当てで埋められる。全てのマクロタイルは、この手法で取り扱われる。最大画面レンダリングが実施された状態で、全てのメモリページ及びルックアップテーブルページは解放される。

10

【0089】

従って、MTEは、一方がデータの書き込まれる実際のアドレス、他方がこのマクロタイルに関するルックアップテーブルページの上述の部分のアドレス（基本アドレス）という2つのアドレスをDPMから受け取る。マクロタイルに関する連続アドレスであるルックアップテーブル内のアドレスは、TEに渡される頂点ポインタ（すなわち、オブジェクトデータに対するメモリアドレス）として用いられ、データが記憶される実際のアドレスは、データをメモリ内に書き込む上で用いられる。

20

【0090】

ルックアップテーブルメモリページの基本アドレスは、レジスタ内に保持することができる。3D画像処理モジュール（ISP）は、各タイル内のプリミティブオブジェクトを処理する前にこの基本アドレスを読み取る。各タイル内のプリミティブオブジェクトの処理中に、ルックアップテーブルページの上述の部分のアドレスは、このルックアップテーブルが属するマクロタイルのディレクトリエントリからフェッチされる。制御ストリームから読み戻された頂点ポインタは、実際のデータが書き込まれたメモリアドレスを得るのに用いられる。従って、連続ルックアップテーブルアドレスは、不連続メモリアドレスへと変換される。連続ルックアップテーブルアドレスと不連続メモリアドレスの間のこの変換をより効率的にするために、これらのアドレスにアクセスを行う上でキャッシュを用いることができる。

30

【0091】

図10は、この実施形態を略示している。第1のルックアップテーブルを1001に示している。ルックアップテーブルに対する基本アドレスが存在する。この基本アドレスは、各マクロタイルMT0, MT1...に対して、各々が2バイトの32個までのエントリ、すなわち、合計64バイトを含むルックアップテーブル1001を指し示す。（図10では、簡略化のためにMT7に関するエントリのみ示しているが、これは、全てのマクロタイルに当て嵌まる。）この実施形態では、16個のマクロタイルに加えて全体リスト（1つよりも多くのマクロタイルを横断するオブジェクトデータの記憶のための）が存在し、従って、 $17 \times 64 \text{ バイト} = 1088 \text{ バイト}$ が生じる。この実施形態における各メモリページは、4096バイトであり、従って、全てのマクロタイルに全体リストを加えたものに対するエントリは、ページ内に容易に収まる。

40

【0092】

各32個の2バイト部分は、各々のサイズが4096バイトであり、すなわち、メモリページを占有する別のルックアップテーブル1003を指し示す。これを矢印Aで示している。上述のルックアップテーブル部分は開始時点で割り当てられるのに対して、これらの追加ページは、より効率的なメモリ使用に向けて要求に応じて割り当てられる。各4096バイトのテーブル1003は、2048個の2バイト変換情報を含む。これは、仮想ドメイン1（ルックアップテーブル1003）から仮想ドメイン2（メモリページ1005）に変換することができることを意味する。各2バイト部分は、4096バイトのペー

50

ジ 1 0 0 5 を参照する。これを矢印 B で示している。従って、2 0 4 8 × 4 0 9 6 バイト = 約 8 M B が存在する。3 2 個のこれらが、各マクロタイルに対して存在し、すなわち、3 2 × 8 M B = 2 5 6 M B である。従って、各マクロタイルは、2 5 6 M B の仮想アドレス空間を保持することができる。

【 0 0 9 3 】

ここで、この実施形態では、メモリの合計量は、2 5 6 M B であり、従って、全てのデータが 1 つのマクロタイルに存在したとしても、全体の空間は、連続的にアドレス指定することができる。入って来る仮想アドレスは、1 6 ビットから成る。マクロタイル番号及び上位 5 ビットは、第 1 のテーブル (ディレクトリと呼ぶ) をアドレス指定し、3 2 個の他のテーブルのうちのどれをアクセスするかを示す上で用いられる。下位 1 1 ビットは、2 0 4 8 個の変換値のうちのどれを実際に用いるべきかを示している。従って、連続するアドレスと仮想アドレスとの間で変換を行うための段階は、1) 適切なマクロタイルを選択する段階、2) 上位 5 ビットを用いて正しい第 2 のルックアップテーブルを選択する段階、及び 3) 1 1 個の下位ビットを用いて 2 0 4 8 個のメモリページのうちのどれが実際の仮想アドレスであるかを選択する段階である。

10

【 0 0 9 4 】

この実施形態では、連続アドレス及び仮想アドレスは、D P M によって T A に供給される。データが書き込まれるアドレスが仮想アドレスであり、制御ストリームに組み込まれるアドレスは連続のものである。ルックアップテーブルは、連続アドレスを仮想アドレスへと変換するのに I S P 及び T S P パラメータフェッチによって用いられる。この第 2 の変換を効率的にするために、このルックアップテーブルはキャッシュに入れられ、このキャッシュは、性能を維持するのに必要である。

20

【 0 0 9 5 】

本発明者は、これらの実施形態を適用した後に、N T E によって用いられるメモリ空間の平均で 1 5 % の低減を達成したことを見出した。2 つの解決法の間差は、第 1 の解決法 (2 つの頂点ポインタを用いた) は、2 つの新しいキャッシュを追加する必要がないことから実装が小さくなることである。しかし、この解決法は、パラメータフォーマットを変更し、複雑さを増すことを要求する。第 2 の解決法 (ルックアップテーブルを用いた) は、2 つのページにわたって書込みを行うことを可能にし、システムの残りの部分に影響を及ぼすことなく実施することができる。また、第 2 のポインタという負荷の増大も存在しない。

30

【 0 0 9 6 】

上述の本発明の第 3 の態様の実施形態では、オブジェクトデータに対するメモリ割り当ての問題に対処した。特定のプリミティブオブジェクトからのデータがページ境界線を超えるのを防ぐために、プリミティブオブジェクト全体に対して現在のページ上に不十分な空間しか存在しなかった場合には、空き記憶領域から新しいページを割り当てる必要がある。z バッファメモリに記憶される深度データにおいて類似の問題が発生する可能性があり (図 2 を参照されたい)、ここで、これに対して更に説明を行うことにする。

【 0 0 9 7 】

図 2 は、部分レンダリングが採用された場合のプロセッサの例を示している。部分レンダリングは、オブジェクトデータのうちの一部を記憶するだけでよく、更に、部分レンダリングは、表示リストメモリ 2 0 5 を解放するように実施することができることから複雑なシーンに対して有効である。得られる深度データは、z バッファメモリ 2 0 9 に記憶され、後のオブジェクトデータと組み合わせられ、最終的に全体画像が生成される。図 2 に示しているように、圧縮 / 解凍ユニット 2 1 1 を用いることができる。

40

【 0 0 9 8 】

画像が最終的に正しく生成されることを保証するために、メモリ 2 0 9 内の深度データを部分レンダリングが実施されたマクロタイル内のタイル上に記憶する必要がある。深度データを記憶するのに用いられる z バッファメモリ 2 0 9 内の空間は、要求に応じてメモリマネージャ (例えば、D P M) によって割り当てられ、特定のタイルに対するメモリア

50

ドレスが保存される。次に、更に別のオブジェクトデータが表示リストメモリ内にロードされ、次に、更に別の部分レンダリングが実施される。この段階では、更に別の深度データは、タイルに関連付けられたメモリアドレス内に置かれる。最終画像は、全ての部分レンダリングの後に正しくレンダリングされることになり、zバッファメモリのメモリページを解放することができる。

【0099】

圧縮/解凍ユニット211が用いられる場合には、ユニット211内にはタイルに関する圧縮深度データを保持するいかなるバッファも存在せず、従って、ユニットは、圧縮しながらデータをメモリ内に書込み、解凍しながらメモリからデータを読み取る。タイル内の圧縮zデータがメモリページを超えることを回避するために、タイルの開始時点に、圧縮/解凍ユニットは、現在のメモリページ内に十分な空き空間が存在するか否かを検査する必要がある。圧縮/解凍ユニットは、zデータを圧縮し、それをzバッファメモリ内に書込み始める時には、タイルに関する圧縮zデータのサイズを把握していない。従って、圧縮/解凍ユニットが、書込みの開始時点に空き空間を検査する時には、タイル内の深度データの最大サイズを仮定する必要がある。深度データのサイズは、常に最大サイズとは限らないので、メモリページの最後には浪費メモリ空間が存在する。

【0100】

本発明のこの実施形態は、ページの最後のリンクページアドレスに対して特別に定められたトークンを与えることによってこの問題を解決することを試み、これを図11に示す。図11に示しているように、DPMによって圧縮zデータに割り当てられた各メモリページ1101において、最後の4バイトワード1103が、ページアドレスリンクに向けて保証される。圧縮/解凍ユニットが、書込みがメモリページを超えるのを検出した場合には、このユニットは、DPMから新しいページを要求することになる。現在のページ内の最後のワードは、図11にあるリンクページヘッダ0x80000000-1103と共に新しいページのためのページアドレスリンクで埋められることになる。リンクページヘッダは、ビット31が1であるか、又は4バイトが右から左に80,00,00,00である32ビット16進値である。この後、圧縮/解凍ユニットは、新しいページに向けてページアドレスリンクによって示された新しいページ内に圧縮深度データを書込み続ける。ページの最後では、いかなるメモリ空間も浪費されない。圧縮深度データを単一ページ内に収容することができる場合には、いかなるページアドレスリンクも必要とされない(最終の4バイトを除き)。この場合、現在のページ内の最後のワードは、0000...で埋められることになる。

読取段階中に、圧縮/解凍ユニットは、メモリから読み取られたデータを解凍する。ページ内の最後のワードは、深度データ自体ではなく、常に、新しいページアドレスへのトークンであるページ内の最後のワードのヘッダ(0x80000000)を解析することにより、圧縮/解凍ユニットは、最後のワードが新しいページにリンクするためのアドレスであるのか、又は空白ワード(0000...)であるのかを判断する。最後のワードからのアドレスは、リンクページヘッダが設定されている場合に、リンクされたページから圧縮データをフェッチし続けるのに用いられる。

【0101】

この実施形態の利点は、新しいページの追加アドレスをタイルの領域ヘッダに含める必要がない点である。ページリンクは、圧縮及び解凍処理内で自然に発生する。この実施は、単純で分り易いものである。

本発明者は、この実施形態を適用した後に、圧縮zバッファによって用いられるメモリ空間の平均で35%の低減を達成することができることを見出した。

【符号の説明】

【0102】

501, 503, 505, 507, 509 第1のプリミティブオブジェクトの頂点
 511, 513, 515, 517, 519 第2のプリミティブオブジェクトの頂点
 C1, C2, C3 タイル

10

20

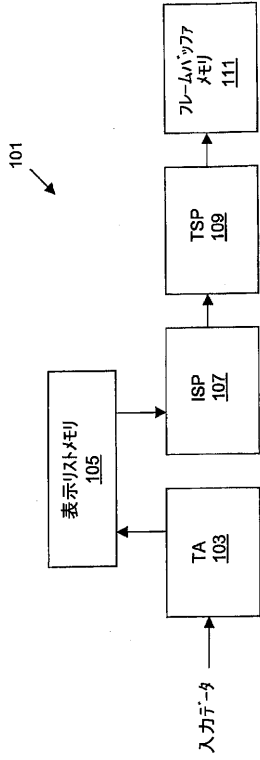
30

40

50

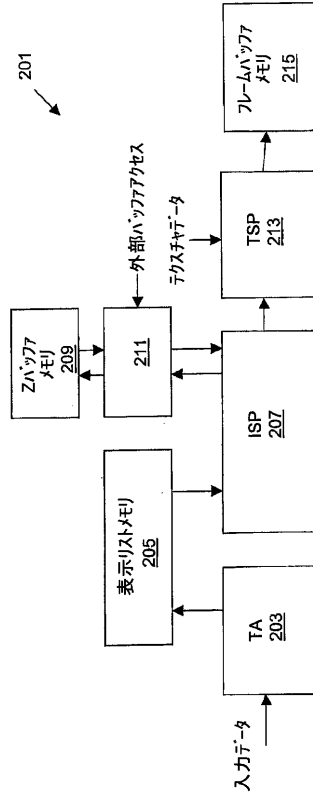
【 図 1 】

Figure 1



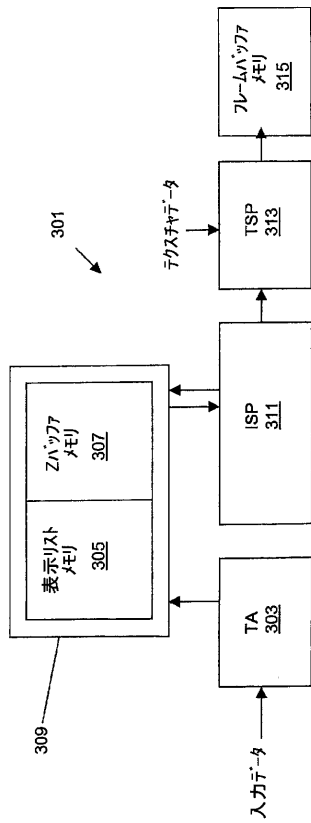
【 図 2 】

Figure 2



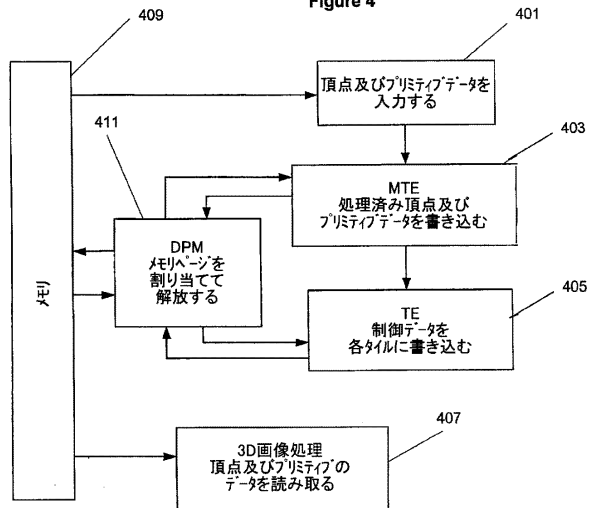
【 図 3 】

Figure 3



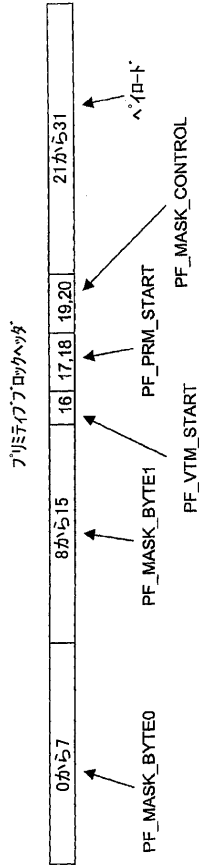
【 図 4 】

Figure 4



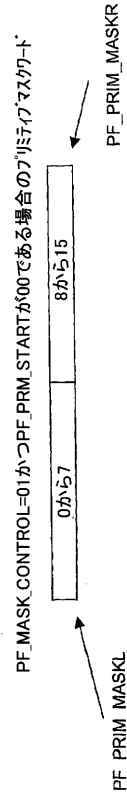
【 8 a 】

Figure 8a



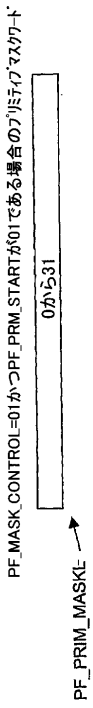
【 8 b 】

Figure 8b



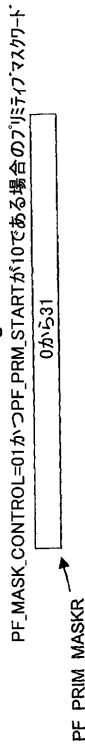
【 8 c 】

Figure 8c



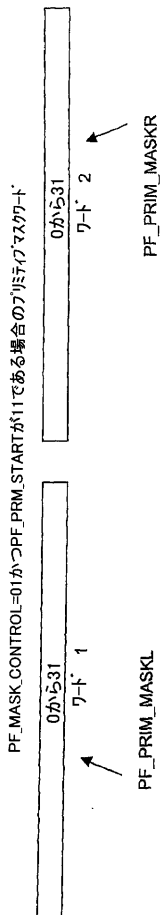
【 8 d 】

Figure 8d



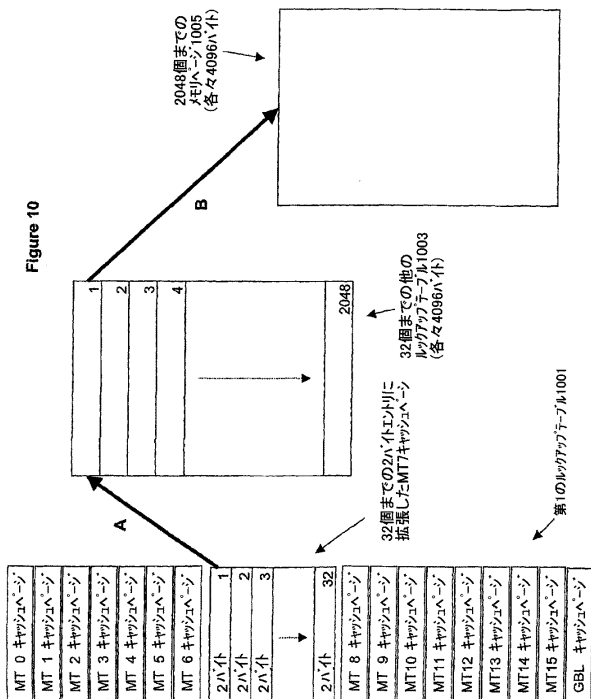
【 図 8 e 】

Figure 8e



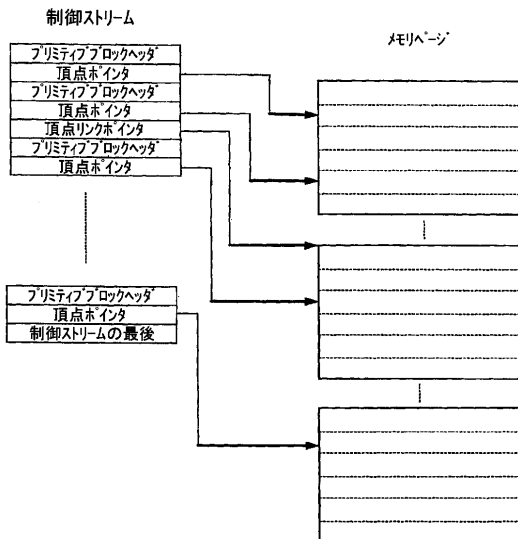
【 図 1 0 】

Figure 10



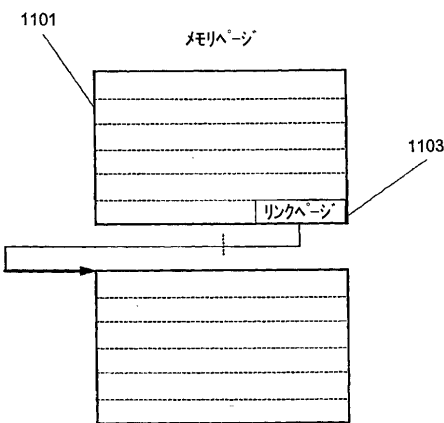
【 図 9 】

Figure 9



【 図 1 1 】

Figure 11



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 レッドショー ジョナサン

イギリス ハートフォードシャー エイエル1 5エスワイ セント アルバンス ロドニー ア
ベニュー 3

(72)発明者 ヤン ザイル

イギリス ハートフォードシャー ダブリュディー3 8アールキュー リックマンズワース ツ
エルソン ウェイ 26

審査官 千葉 久博

(56)参考文献 特開平08-096150(JP,A)

特表2004-537129(JP,A)

特表2003-536153(JP,A)

特表2002-529871(JP,A)

特表2002-529868(JP,A)

米国特許第07102646(US,B1)

英国特許出願公開第02442266(GB,A)

(58)調査した分野(Int.Cl., DB名)

G06T 15/00 - 15/87

A63F 9/24, 13/00 - 13/12

G06T 19/00, 19/20

G09G 5/00 - 5/42