(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第3687945号
(P3687945)
(45)発行日　平成17年8月24日(2005.8.24)　(24)登録日　平成17年6月17日(2005.6.17)

(51)Int.Cl.[7]	F I		
G06T 11/20	G06T	11/20	100
G09G 3/20	G09G	3/20	660X
G09G 5/36	G09G	5/36	510V

請求項の数 9　(全 22 頁)

(21)出願番号　特願平10-353153
(22)出願日　平成10年12月11日(1998.12.11)
(65)公開番号　特開2000-182069(P2000-182069A)
(43)公開日　平成12年6月30日(2000.6.30)
審査請求日　平成14年10月18日(2002.10.18)

(73)特許権者　000002185
ソニー株式会社
東京都品川区北品川６丁目７番３５号
(73)特許権者　000003078
株式会社東芝
東京都港区芝浦一丁目１番１号
(74)代理人　100094053
弁理士　佐藤　隆久
(72)発明者　飯田　亮平
東京都品川区東五反田１丁目１４番１０号　株式会社ソニー木原研究所内
(72)発明者　竹本　卓
神奈川県川崎市幸区堀川町５８０番１号　株式会社東芝半導体システム技術センター内

最終頁に続く

(54)【発明の名称】画像処理装置およびその方法

(57)【特許請求の範囲】
【請求項１】
画像データに対してαブレンディング処理およびディザ処理を行う画像処理装置であって、
少なくとも表示用画像データが描画される記憶回路と、
これから描画すべき現画像データから、既に上記記憶回路に記憶されている画像データを減算した値に与えられた混合係数αを乗じたデータを求めるとともに、上記記憶回路に記憶されている画像データに雑音データを加えたデータを求め、得られた両データを加算することにより、２つの色の線形補間をしたデータに雑音データを加えたデータを求め、このデータから色の有効値を抽出し、この抽出データからデータを間引いて、上記記憶回路に書き戻すロジック回路と
を有する画像処理装置。
【請求項２】
上記ロジック回路は、
これから描画すべき現画像データＳから既に上記記憶回路に記憶されている画像データＤを減算する減算器と、
既に上記記憶回路に記憶されている画像データＤに雑音データである誤差データＥを加算する第１の加算器と、
上記減算器の出力データ（Ｓ－Ｄ）に混合係数αを乗算する乗算器と、
上記乗算器の出力データ｛α×（Ｓ－Ｄ）｝と上記第１の加算器の出力データ（Ｄ＋Ｅ）

を加算する第２の加算器と、
上記第２の加算器の出力データから色の有効値を抽出するクランプ回路と、
上記クランプ回路の出力データから所定のデータを間引いて上記記憶回路に書き戻す切り捨て回路と
を有する請求項１記載の画像処理装置。
【請求項３】
上記記憶回路は、表示用画像データに加えて、少なくとも一つの図形要素が必要とするテクスチャデータを記憶し、
上記ロジック回路は、上記記憶回路の記憶データに基づいて、表示データの図形要素の表面へのテクスチャデータの張り付け処理を行い、
上記記憶回路および上記ロジック回路が一つの半導体チップ内に混載されている
請求項１記載の画像処理装置。
【請求項４】
上記記憶回路は、同一機能を有する複数のモジュールに分割され、
上記ロジック回路は、各モジュールを並列にアクセスする
請求項３記載の画像処理装置。
【請求項５】
上記記憶回路には、表示アドレス空間において、隣接するアドレスにおける表示要素が、異なる記憶ブロックになるように配置される
請求項４記載の画像処理装置。
【請求項６】
前記単位図形の頂点について、３次元座標（ｘ，ｙ，ｚ）、Ｒ（赤），Ｇ（緑），Ｂ（青）データ、混合係数α、テクスチャの同次座標（ｓ，ｔ）および同次項ｑを含むポリゴンレンダリングデータを受けてレンダリング処理を行う画像処理装置であって、
表示用画像データと少なくとも一つの図形要素が必要とするテクスチャデータを記憶する記憶回路と、
これから描画すべき現画像データから、既に上記記憶回路に記憶されている画像データを減算した値に与えられた混合係数αを乗じたデータを用いて求めるとともに、上記記憶回路に記憶されている画像データに雑音データを加えたデータを求め、得られた両データを加算することにより、２つの色の線形補間をしたデータに雑音データを加えたデータを求め、このデータから色の有効値を抽出し、この抽出データからデータを間引いて、上記記憶回路に書き戻す描画データ制御回路と、
前記単位図形の頂点のポリゴンレンダリングデータを補間して、前記単位図形内に位置する画素の補間データを生成する補間データ生成回路と、
前記補間データに含まれるテクスチャの同次座標（ｓ，ｔ）を同次項ｑで除算して「ｓ／ｑ」および「ｔ／ｑ」を生成し、前記「ｓ／ｑ」および「ｔ／ｑ」に応じたテクスチャアドレスを用いて、前記記憶回路からテクスチャデータを読み出し、表示用画像データの図形要素の表面へのテクスチャデータの張り付け処理を行うテクスチャ処理回路と
を少なくとも有し、
前記記憶回路、描画データ制御回路、補間データ生成回路およびテクスチャ処理回路が一つの半導体チップ内に混載されている
画像処理装置。
【請求項７】
上記記憶回路は、同一機能を有する複数のモジュールに分割され、
上記各モジュールは並列にアクセスされる
請求項６記載の画像処理装置。
【請求項８】
上記記憶回路には、表示アドレス空間において、隣接するアドレスにおける表示要素が、異なる記憶ブロックになるように配置される
請求項７記載の画像処理装置。

【請求項９】
画像データに対してαブレンディング処理およびディザ処理を行い記憶回路に描画する画像処理方法であって、
これから描画すべき現画像データから、既に上記記憶回路に記憶されている画像データを減算した値に与えられた混合係数αを乗じる処理と、上記記憶回路に記憶されている画像データに雑音データを加える処理とを並行して行い、
両処理で得られたデータを加算することにより、２つの色の線形補間をしたデータに雑音データを加えたデータを求め、
このデータから色の有効値を抽出し、この抽出データからデータを間引いて、上記記憶回路に書き戻す
画像処理方法。
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】
本発明は、いわゆるαブレンディング処理およびディザ処理を行う画像処理装置およびその方法に関するものである。
【０００２】
【従来の技術】
種々のＣＡＤ(Computer Aided Design) システムや、アミューズメント装置などにおいて、コンピュータグラフィックスがしばしば用いられている。特に、近年の画像処理技術の進展に伴い、３次元コンピュータグラフィックスを用いたシステムが急速に普及している。
このような３次元コンピュータグラフィックスでは、各画素（ピクセル）に対応する色を決定するときに、各画素の色の値を計算し、この計算した色の値を、当該画素に対応するディスプレイバッファ（フレームバッファ）のアドレスに書き込むレンダリング(Rendering) 処理を行う。
【０００３】
レンダリング処理の手法の一つに、ポリゴン（Polygon)レンダリングがある。この手法では、立体モデルを三角形の単位図形（ポリゴン）の組み合わせとして表現しておき、このポリゴンを単位として描画を行うことで、表示画面の色を決定する。
【０００４】
ポリゴンレンダリングでは、物理座標系における三角形の各頂点についての、座標（ｘ，ｙ，ｚ）と、色データ（Ｒ，Ｇ，Ｂ，α）と、張り合わせのイメージパターンを示すテクスチャデータの同次座標（ｓ，ｔ）および同次項ｑの値とを入力とし、これらの値を三角形の内部で補間する処理が行われる。
ここで、同次項ｑは、簡単にいうと、拡大縮小率のようなもので、実際のテクスチャバッファのＵＶ座標系における座標、すなわち、テクスチャ座標データ（ｕ，ｖ）は、同次座標（ｓ，ｔ）を同次項ｑで除算した「ｓ／ｑ」および「ｔ／ｑ」に、それぞれテクスチャサイズＵＳＩＺＥおよびＶＳＩＺＥを乗じたものとなる。
【０００５】
図１０は、３次元コンピュータグラフィックスシステムの基本的な概念を示すシステム構成図である。
【０００６】
この３次元コンピュータグラフィックスシステムにおいては、グラフィックス描画等のデータは、メインプロセッサ１のメインメモリ２、あるいは外部からのグラフィックスデータを受けるＩ／Ｏインタフェース回路３からメインバス４を介してレンダリングプロセッサ５ａ、フレームバッファメモリ５ｂを有するレンダリング回路５に与えられる。
【０００７】
レンダリングプロセッサ５ａには、表示するためのデータを保持することを目的とするフレームバッファ５ｂと、描画する図形要素（たとえば三角形）の表面に張り付けるテクス

チャデータを保持しているテクスチャメモリ６が結合されている。
そして、レンダリングプロセッサ５ａによって、図形要素毎に表面にテクスチャを張り付けた図形要素を、フレームバッファ５ｂに描画するという処理が行われる。
【０００８】
フレームバッファ５ｂとテクスチャメモリ６は、一般的にＤＲＡＭ(Dynamic Random Access Memory)により構成される。
そして、図９のシステムにおいては、フレームバッファ５ｂとテクスチャメモリ６は、物理的に別々のメモリシステムとして構成されている。
【０００９】
ところで、画像データを描画するにあたっては、レンダリングプロセッサ５ａにおいて、必要に応じて、現画像データに含まれる（Ｒ，Ｇ，Ｂ）データと、既にフレームバッファ５ｂに記憶されている（Ｒ，Ｇ，Ｂ）データとが、現画像データに対応するαデータが示す混合値で混合されるαブレンディング処理が行われ、さらにαブレンディング後の画像データをフレームバッファ５ｂの容量等を考慮してデータを間引くディザ(dither)処理が行われて、ディザ処理後の（Ｒ，Ｇ，Ｂ）データがフレームバッファ５に書き戻される。
【００１０】
換言すれば、αブレンディング処理は、２つの色を線形補間して間に色を付ける処理であり、ディザ処理は、αブレンディング処理を受けたデータに雑音データを加え、その後にデータを間引いて、少ない色数で多くの色に見えるようにするための処理である。
【００１１】
図１１は、従来のαブレンディング処理回路およびディザ処理回路の構成例を示すブロック図である。
【００１２】
αブレンディング処理回路６は、現画像データ（たとえば〔０，２５５〕を表現する８ビット整数）Ｓと混合係数α（たとえば〔０，２〕を表現する８ビット整数を乗算する乗算器６１と、１から混合係数αを減算する減算器６２、既にフレームバッファ５ｂに記憶されている画像データ（たとえば〔０，２５５〕を表現する８ビット整数）Ｄと減算器６２の出力とを乗算する乗算器６３と、乗算器６１の出力と乗算器６３の出力とを加算する加算器６４と、加算器６４で得られたデータから色（カラー）値の有効値（たとえば〔０，２５５〕）を抽出するクランプ回路６５とから構成されている。
【００１３】
αブレンディング処理回路６においては、図１１に示すように、加算器６４の出力のように、入力値Ｓ，Ｄ，αからα×Ｓ＋（１－α）×Ｄなるデータが得られる。
【００１４】
また、ディザ処理回路７は、αブレンディング処理回路６の出力信号Ｓ６に雑音データである誤差データ（たとえば〔－４，３〕を表現する３ビット整数）Ｅを加算する加算器７１と、加算器７１の出力からカラー値の有効値を抽出するクランプ回路７２と、クランプ回路７２の出力から下位３ビットを切り捨てて（間引いて）上位５ビットをフレームバッファ５ｂに書き戻す切り捨て回路（除算回路）７３とから構成されている。
【００１５】
ディザ処理回路７においては、図１１に示すように、加算器７１の出力のように、入力値α×Ｓ＋（１－α）×ＤおよびＥからα（Ｓ－Ｄ）＋Ｄ＋Ｅなるデータが得られる。
【００１６】
【発明が解決しようとする課題】
しかしながら、上述したように、従来の画像処理装置では、αブレンディング処理回路６およびディザ処理回路７が別々に設けられ、かつ直列に接続された構成であることから、回路規模が大きくなり、また、演算時間がかかり、高速処理の障害となっている。
【００１７】
また、上述した３次元コンピュータグラフィックスシステムにおける従来のいわゆる内蔵ＤＲＡＭシステムにおいて、フレームバッファメモリとテクスチャメモリが別々のメモリ

システムに別れている場合においては、以下のような不利益があった。

【００１８】

第１に、表示の解像度の変化によって空きとなったフレームバッファをテクスチャ用に利用できない、あるいはフレームバッファメモリとテクスチャメモリを物理的に同一にすると、フレームバッファメモリとテクスチャーメモリの同時アクセスにおいて、ＤＲＡＭのページ切り替え等のオーバーヘッドが大きくなり、性能を犠牲にしなければならなくなる等の不利益がある。

【００１９】

本発明は、かかる事情に鑑みてなされたものであり、その目的は、αブレンディングおよびディザ処理用回路の回路規模を小さくでき、しかも高速処理を実現でき、また、表示の解像度の変化によって空きとなったメモリ領域をテクスチャ用に利用でき、ページ切り替え等のオーバーヘッドの増大を防止き、性能を低下を招くことがない、柔軟でかつ高速処理が可能な画像処理装置およびその方法を提供することにある。

【００２０】

【課題を解決するための手段】

上記目的を達成するため、本発明は、画像データに対してαブレンディング処理およびディザ処理を行う画像処理装置であって、少なくとも表示用画像データが描画される記憶回路と、これから描画すべき現画像データから、既に上記記憶回路に記憶されている画像データを減算した値に与えられた混合係数αを乗じたデータを用いて求めるとともに、上記記憶回路に記憶されている画像データに雑音データを加えたデータを求め、得られた両データを加算することにより、２つの色の線形補間をしたデータに雑音データを加えたデータを求め、このデータから色の有効値を抽出し、この抽出データからデータを間引いて、上記記憶回路に書き戻すロジック回路とを有する。

【００２１】

本発明では、上記ロジック回路は、これから描画すべき現画像データＳから既に上記記憶回路に記憶されている画像データＤを減算する減算器と、既に上記記憶回路に記憶されている画像データＤに雑音データである誤差データＥを加算する第１の加算器と、上記減算器の出力データ（Ｓ－Ｄ）に混合係数αを乗算する乗算器と、上記乗算器の出力データ｛α×（Ｓ－Ｄ）｝と上記第１の加算器の出力データ（Ｄ＋Ｅ）を加算する第２の加算器と、上記第２の加算器の出力データから色の有効値を抽出するクランプ回路と、上記クランプ回路の出力データから所定のデータを間引いて上記記憶回路に書き戻す切り捨て回路とを有する。

【００２２】

また、本発明では、上記記憶回路は、表示用画像データに加えて、少なくとも一つの図形要素が必要とするテクスチャデータを記憶し、上記ロジック回路は、上記記憶回路の記憶データに基づいて、表示データの図形要素の表面へのテクスチャデータの張り付け処理を行い、上記記憶回路および上記ロジック回路が一つの半導体チップ内に混載されている。

【００２３】

また、本発明では、上記記憶回路は、同一機能を有する複数のモジュールに分割され、上記ロジック回路は、各モジュールを並列にアクセスする。

【００２４】

また、本発明では、上記記憶回路には、表示アドレス空間において、隣接するアドレスにおける表示要素が、異なる記憶ブロックになるように配置される。

【００２５】

また、本発明は、前記単位図形の頂点について、３次元座標（ｘ，ｙ，ｚ）、Ｒ（赤），Ｇ（緑），Ｂ（青）データ、混合係数α、テクスチャの同次座標（ｓ，ｔ）および同次項ｑを含むポリゴンレンダリングデータを受けてレンダリング処理を行う画像処理装置であって、表示用画像データと少なくとも一つの図形要素が必要とするテクスチャデータを記憶する記憶回路と、これから描画すべき現画像データから、既に上記記憶回路に記憶されている画像データを減算した値に与えられた混合係数αを乗じたデータを求めるとともに

、上記記憶回路に記憶されている画像データに雑音データを加えたデータを求め、得られた両データを加算することにより、２つの色の線形補間をしたデータに雑音データを加えたデータを求め、このデータから色の有効値を抽出し、この抽出データからデータを間引いて、上記記憶回路に書き戻す描画データ制御回路と、前記単位図形の頂点のポリゴンレンダリングデータを補間して、前記単位図形内に位置する画素の補間データを生成する補間データ生成回路と、前記補間データに含まれるテクスチャの同次座標（ｓ，ｔ）を同次項ｑで除算して「ｓ／ｑ」および「ｔ／ｑ」を生成し、前記「ｓ／ｑ」および「ｔ／ｑ」に応じたテクスチャアドレスを用いて、前記記憶回路からテクスチャデータを読み出し、表示用画像データの図形要素の表面へのテクスチャデータの張り付け処理を行うテクスチャ処理回路と、を少なくとも有し、前記記憶回路、描画データ制御回路、補間データ生成回路およびテクスチャ処理回路が一つの半導体チップ内に混載されている。

【００２６】

また、本発明は、画像データに対してαブレンディング処理およびディザ処理を行い記憶回路に描画する画像処理方法であって、これから描画すべき現画像データから、既に上記記憶回路に記憶されている画像データを減算した値に与えられた混合係数αを乗じる処理と、上記記憶回路に記憶されている画像データに雑音データを加える処理とを並行して行い、両処理で得られたデータを加算することにより、２つの色の線形補間をしたデータに雑音データを加えたデータを求め、このデータから色の有効値を抽出し、この抽出データからデータを間引いて、上記記憶回路に書き戻す。

【００２７】

本発明によれば、ロジック回路において、まず、これから描画すべき現画像データの、既に上記記憶回路に記憶されている画像データに対する更新量データが、与えられた混合係数αを用いて求められ、これと並行して、記憶回路に記憶されている画像データに雑音データを加えたデータが求められる。

次に、両処理で得られたデータが加算されて、αブレンディング処理が行われた画像データに雑音データが加えられたデータが得られる。

そして、この加算データから色の有効値が抽出され、この抽出データからデータが切り捨てる等の処理で間引かれていて、記憶回路に書き戻される。

すなわち、αブレンディング処理およびディザ処理が、簡単化された回路で、短時間に行われる。

【００２８】

また、本発明によれば、一つの半導体チップの内部に、ＤＲＡＭ等の記憶回路とロジック回路を混載させ、表示用画像データと少なくとも一つの図形要素が必要とするテクスチャデータを、内蔵の記憶回路に記憶させていることにより、表示領域以外の部分にテクスチャデータを格納できることになり、内蔵メモリの有効利用が可能となる。

【００２９】

また、記憶回路における同一機能を独立した複数のモジュールとして並列にもつことにより、並列動作の効率が向上する。単にデータのビット数が多いだけでは、データの使用効率は悪化し、性能向上できるのは一部の条件の場合に限定されることになるが、平均的な性能を向上させるためには、ある程度の機能をもったモジュールを、複数設けることで、ビット線が有効に利用できる。

【００３０】

また、内蔵記憶回路の配置、すなわち、それぞれの独立されたメモリ＋機能モジュールが、占有するアドレス空間を工夫することで、さらにビット線の有効利用が可能となる。

グラフィックス描画におけるような、比較的固まった表示領域へのアクセスが多い場合には、表示アドレス空間において、隣接するアドレスにおける表示要素が、それぞれ異なるメモリのブロックになるように配置することで、それぞれのモジュールが同時に処理できる確率が増加し、描画性能の向上が可能となる。固まった表示領域へのアクセスが多いというのは、三角形等の閉領域の内部を描画しようとした場合、その内部の表示要素どうしは隣接しているので、そのような領域へのアクセスはアドレス隣接することになる。

【００３１】
【発明の実施の形態】
以下、本実施形態においては、パーソナルコンピュータなどに適用される、任意の３次元物体モデルに対する所望の３次元画像をＣＲＴ(Cathode Ray Tube)などのディスプレイ上に高速に表示する３次元コンピュータグラフィックスシステムについて説明する。
【００３２】
図１は、本発明に係る画像処理装置としての３次元コンピュータグラフィックスシステム１０のシステム構成図である。
【００３３】
３次元コンピュータグラフィックスシステム１０は、立体モデルを単位図形である三角形（ポリゴン）の組み合わせとして表現し、このポリゴンを描画することで表示画面の各画素の色を決定し、ディスプレイに表示するポリゴンレンダリング処理を行うシステムである。
また、３次元コンピュータグラフィックスシステム１０では、平面上の位置を表現する（ｘ，ｙ）座標の他に、奥行きを表すｚ座標を用いて３次元物体を表し、この（ｘ，ｙ，ｚ）の３つの座標で３次元空間の任意の一点を特定する。
【００３４】
図１に示すように、３次元コンピュータグラフィックスシステム１０は、メインプロセッサ１１、メインメモリ１２、Ｉ／Ｏインタフェース回路１３、およびレンダリング回路１４が、メインバス１５を介して接続されている。
以下、各構成要素の機能について説明する。
【００３５】
メインプロセッサ１１は、たとえば、アプリケーションの進行状況などに応じて、メインメモリ１２から必要なグラフィックデータを読み出し、このグラフィックデータに対してクリッピング(Clipping)処理、ライティング(Lighting)処理などのジオメトリ(Geometry)処理などを行い、ポリゴンレンダリングデータを生成する。メインプロセッサ１１は、ポリゴンレンダリングデータＳ１１を、メインバス１５を介してレンダリング回路１４に出力する。
【００３６】
Ｉ／Ｏインタフェース回路１３は、必要に応じて、外部から動きの制御情報またはポリゴンレンダリングデータ等を入力し、これをメインバス１５を介してレンダリング回路１４に出力する。
【００３７】
ここで、ポリゴンレンダリングデータは、ポリゴンの各３頂点の（ｘ，ｙ，ｚ，Ｒ，Ｇ，Ｂ，α，ｓ，ｔ，ｑ，Ｆ）のデータを含んでいる。
ここで、（ｘ，ｙ，ｚ）データは、ポリンゴの頂点の３次元座標を示し、（Ｒ，Ｇ，Ｂ）データは、それぞれ当該３次元座標における赤、緑、青の輝度値を示している。
データαは、これから描画する画素と、ディスプレイバッファ１４７ｂに既に記憶されている画素とのＲ，Ｇ，Ｂデータのブレンド（混合）係数を示している。
（ｓ，ｔ，ｑ）データのうち、（ｓ，ｔ）は、対応するテクスチャの同次座標を示しており、ｑは同次項を示している。ここで、「ｓ／ｑ」および「ｔ／ｑ」に、それぞれテクスチャサイズＵＳＩＺＥおよびＶＳＩＺＥを乗じてテクスチャ座標データ（ｕ，ｖ）が得られる。テクスチャバッファ１４７ａに記憶されたテクスチャデータへのアクセスは、テクスチャ座標データ（ｕ，ｖ）を用いて行われる。
Ｆデータは、フォグのα値を示している。
すなわち、ポリゴンレンダリングデータは、三角形の各頂点の物理座標値と、それぞれの頂点の色とテクスチャおよびフォグの値のデータである。
【００３８】
以下、レンダリング回路１４について詳細に説明する。
図１に示すように、レンダリング回路１４は、ＤＤＡ(Digital Differential Anarizer)

セットアップ回路１４１、トライアングルＤＤＡ回路１４２、テクスチャエンジン回路１４３、メモリインタフェース（Ｉ／Ｆ）回路１４４、ＣＲＴコントロール回路１４５、ＲＡＭＤＡＣ回路１４６、ＤＲＡＭ１４７およびＳＲＡＭ(Static RAM)１４８を有する。
本実施形態におけるレンダリング回路１４は、一つの半導体チップ内にロジック回路と少なくとも表示用画像データ（以下、表示データという）とテクスチャデータとを記憶するＤＲＡＭ１４７とが混載されている。

【００３９】

ＤＲＡＭ１４７

ＤＲＡＭ１４７は、テクスチャバッファ１４７ａ、ディスプレイバッファ１４７ｂ、ｚバッファ１４７ｃおよびテクスチャＣＬＵＴ(Color Look Up Table) バッファ１４７ｄとして機能する。
また、ＤＲＡＭ１４７は、後述するように、同一機能を有する複数（本実施形態では４個）のモジュールに分割されている。

【００４０】

また、ＤＲＡＭ１４７には、より多くのテクスチャデータを格納するために、インデックスカラーにおけるインデックスと、そのためのカラールックアップテーブル値が、テクスチャＣＬＵＴバッファ１４７ｄに格納されている。
インデックスおよびカラールックアップテーブル値は、テクスチャ処理に使われる。すなわち、通常はＲ，Ｇ，Ｂそれぞれ８ビットの合計２４ビットでテクスチャ要素を表現するが、それではデータ量が膨らむため、あらかじめ選んでおいたたとえば２５６色等の中から一つの色を選んで、そのデータをテクスチャ処理に使う。このことで２５６色であればそれぞれのテクスチャ要素は８ビットで表現できることになる。インデックスから実際のカラーへの変換テーブルは必要になるが、テクスチャの解像度が高くなるほど、よりコンパクトなテクスチャデータとすることが可能となる。
これにより、テクスチャデータの圧縮が可能となり、内蔵ＤＲＡＭの効率良い利用が可能となる。

【００４１】

さらにＤＲＡＭ１４７には、描画と同時並行的に隠れ面処理を行うため、描画しようとしている物体の奥行き情報が格納されている。
なお、表示データと奥行きデータおよびテクスチャデータの格納方法としては、メモリブロックの先頭から連続して表示データが格納され、次に奥行きデータが格納され、残りの空いた領域に、テクスチャの種類毎に連続したアドレス空間でテクスチャデータが格納される。これにより、テクスチャデータを効率よく格納できることになる。

【００４２】

ＤＤＡセットアップ回路１４１

ＤＤＡセットアップ回路１４１は、後段のトライアングルＤＤＡ回路１４２において物理座標系上の三角形の各頂点の値を線形補間して、三角形の内部の各画素の色と深さ情報を求めるに先立ち、ポリゴンレンダリングデータＳ１１が示す（ｚ，Ｒ，Ｇ，Ｂ，α，ｓ，ｔ，ｑ，Ｆ）データについて、三角形の辺と水平方向の差分などを求めるセットアップ演算を行う。
このセットアップ演算は、具体的には、開始点の値と終点の値と、開始点と終点との距離を用いて、単位長さ移動した場合における、求めようとしている値の変分を算出する。
ＤＤＡセットアップ回路１４１は、算出した変分データＳ１４１をトライアングルＤＤＡ回路１４２に出力する。

【００４３】

ＤＤＡセットアップ回路１４１の機能について図２に関連付けてさらに説明する。
上述したように、ＤＤＡセットアップ回路１４１の主な処理は、前段のジオメトリ処理を経て物理座標にまで落ちてきた各頂点における各種情報（色、ブレンドの割合、テクスチャ座標、Ｆｏｇカラー）の与えられた三頂点により構成される三角形内部で変分を求めて、後段の線形補間処理の基礎データを算出することである。

なお、三角形の各頂点データは、たとえばｘ，ｙ座標が１６ビット、ｚ座標が２４ビット、ＲＧＢカラー値が各１２ビット（＝８＋４）、ｓ，ｔ，ｑテクスチャ座標は各３２ビット浮動少数値（ＩＥＥＥフォーマット）、α係数が１２ビット、並びにＦｏｇ係数が１２ビットで構成される。

【００４４】

三角形の描画は水平ラインの描画に集約されるが、そのために水平ラインの描画開始点における最初の値を求める必要がある。

この水平ラインの描画においては、一つの三角形の中でその描画方向は一定にする。たとえば左から右へ描画する場合は、左側の辺におけるＹ方向変位に対するＸおよび上記各種の変分を算出しておいて、それを用いて頂点から次の水平ラインに移った場合の最も左の点のｘ座標と、上記各種情報の値を求める（辺上の点はＹ，Ｘ両方向に変化するのでＹ方向の傾きのみでは計算できない。）。

右側の辺に関しては終点の位置がわかればよいので、Ｙ方向変位に対するｘの変分のみを調べておけばよい。

水平ラインの描画に関しては、水平方向の傾きは同一三角形内では均一なので、上記各種情報の傾きを算出しておく。

与えられた三角形をＹ方向にソートして最上位の点をＡとする。次に残りの２頂点のＸ方向の位置を比較して右側の点をＢとする。こうすることで、処理の場合分け等が２通り程度にできる。

【００４５】

トライアングルＤＤＡ回路１４２

トライアングルＤＤＡ回路１４２は、ＤＤＡセットアップ回路１４１から入力した変分データＳ１４１を用いて、三角形内部の各画素における線形補間された（ｚ，Ｒ，Ｇ，Ｂ，α，ｓ，ｔ，ｑ，Ｆ）データを算出する。

トライアングルＤＤＡ回路１１は、各画素の（ｘ，ｙ）データと、当該（ｘ，ｙ）座標における（ｚ，Ｒ，Ｇ，Ｂ，α，ｓ，ｔ，ｑ，Ｆ）データとを、ＤＤＡデータ（補間データ）Ｓ１４２としてテクスチャエンジン回路１４３に出力する。

たとえば、トライアングルＤＤＡ回路１４２は、並行して処理を行う矩形内に位置する８（＝２×４）画素分のＤＤＡデータＳ１４２をテクスチャエンジン回路１４３に出力する。

【００４６】

トライアングルＤＤＡ回路１４２の機能について図３に関連付けてさらに説明する。

上述したように、ＤＤＡセットアップ回路１４１により、三角形の各辺と水平方向における先出の各種情報の傾き情報が準備され、この情報を受けたトライアングルＤＤＡ回路１４２の基本的処理は、三角形の辺上の各種情報の補間処理による水平ラインの初期値の算出と、水平ライン上での各種情報の補間処理である。

ここで最も注意しなければならないことは、補間結果の算出は、画素中心における値を算出する必要があるということである。

その理由は、算出する値が画素中心からはずれたところを求めていては、静止画の場合はさほど気にならないが、動画にした場合には、画像の揺らぎが目立つようになるからである。

【００４７】

最初の水平ライン（当然画素中心を結んだライン）の一番左側における各種情報は、辺上の傾きに頂点からその最初の水平ラインまでの距離をかけてやることで求めることができる。

次のラインにおける開始位置での各種情報は、辺上の傾きを足してゆくことで算出できる。

水平ラインにおける最初の画素での値は、ラインの開始位置における値に、最初の画素までの距離と水平方向の傾きをかけた値を足すことで算出できる。水平ラインにおける次の画素における値は、最初の画素の値に対してつぎつぎに水平方向の傾きを足し込んでゆけ

ば算出できる。

【００４８】

次に、頂点のソートについて図４に関連付けて説明する。

頂点をあらかじめソートしておくことで、以降の処理の場合分けを最大限に減らし、かつ、補間処理においてもできるだけ一つの三角形の内部においては、矛盾が生じにくくすることができる。

ソートのやり方としては、まずすべての与えられた頂点をＹ方向にソートして、最上位の点と最下位の点を決めそれぞれＡ点、Ｃ点とする。残りの点はＢ点とする。

このようにすることで、Ｙ方向に最も長く伸びた辺が辺ＡＣとなり、最初に辺ＡＣと辺ＡＢを用いてその二つの辺で挟まれた領域の補間処理を行い、次に辺ＡＣはそのままで、辺ＡＢに変えて辺ＢＣと辺ＡＣで挟まれた領域の補間を行うという処理になる。また、Ｙ方向の画素座標格子上への補正に関しても、辺ＡＣと辺ＢＣについて行っておけばよいこともわかる。

このようにして、ソート後の処理に場合分けが不必要になることで、データを単純に流すだけの処理で可能となりバグも発生しにくくなるし、構造もシンプルになる。

また、一つの三角形の中で補間処理の方向が辺ＢＣ上を開始点として一定にできるため、水平方向の補間(Span)の方向が一定となり、演算誤差があったとしても辺ＢＣから他の辺に向かって誤差が蓄積されるかたちとなり、その蓄積の方向が一定となるため、隣接する辺同士での誤差は目立たなくなる。

【００４９】

次に、水平方向の傾き算出について図５に関連付けて説明する。

三角形内における各種変数（ｘ，ｚ，α，Ｒ，Ｇ，Ｂ，ｓ，ｔ，ｑ）の（ｘ，ｙ）に対する傾き（変数分）は、線形補間であることから一定となる。

したがって、水平方向の傾き、すなわち、各水平ライン(Span)上での傾きはどのSpanにおいても、一定となるので、各Spanの処理に先立ってその傾きを求めておくことになる。

三角形の与えられた頂点をＹ方向にソートした結果、辺ＡＣが最も長く伸びた辺と再定義されているので、頂点Ｂを水平方向に伸ばしたラインと辺ＡＣの交点が必ず存在するのでその点をＤとする。

後は単純に点Ｂと点Ｄの間の変分を求めるようなことを行えば、水平方向すなわちｘ方向の傾きを求めることができる。

【００５０】

具体的には、Ｄ点でのｘおよびｚ座標は次式のようになる。

【００５１】

【数１】

$$x_d = \{(y_d - y_a) / (y_c - y_a)\} \cdot (x_c - x_a)$$
$$z_d = \{(y_d - y_a) / (y_c - y_a)\} \cdot (z_c - z_a)$$

【００５２】

これに基づいて、変数ｚのｘ方向の傾きを求めると、次のようになる。

【００５３】

【数２】

$$\begin{aligned}\Delta z / \Delta x &= (z_d - z_b) / (x_d - x_b) \\ &= [\{(y_d - y_a) / (y_c - y_a)\} \cdot (z_c - z_a) - z_b] \\ &\quad / [\{(y_d - y_a) / (y_c - y_a)\} \cdot (x_c - x_a) - x_b] \\ &= \{z_b (y_c - y_a) - (z_c - z_a)(y_c - y_a)\} \\ &\quad / \{x_b (y_c - y_a) - (z_c - z_a)(y_c - y_a)\}\end{aligned}$$

【００５４】

次に、頂点データの補間手順の一例について、図６および図７に関連付けて説明する。

頂点のソート、水平方向の傾き算出、各辺上での傾きの算出処理を経て、それらの結果を使って補間処理を行う。
Ｂ点の位置によって、Spanでの処理の向きは２通りに別れる。これは、一つの三角形の内部での補間における各Span同士での誤差の蓄積方向を、一定にすることで、できるだけ不具合が発生しないようにするために、Ｙ方向に最も長く伸びた辺を常に始点として、処理するようにしようとしているからである。
Ｂ点がＡ点と同じ高さにあった場合には、前半の処理はスキップされることになる。よって、場合分けというよりは、スキップが可能な機構を設けておくだけで処理としてはすっきりしたものとできる。
複数のSpanを同時処理することで、処理能力をあげようとした場合には、Ｙ方向における傾きを求めたくなるが、頂点のソートからやり直す必要があることになる。しかしながら、補間処理の前処理だけでことが済むために、全体としての処理系は簡単にできる。
【００５５】
具体的には、Ｂ点がＡ点と同じ高さでない場合には、ＡＣ，ＡＢのＹ方向補正（画素格子上の値算出）を行い（ＳＴ１，ＳＴ２）、ＡＣ辺上の補間およびＡＢ辺上の補間を行う（ＳＴ３）。
そして、ＡＣ水平方向の補正およびＡＣ辺からＡＢ辺方向の水平ライン(Span)上を補間する（ＳＴ４）。
以上のステップＳＴ３，ＳＴ４の処理をＡＢ辺の端点まで行う（ＳＴ５）。
ＡＢ辺の端点までステップＳＴ２～ＳＴ４の処理が終了した場合、あるいはステップＳＴ１においてＢ点がＡ点が同じ高さであると判別した場合には、ＢＣのＹ方向補正（画素格子上の値算出）を行い（ＳＴ６）、ＡＣ辺上の補間およびＢＣ辺上の補間を行う（ＳＴ７）。
そして、ＡＣ水平方向の補正およびＡＣ辺からＢＣ辺方向の水平ライン(Span)上を補間する（ＳＴ８）。
以上のステップＳＴ７，ＳＴ８の処理をＢＣ辺の端点まで行う（ＳＴ９）。
【００５６】
テクスチャエンジン回路１４３
テクスチャエンジン回路１４３は、「ｓ／ｑ」および「ｔ／ｑ」の算出処理、テクスチャ座標データ（ｕ，ｖ）の算出処理、テクスチャバッファ１４７ａからの（Ｒ，Ｇ，Ｂ，α）データの読み出し処理、および、混合処理（αブレンディング処理）を順にパイプライン方式で行う。
なお、テクスチャエンジン回路１４３は、たとえば所定の矩形内に位置する８画素についての処理を同時に並行して行う。
【００５７】
テクスチャエンジン回路１４３は、ＤＤＡデータＳ１４２が示す（ｓ，ｔ，ｑ）データについて、ｓデータをｑデータで除算する演算と、ｔデータをｑデータで除算する演算とを行う。
テクスチャエンジン回路１４３には、たとえば図示しない除算回路が８個設けられており、８画素についての除算「ｓ／ｑ」および「ｔ／ｑ」が同時に行われる。
【００５８】
また、テクスチャエンジン回路１４３は、除算結果である「ｓ／ｑ」および「ｔ／ｑ」に、それぞれテクスチャサイズＵＳＩＺＥおよびＶＳＩＺＥを乗じて、テクスチャ座標データ（ｕ，ｖ）を生成する。
また、テクスチャエンジン回路１４３は、メモリＩ／Ｆ回路１４４を介して、ＳＲＡＭ１４８あるいはＤＲＡＭ１４７に、生成したテクスチャ座標データ（ｕ，ｖ）を含む読み出し要求を出力し、メモリＩ／Ｆ回路１４４を介して、ＳＲＡＭ１４８あるいはテクスチャバッファ１４７ａに記憶されているテクスチャデータを読み出すことで、（ｓ，ｔ）データに対応したテクスチャアドレスに記憶された（Ｒ，Ｇ，Ｂ，α）データＳ１４８を得る。

ここで、ＳＲＡＭ１４８には、テクスチャバッファ１４７ａに格納されているテクスチャデータが記憶される。
テクスチャエンジン回路１４３は、読み出した（Ｒ，Ｇ，Ｂ，α）データＳ１４８の（Ｒ，Ｇ，Ｂ）データと、前段のトライアングルＤＤＡ回路１４２からのＤＤＡデータＳ１４２に含まれる（Ｒ，Ｇ，Ｂ）データとを、（Ｒ，Ｇ，Ｂ，α）データＳ１４８に含まれるαデータが示す割合で混合し（それぞれかけあわせるなどして）、画素データＳ１４３を生成する。
テクスチャエンジン回路１４３は、この画素データＳ１４３をメモリＩ／Ｆ回路１４４に出力する。

【００５９】

なお、テクスチャバッファ１４７ａには、ＭＩＰＭＡＰ（複数解像度テクスチャ）などの複数の縮小率に対応したテクスチャデータが記憶されている。ここで、何れの縮小率のテクスチャデータを用いるかは、所定のアルゴリズムを用いて、前記三角形単位で決定される。

【００６０】

テクスチャエンジン回路１４３は、フルカラー方式の場合には、テクスチャバッファ１４７ａから読み出した（Ｒ，Ｇ，Ｂ，α）データを直接用いる。
一方、テクスチャエンジン回路１４３は、インデックスカラー方式の場合には、あらかじめ作成したカラールックアップテーブル（ＣＬＵＴ）をテクスチャＣＬＵＴバッファ１４７ｄから読み出して、内蔵するＳＲＡＭに転送および記憶し、このカラールックアップテーブルを用いて、テクスチャバッファ１４７ａから読み出したカラーインデックスに対応する（Ｒ，Ｇ，Ｂ）データを得る。

【００６１】

メモリＩ／Ｆ回路１４４

メモリＩ／Ｆ回路１４４は、テクスチャエンジン回路１４３から入力した画素データＳ１４３に対応するｚデータと、ｚバッファ１４７ｃに記憶されているｚデータとの比較を行い、入力した画素データＳ１４３によって描画される画像が、前回、ディスプレイバッファ１４７ｂに書き込まれた画像より、手前（視点側）に位置するか否かを判断し、手前に位置する場合には、画像データＳ１４３に対応するｚデータでｚバッファ１４７ｃに記憶されたｚデータを更新する。
また、メモリＩ／Ｆ回路１４４は、必要に応じて、画像データＳ１４３に含まれる（Ｒ，Ｇ，Ｂ）データと、既にディスプレイバッファ１４７ｂに記憶されている（Ｒ，Ｇ，Ｂ）データとを、画像データＳ１４３に対応するαデータが示す混合値で混合するαブレンディング処理と、ディスプレイバッファ１４７ｂの容量等を考慮してデータを間引く（切り捨てる）ディザ処理とを並列的に行い、処理後の（Ｒ，Ｇ，Ｂ）データをディスプレイバッファ１４７ｂに書き込む（打ち込む）。

【００６２】

図８は、αブレンディング処理とディザ処理とを並列的に行うαブレンディング／ディザ処理回路１４４０の構成例を示すブロック図である。

【００６３】

このロジック回路または描画データ制御回路としてのαブレンディング／ディザ処理回路１４４０は、図８に示すように、これから描画すべき現画像データ（たとえば〔０，２５５〕を表現する８ビット整数）Ｓから既にディスプレイバッファ１４７ｂに記憶されている画像データ（たとえば〔０，２５５〕を表現する８ビット整数）Ｄを減算する減算器１４４１と、既にディスプレイバッファ１４７ｂに記憶されている画像データＤに雑音データである誤差データ（たとえば〔－４，３〕を表現する３ビット整数）Ｅを加算する第１の加算器１４４２と、減算器１４４１の出力データ（Ｓ－Ｄ）に混合係数（たとえば〔０，２〕を表現する８ビット整数）αを乗算する乗算器１４４３と、乗算器１４４３の出力データ｛α×（Ｓ－Ｄ）｝と第１の加算器１４４２の出力データ（Ｄ＋Ｅ）を加算する第２の加算器１４４４と、第２の加算器１４４４の出力データからカラー値の有効値（たと

えば〔０，２５５〕）を抽出するクランプ回路１４４５と、クランプ回路１４４５の出力データから下位３ビットを切り捨てて（間引いて）上位５ビットをディスプレイバッファ１４７ｂに書き戻す切り捨て回路（除算回路）１４４６とから構成されている。

【００６４】

このαブレンディング／ディザ処理回路１４４０では、減算器１４４１および乗算器１４４３で、これから描画すべき現画像データＳの既にディスプレイバッファ１４７ｂに記憶されている画像データＤに対する更新量データを混合係数αを用いて求める処理と、第１の加算器１４４２で既にディスプレイバッファ１４７ｂに記憶されている画像データＤに雑音データＥを加える処理とが同時並列的に行われ、両処理で得られたデータを第２の加算器１４４４で加算することにより、２つの色の線形補間をしたデータに雑音データを加えたデータ｛α×（Ｓ－Ｄ）＋Ｄ＋Ｅが求められ、その後、クランプ回路１４４５でカラーの有効値が抽出され、この抽出データから切り捨て回路１４４６でデータが間引かれて、ディスプレイバッファ１４７ｂに書き戻される。

【００６５】

この回路は、従来回路に比べて乗算器およびクランプ回路が１つずつ少ない構成となっており、回路規模が小さく、また、αブレンディング処理とディザ処理とが並列的に行われることから、演算時間が短縮されている。

【００６６】

なお、メモリＩ／Ｆ回路１４４によるＤＲＡＭ１４７に対してのアクセスは、１６画素について同時に行われる。

【００６７】

本実施形態においては、ＤＲＡＭ１４７は、たとえば図９に示すように、４つのＤＲＡＭモジュール１４７１～１４７４に分割されており、メモリＩ／Ｆ回路１４４には、各ＤＲＡＭモジュール１４７１～１４７４に対応したメモリコントローラ１４４７～１４５０、並びにこれらメモリコントローラ１４４１にデータを分配するディストリビュータ１４５１が設けられている。

そして、メモリＩ／Ｆ回路１４４は、各ＤＲＡＭモジュール１４７１～１４７４に対して、図９に示すように、画素データを、表示領域において隣接した部分は、異なるＤＲＡＭモジュールとなるように配置する。

これにより、三角形のような平面を描画する場合には面で同時に処理できることになるため、それぞれのＤＲＡＭモジュールの動作確率は非常に高くなっている。

【００６８】

ＣＲＴコントロール回路１４５

ＣＲＴコントローる回路１４５は、与えられた水平および垂直同期信号に同期して、図示しないＣＲＴに表示するアドレスを発生し、ディスプレイバッファ１４７ｂから表示データを読み出す要求をメモリＩ／Ｆ回路１４４に出力する。この要求に応じて、メモリＩ／Ｆ回路１４４は、ディスプレイバッファ１４７ｂから一定の固まりで表示データを読み出す。ＣＲＴコントローラ回路１４５は、ディスプレイバッファ１４７ｂから読み出した表示データを記憶するＦＩＦＯ(First In First Out)回路を内蔵し、一定の時間間隔で、ＲＡＭＤＡＣ回路１４６に、ＲＧＢのインデックス値を出力する。

【００６９】

ＲＡＭＤＡＣ回路１４６

ＲＡＭＤＡＣ回路１４６は、各インデックス値に対応するＲ，Ｇ，Ｂデータを記憶しており、ＣＲＴコントローラ回路１４５から入力したＲＧＢのインデックス値に対応するデジタル形式のＲ，Ｇ，Ｂデータを、図示しないＤ／Ａコンバータ(Digital/Analog Converter)に転送し、アナログ形式のＲ，Ｇ，Ｂデータを生成する。ＲＡＭＤＡＣ回路１４６は、この生成されたＲ，Ｇ，ＢデータをＣＲＴに出力する。

【００７０】

次に、上記構成による動作を説明する。

３次元コンピュータグラフィックスシステム１０においては、グラフィックス描画等のデ

ータは、メインプロセッサ１１のメインメモリ１２、あるいは外部からのグラフィックスデータを受けるＩ／Ｏインタフェース回路１３からメインバス１５を介してレンダリング回路１４に与えられる。
なお、必要に応じて、グラフィックス描画等のデータは、メインプロセッサ１１等において、座標変換、クリップ処理、ライティング処理等のジオメトリ処理が行われる。
ジオメトリ処理が終わったグラフィックスデータは、三角形の各３頂点の頂点座標ｘ，ｙ，ｚ、輝度値Ｒ，Ｇ，Ｂ、描画しようとしている画素とディスプレイバッファ内の画素とのＴＧＢ値のブレンド係数α、対応するテクスチャ座標ｓ，ｔ，ｑとからなるポリゴンレンダリングデータＳ１１となる。

【００７１】

このポリゴンレンダリングデータＳ１１は、レンダリング回路１４のＤＤＡセットアップ回路１４１に入力される。
ＤＤＡセットアップ回路１４１においては、ポリゴンレンダリングデータＳ１１に基づいて、三角形の辺と水平方向の差分などを示す変分データＳ１４１が生成される。具体的には、開始点の値と終点の値、並びに、その間の距離を用いて、単位長さ移動した場合における、求めようとしている値の変化分である変分が算出され、変分データＳ１４１としてトライアングルＤＤＡ回路１４２に出力される。

【００７２】

トライアングルＤＤＡ回路１４２においては、変分データＳ１４１を用いて、、三角形内部の各画素における線形補間された（ｚ，Ｒ，Ｇ，Ｂ，α，ｓ，ｔ，ｑ，Ｆ）データが算出される。
そして、この算出された（ｚ，Ｒ，Ｇ，Ｂ，α，ｓ，ｔ，ｑ，Ｆ）データと、三角形の各頂点の（ｘ，ｙ）データとが、ＤＤＡデータＳ１４２として、トライアングルＤＤＡ回路１４２からテクスチャエンジン回路１４３に出力される。

【００７３】

テクスチャエンジン回路１４３においては、ＤＤＡデータＳ１４２が示す（ｓ，ｔ，ｑ）データについて、ｓデータをｑデータで除算する演算と、ｔデータをｑデータで除算する演算とが行われる。そして、除算結果「ｓ／ｑ」および「ｔ／ｑ」に、それぞれテクスチャサイズＵＳＩＺＥおよびＶＳＩＺＥが乗算され、テクスチャ座標データ（ｕ，ｖ）が生成される。

【００７４】

次に、テクスチャエンジン回路１４３からメモリＩ／Ｆ回路１４４を介して、テクスチャエンジン回路１４３からＳＲＡＭ１４８に、生成されたテクスチャ座標データ（ｕ，ｖ）を含む読み出し要求が出力され、メモリＩ／Ｆ回路１４４を介して、ＳＲＡＭ１４８に記憶された（Ｒ，Ｇ，Ｂ，α）データＳ１４８が読み出される。
次に、テクスチャエンジン回路１４３において、読み出した（Ｒ，Ｇ，Ｂ，α）データＳ１４８の（Ｒ，Ｇ，Ｂ）データと、前段のトライアングルＤＤＡ回路１４２からのＤＤＡデータＳ１４２に含まれる（Ｒ，Ｇ，Ｂ）データとが、（Ｒ，Ｇ，Ｂ，α）データＳ１４８に含まれるαデータが示す割合で混合され、ｘ，ｙ座標におけるテクスチャの色が算出され、画素データＳ１４３として生成される。
この画素データＳ１４３は、テクスチャエンジン回路１４３からメモリＩ／Ｆ回路１４４に出力される。

【００７５】

フルカラーの場合には、テクスチャバッファ１４７ａからのデータ（Ｒ，Ｇ，Ｂ，α）を直接用いればよいが、インデックスカラーの場合には、あらかじめ作成しておいたカラーインデックステーブル（Color Index Table ）のデータが、テクスチャＣＬＵＴ（Color Look Up Table)バッファ１４７ｄより、ＳＲＡＭ等で構成される一時保管バッファへ転送され、この一時保管バッファのＣＬＵＴを用いてカラーインデックスから実際のＲ，Ｇ，Ｂカラーが得られる。
なお、ＣＵＬＴがＳＲＡＭで構成された場合は、カラーインデックスをＳＲＡＭのアドレ

スに入力すると、その出力には実際のＲ，Ｇ，Ｂカラーが出てくるといった使い方となる。

【００７６】

そして、メモリＩ／Ｆ回路１４４において、テクスチャエンジン回路１４３から入力した画素データＳ１４３に対応するｚデータと、ｚバッファ１４７ｃに記憶されているｚデータとの比較が行われ、入力した画素データＳ１２によって描画される画像が、前回、ディスプレイバッファ２１に書き込まれた画像より、手前（視点側）に位置するか否かが判断される。

判断の結果、手前に位置する場合には、画像データＳ１４３に対応するｚデータでｚバッファ１４７ｃに記憶されたｚデータが更新される。

【００７７】

次に、メモリＩ／Ｆ回路１４４においては、αブレンディング／ディザ処理回路１４４０で、必要に応じて、画像データＳ１４３に含まれる（Ｒ，Ｇ，Ｂ）データと、既にディスプレイバッファ１４７ｂに記憶されている（Ｒ，Ｇ，Ｂ）データとが、画像データＳ１４３に対応するαデータが示す混合値で混合されるαブレンディング処理と、ディスプレイバッファ１４７ｂの容量等を考慮してデータを間引く（切り捨てる）ディザ処理とが並列的に行われ、処理後の（Ｒ，Ｇ，Ｂ）データがディスプレイバッファ１４７ｂに書き込まれる。

【００７８】

具体的には、αブレンディング／ディザ処理回路１４４０において、減算器１４４１および乗算器１４４３で、これから描画すべき現画像データＳの既にディスプレイバッファ１４７ｂに記憶されている画像データＤに対する更新量データが混合係数αを用いて求められ、これに並行して、第１の加算器１４４２で既にディスプレイバッファ１４７ｂに記憶されている画像データＤに雑音データＥが加えられる。

そして、乗算器１４４３で得られたデータと第１の加算器１４４２で得られたデータとが第２の加算器１４４４で加算され、２つの色の線形補間をしたデータに雑音データを加えたデータ｛α×（Ｓ－Ｄ）＋Ｄ＋Ｅ｝が求められ、その後、クランプ回路１４４５でカラーの有効値が抽出され、この抽出データから切り捨て回路１４４６でデータが間引かれて、ディスプレイバッファ１４７ｂに書き戻される。

【００７９】

メモリＩ／Ｆ回路１４４においては、今から描画しようとしている画素におけるテクスチャアドレスに対応したテクスチャを格納しているメモリブロックがそのテクスチャアドレスにより算出され、そのメモリブロックにのみ読みだし要求が出され、テクスチャデータが読み出される。

この場合、該当するテクスチャデータを保持していないメモリブロックにおいては、テクスチャ読み出しのためのアクセスが行われないため、描画により多くのアクセス時間を提供することが可能となっている。

【００８０】

描画においても同様に、今から描画しようとしている画素アドレスに対応する画素データを格納しているメモリブロックに対して、該当アドレスから画素データがモディファイ書き込み(Modify Write)を行うために読み出され、モディファイ後、同じアドレスへ書き戻される。

【００８１】

隠れ面処理を行う場合には、やはり同じように今から描画しようとしている画素アドレスに対応する奥行きデータを格納しているメモリブロックに対して、該当アドレスから奥行きデータがモディファイ書き込み(Modify Write)を行うために読み出され、必要ならばモディファイ後、同じアドレスへ書き戻される。

【００８２】

このようなメモリＩ／Ｆ回路１４４に基づくＤＲＡＭ１４７とのデータのやり取りにおいては、それまでの処理を複数並行処理することで、描画性能を向上させることができる。

特に、トライアングルＤＤＡ回路１４２とテクスチャエンジン１４３の部分を並列実行形式で、同じ回路に設ける（空間並列）か、または、パイプラインを細かく挿入する（時間並列）ことで、部分的に動作周波数を増加させるという手段により、複数画素の同時算出が行われる。

【００８３】

また、画素データは、メモリＩ／Ｆ回路１４４の制御のもと、表示領域において隣接した部分は、異なるＤＲＡＭモジュールとなるように配置される。

これにより、三角形のような平面を描画する場合には面で同時に処理される。このため、それぞれのＤＲＡＭモジュールの動作確率は非常に高い。

【００８４】

そして、図示しないＣＲＴに画像を表示する場合には、ＣＲＴコントロール回路１４５において、与えられた水平垂直同期周波数に同期して、表示アドレスが発生され、メモリＩ／Ｆ回路１４４へ表示データ転送の要求が出される。

メモリＩ／Ｆ回路１４４では、その要求に従い、一定のまとまった固まりで、表示データがＣＲＴコントロール回路１４５に転送される。

ＣＲＴコントロール回路１４５では、図示しないディスプレイ用ＦＩＦＯ(First In First Out)等にその表示データが貯えられ、一定の間隔でＲＡＭＤＡＣ１４６へＲＧＢのインデックス値が転送される。

【００８５】

ＲＡＭＤＡＣ１４６においては、ＲＡＭ内部にＲＧＢのインデックスに対するＲＧＢ値が記憶されていて、インデックス値に対するＲＧＢ値が図示しないＤ／Ａコンバータへ転送される。

そして、Ｄ／Ａコンバータでアナログ信号に変換されたＲＧＢ信号がＣＲＴへ転送される。

【００８６】

以上説明したように、本実施形態によれば、減算器１４４１および乗算器１４４３で、これから描画すべき現画像データＳの既にディスプレイバッファ１４７ｂに記憶されている画像データＤに対する更新量データを混合係数αを用いて求める処理と、第１の加算器１４４２で既にディスプレイバッファ１４７ｂに記憶されている画像データＤに雑音データＥを加える処理とを並行して行い、両処理で得られたデータを第２の加算器１４４４で加算することにより、２つの色の線形補間をしたデータに雑音データを加えたデータ｛α×（Ｓ－Ｄ）＋Ｄ＋Ｅ｝を求め、その後、クランプ回路１４４５でカラーの有効値を抽出し、この抽出データから切り捨て回路１４４６でデータを間引いて、ディスプレイバッファ１４７ｂに書き戻すαブレンディング／ディザ処理回路１４４０を設けたので、従来回路に比べて乗算器およびクランプ回路が１つずつ少ない回路構成であることから、回路規模が小さく、また、αブレンディング処理とディザ処理とが並列的に行われることから、演算時間を短縮でき、高速処理を実現できる利点がある。

【００８７】

また、本実施形態によれば、半導体チップ内部に内蔵されたＤＲＡＭ１４７に、表示データと少なくとも一つの図形要素が必要とするテクスチャデータを記憶させた構成を有することから、表示領域以外の部分にテクスチャデータを格納できることになり、内蔵ＤＲＡＭの有効利用が可能となり、高速処理動作、並びに低消費電力化を並立させるようにした画像処理装置が実現可能となる。

そして、単一メモリシステムを実現でき、すべてが内蔵された中だけで処理ができる。その結果、アーキテクチャとしても大きなパラダイムシフトとなる。

また、メモリの有効利用ができることで、内部に持っているＤＲＡＭのみでの処理が可能となり、内部にあるがゆえのメモリと描画システムの間の大きなバンド幅が、十分に活用可能となる。また、ＤＲＡＭにおいても特殊な処理を組み込むことが可能となる。

【００８８】

また、ＤＲＡＭにおける同一機能を独立した複数のモジュール１４７１～１４７４として

並列にもつことから、並列動作の効率を向上させることができる。単にデータのビット数が多いだけでは、データの使用効率は悪化し、性能向上できるのは一部の条件の場合に限定されることになる。平均的な性能を向上させるためには、ある程度の機能をもったモジュールを複数設けることで、ビット線の有効利用を行うことができる。

【００８９】

さらに、表示アドレス空間において、隣接するアドレスにおける表示要素が、それぞれ異なるＤＲＡＭのブロックになるように配置するので、さらにビット線の有効利用が可能となり、グラフィックス描画におけるような、比較的固まった表示領域へのアクセスが多い場合には、それぞれのモジュールが同時に処理できる確率が増加し、描画性能の向上が可能となる。

【００９０】

また、より多くのテクスチャデータを格納するために、インデックスカラーにおけるインデックスと、そのためのカラールックアップテーブル値を内蔵ＤＲＡＭ１４７内部に格納するので、テクスチャデータの圧縮が可能となり、内蔵ＤＲＡＭの効率良い利用が可能となる。

【００９１】

また、描画しようとしている物体の奥行き情報を、内蔵のＤＲＡＭに格納するので、描画と同時並行的に隠れ面処理を行うことが可能となる。

描画を行って、通常はそれを表示しようとするわけだが、ユニファイドメモリとして、テクスチャデータと表示データを同一のメモリシステムに同居させることができることから、直接表示に使わずに、描画データをテクスチャデータとして使ってしまうということも可能となる。

このようなことは、必要なときに必要なテクスチャデータを、描画によって作成する場合に有効となり、これもテクスチャデータを膨らませないための効果的な機能となる。

【００９２】

また、チップ内部にＤＲＡＭを内蔵することで、その高速なインターフェース部分がチップの内部だけで完結することになるため、大きな付加容量のＩ／Ｏバッファであるとか、チップ間配線容量をドライブする必要がなくなり、消費電力は内蔵しない場合に比較して小さくなる。

よって、さまざまな技術を使って、一つのチップの中だけですべてができるような仕組みは、今後の携帯情報端末等の身近なデジタル機器のためには、必要不可欠な技術要素となっている。

【００９３】

なお、本発明は上述した実施形態には限定されない。

また、上述した図１に示す３次元コンピュータグラフィックスシステム１０では、ＳＲＡＭ１４８を用いる構成を例示したが、ＳＲＡＭ１４８を設けない構成にしてもよい。

【００９４】

さらに、図１に示す３次元コンピュータグラフィックスシステム１０では、ポリゴンレンダリングデータを生成するジオメトリ処理を、メインプロセッサ１１で行う場合を例示したが、レンダリング回路１４で行う構成にしてもよい。

【００９５】

【発明の効果】

以上説明したように、本発明によれば、回路規模を小さくでき、また、αブレンディング処理とディザ処理とを並列的に行うことができ、演算時間を短縮でき、高速処理を実現できる利点がある。

【００９６】

また、半導体チップ内部にロジック回路と混載された記憶回路に、表示データと少なくとも一つの図形要素が必要とするテクスチャデータを記憶させた構成を有することから、表示領域以外の部分にテクスチャデータを格納できることになり、内蔵記憶回路の有効利用が可能となり、高速処理動作、並びに低消費電力化を並立させるようにした画像処理装置

が実現可能となる。

【００９７】

また、メモリにおける同一機能を独立した複数のモジュールとして並列にもつことから、並列動作の効率を向上させることができる。

【００９８】

さらに、表示アドレス空間において、隣接するアドレスにおける表示要素を、それぞれ異なるメモリのブロックになるように配置するので、グラフィックス描画におけるような、比較的固まった表示領域へのアクセスが多い場合には、それぞれのモジュールが同時に処理できる確率が増加し、描画性能の向上が可能となる。

【図面の簡単な説明】

【図１】本発明に係る３次元コンピュータグラフィックスシステムの構成を示すブロック図である。

【図２】本発明に係るＤＤＡセットアップ回路の機能を説明するための図である。

【図３】本発明に係るトライアングルＤＤＡ回路の機能を説明するための図である。

【図４】本発明に係るトライアングルＤＤＡ回路の頂点のソート処理を説明するための図である。

【図５】本発明に係るトライアングルＤＤＡ回路の水平方向の傾き算出処理を説明するための図である。

【図６】本発明に係るトライアングルＤＤＡ回路の頂点データの補間手順を説明するための図である。

【図７】本発明に係るトライアングルＤＤＡ回路の頂点データの補間手順を説明するためのフローチャートである。

【図８】本発明に係るαブレンディング／ディザ処理回路の構成例を示すブロック図である。

【図９】本発明に係るデータ格納方法を説明するための図である。

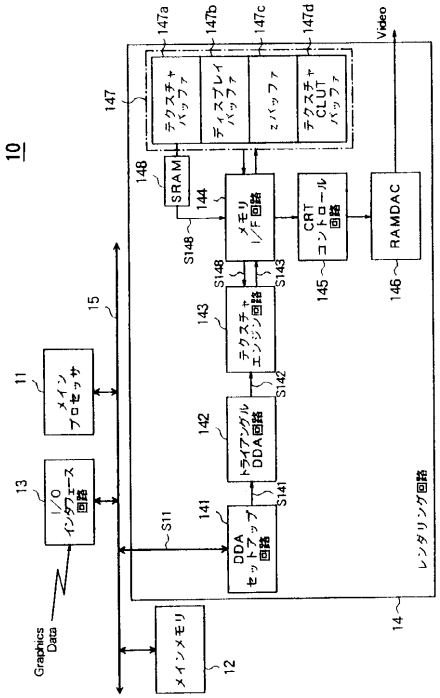
【図１０】３次元コンピュータグラフィックスシステムの基本的な概念を示すシステム構成図である。

【図１１】従来のαブレンディング処理回路およびディザ処理回路の構成例を示すブロック図である。

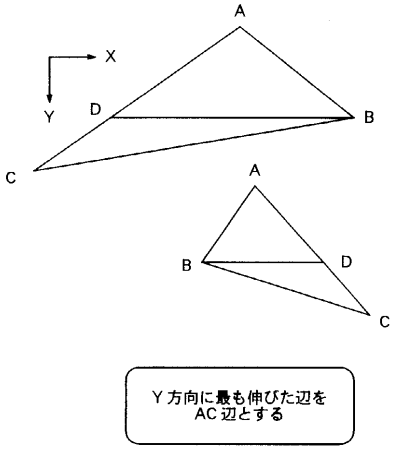
【符号の説明】

１０…３次元コンピュータグラフィックスシステム、１１…メインプロセッサ、１２…メインメモリ、１３…Ｉ／Ｏインタフェース回路、１４…レンダリング回路、１４１…ＤＤＡセットアップ回路、１４２…トライアングルＤＤＡ回路、１４３…テクスチャエンジン回路、１４４…メモリＩ／Ｆ回路、１４４０…αブレンディング／ディザ処理回路（描画データ制御回路）、１４４１…減算器、１４４２…第１の加算器、１４４３…乗算器、１４４４…第２の加算器、１４４５…クランプ回路、１４４６…切り捨て回路、１４４７～１４５０…メモリコントローラ、１４５１…ディストリビュータ、１４５…ＣＲＴコントローラ回路、１４６…ＲＡＭＤＡＣ回路、１４７…ＤＲＡＭ、１４７１～１４７４…ＤＲＡＭモジュール、１４７ａ…テクスチャバッファ、１４７ｂ…ディスプレイバッファ、１４７ｃ…ｚバッファ、１４７ｄ…テクスチャＣＬＵＴバッファ、１４８…ＳＲＡＭ。

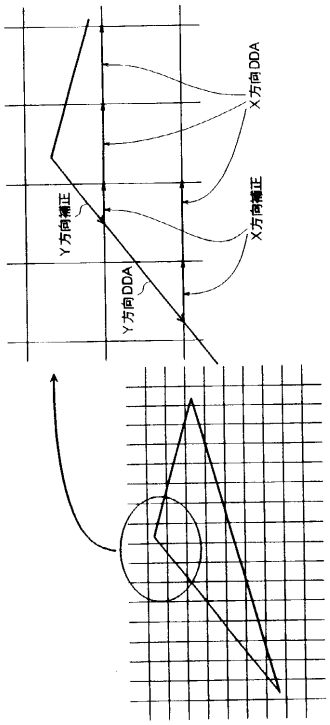
【図１】


10
147a
147b
147c
147d
Video
147
テクスチャバッファ
ディスプレイバッファ
Zバッファ
テクスチャCLUTバッファ
148
SRAM
144
メモリI/F回路
CRTコントロール回路
RAMDAC
145
146
S148
S143
143
テクスチャエンジン回路
S142
142
トライアングルDDA回路
S141
141
DDAセットアップ回路
S11
15
11
メインプロセッサ
13
I/Oインタフェース回路
Graphics Data
メインメモリ
12
レンダリング回路
14

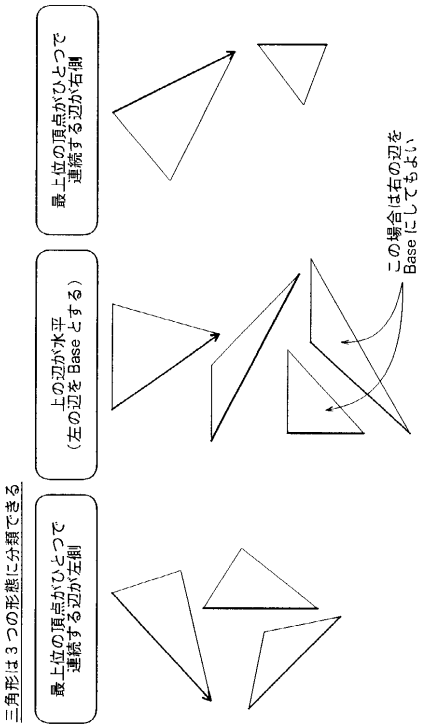
【図２】


A
X
Y
D
B
C
A
B
D
C
Y方向に最も伸びた辺を
AC辺とする

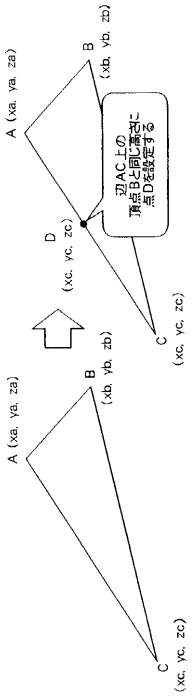
【図３】


X方向DDA
Y方向補正
X方向補正
Y方向DDA

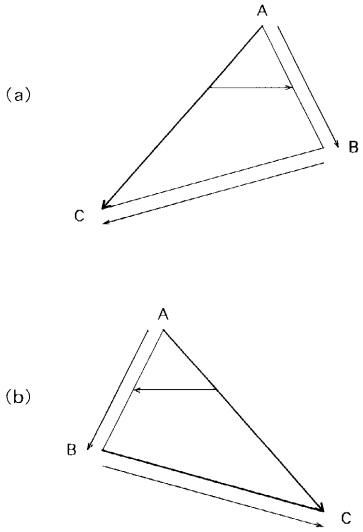
【図４】


三角形は３つの形態に分類できる
最上位の頂点がひとつで
連続する辺が左側
上の辺が水平
（左の辺をBaseとする）
最上位の頂点がひとつで
連続する辺が右側
この場合は右の辺を
Baseにしてもよい

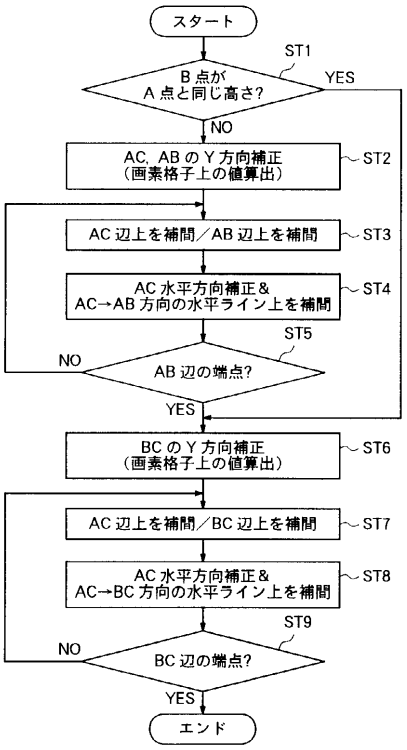
【図５】


A (xa, ya, za)
B (xb, yb, zb)
C (xc, yc, zc)
D (xc, yc, zc)
辺AC上の
頂点Bと同じ高さに
点Dを設定する
A (xa, ya, za)
B (xb, yb, zb)
C (xc, yc, zc)

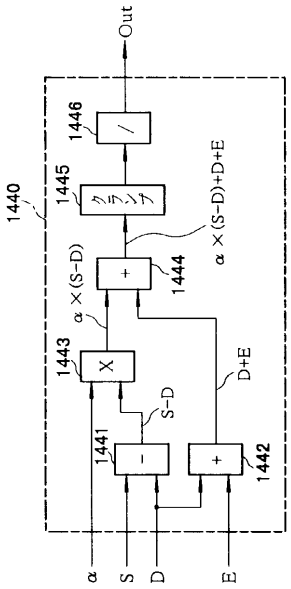
【図６】


(a)
A
B
C
(b)
A
B
C

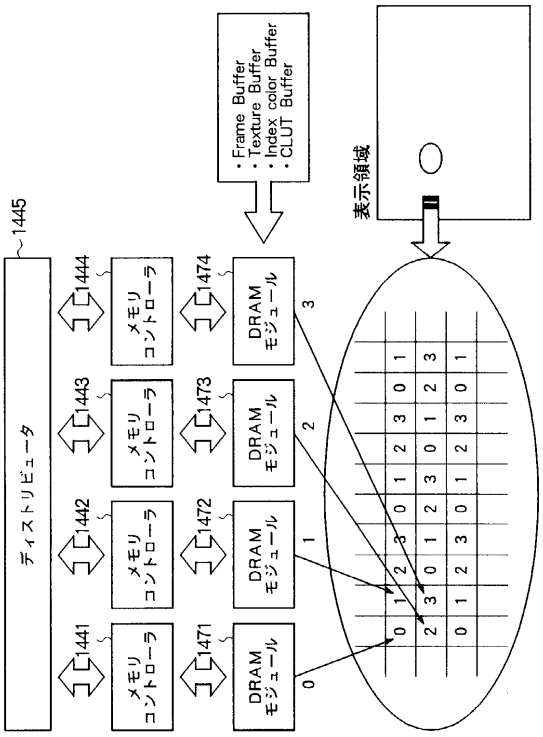
【図７】


スタート
ST1
B 点が
A 点と同じ高さ?
YES
NO
AC, AB の Y 方向補正
（画素格子上の値算出）
ST2
AC 辺上を補間／AB 辺上を補間
ST3
AC 水平方向補正＆
AC→AB 方向の水平ライン上を補間
ST4
ST5
NO
AB 辺の端点?
YES
BC の Y 方向補正
（画素格子上の値算出）
ST6
AC 辺上を補間／BC 辺上を補間
ST7
AC 水平方向補正＆
AC→BC 方向の水平ライン上を補間
ST8
ST9
NO
BC 辺の端点?
YES
エンド

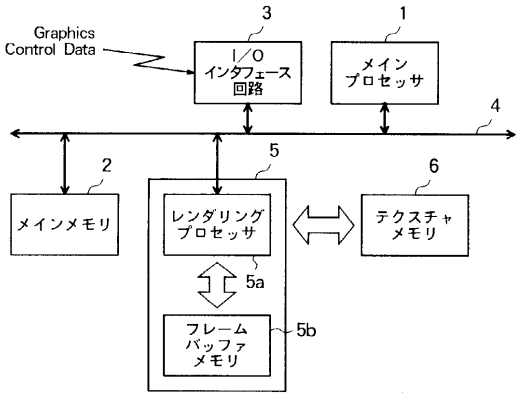
【図８】


Out
1446
1445
クランプ
1444
1443
X
1441
-
+
1442
1440
α×(S-D)
α×(S-D)+D+E
S-D
D+E
α
S
D
E

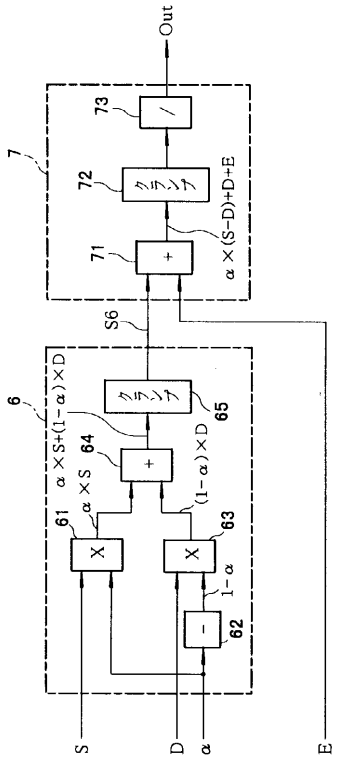
【図９】


・Frame Buffer
・Texture Buffer
・Index color Buffer
・CLUT Buffer
表示領域
1445
ディストリビュータ
1444
1443
1442
1441
メモリコントローラ
1474
1473
1472
1471
DRAMモジュール
3
2
1
0

【図１０】


Graphics Control Data
3
I/O インタフェース 回路
1
メイン プロセッサ
4
2
メインメモリ
5
レンダリング プロセッサ
5a
フレーム バッファ メモリ
5b
6
テクスチャ メモリ

【図１１】


Out
7
73
72
71
α×(S−D)+D+E
S6
6
α×S+(1−α)×D
65
64
α×S
(1−α)×D
61
63
1−α
62
S
D
α
E

フロントページの続き

審査官　伊知地　和之

(56)参考文献　特開平０７－２４９１１６（ＪＰ，Ａ）
特開平０９－０６２８５３（ＪＰ，Ａ）
特開平０７－３１９４３６（ＪＰ，Ａ）
特開平０５－３０７６１０（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.[7]，ＤＢ名)
G06T 11/20
H04N 1/40
CSDB(日本国特許庁)