

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第4411713号
(P4411713)

(45) 発行日 平成22年2月10日(2010.2.10)

(24) 登録日 平成21年11月27日(2009.11.27)

(51) Int.Cl.

F I

G O 6 T 15/00 (2006.01)

G O 6 T 15/00 1 O O A

請求項の数 12 (全 27 頁)

(21) 出願番号	特願平11-352855	(73) 特許権者	000132471
(22) 出願日	平成11年12月13日(1999.12.13)		株式会社セガ
(65) 公開番号	特開2001-167289(P2001-167289A)		東京都大田区羽田1丁目2番12号
(43) 公開日	平成13年6月22日(2001.6.22)	(74) 代理人	100094525
審査請求日	平成18年12月13日(2006.12.13)		弁理士 土井 健二
		(74) 代理人	100094514
			弁理士 林 恒徳
		(72) 発明者	清水 祐介
			東京都大田区羽田1丁目2番12号 株
			式会社セガ・エンタープライゼス内
		(72) 発明者	大矢 忠史
			東京都大田区羽田1丁目2番12号 株
			式会社セガ・エンタープライゼス内
		審査官	千葉 久博
			最終頁に続く

(54) 【発明の名称】 ブレンディング処理を含む画像処理装置及びその方法

(57) 【特許請求の範囲】

【請求項1】

重なり合う複数のポリゴンに対する画像処理を行って、表示用画像データを生成する画像処理装置において、

表示画面内の手前側から深さ方向に順に並べた時の n 番目のポリゴンに、ピクセル色データ $TXCOn$ と、不透明度 n または透明度 $(1 - n)$ と、光源色データ $INTn$ が与えられ、

前記画像処理装置は、

重なり合うピクセルの前記透明度 $(1 - n)$ の乗算値 Xn を、前記手前側から深さ方向の順に求めるブレンド処理ユニットと、

ピクセルの光源データに従って、深さ方向の各ピクセルに対する光源色データ $INTn$ を求める光源計算ユニットと、

ポリゴン色データ $PCOLn$ 及びテクスチャ座標に従って、前記深さ方向の各ピクセルに対するピクセル色データ $TXCOn$ を求めるテクスチャ処理ユニットと、

前記深さ方向の各ピクセルに対して、当該ピクセルのピクセル色データ $TXCOn$ を、光源色データ $INTn$ 、不透明度 n 、及び一つ手前側のピクセルの前記乗算値 $Xn-1$ に応じて修正し、当該修正したピクセル色データを累積するカラー合成処理を行うカラー合成ユニットとを有し、

前記光源計算ユニット及びテクスチャ処理ユニットの少なくとも一方が、対応するデータ生成処理を並列に行い、前記カラー合成ユニットが、前記カラー合成処理を深さ方向の

10

20

ピクセルについて順不同で行うことを特徴とする画像処理装置。

【請求項 2】

請求項 1 において、

更に、フレーム内の複数のポリゴンデータまたはそのピクセルデータを、前記表示画面内の手前側から深さ方向に順番に並び替えるソーターユニットを有し、当該ソーターユニットが、光源計算ユニット及びテクスチャ処理ユニットに、前記ピクセルデータを供給し、前記ブレンド処理ユニットに前記ピクセルデータを手前側から深さ方向の順に供給することを特徴とする画像処理装置。

【請求項 3】

請求項 1 において、

前記カラー合成処理は、処理中のピクセルのピクセル色データ $TXC0n$ 、光源色データ $INTn$ 、不透明度 n 、及び一つ手前側のピクセルの前記乗算値 X_{n-1} との乗算値を求め、それらを累積する処理であることを特徴とする画像処理装置。

【請求項 4】

請求項 3 において、

前記カラー合成処理により求められた累積値が、前記表示用画像データとして使用されることを特徴とする画像処理装置。

【請求項 5】

請求項 1 において、

更に、深さ方向のピクセルの前記乗算値 Xn 、ピクセル色データ $TXC0n$ 、光源色データ $INTn$ 、不透明度 n のデータを記録するピクセルワークメモリユニットを有し、前記カラー合成ユニットは、該ピクセルワークメモリユニットに上記データが全て記録されたピクセルについて、前記カラー合成処理を行うことを特徴とする画像処理装置。

【請求項 6】

請求項 2 において、

前記ブレンド処理ユニットは、更に、前記ソーターユニットから供給されるポリゴンの不透明度または透明度と、テクスチャの不透明度または透明度から、ピクセルの不透明度 n または透明度 $(1 - n)$ を生成することを特徴とする画像処理装置。

【請求項 7】

請求項 1 において、

前記テクスチャ処理ユニットは、更に、ポリゴン色データ $PCOLn$ と前記テクスチャ座標に対応するテクスチャの色データ T とに従って、前記ピクセル色データ $TXC0n$ を求める複数のテクスチャ処理回路を有し、空き状態にあるテクスチャ処理回路が順次前記ピクセル色データを生成することを特徴とする画像処理装置。

【請求項 8】

請求項 1 において、

前記光源計算ユニットは、光源色データ $INTn$ を求める複数の光源計算回路を有し、空き状態にある光源処理回路が順次前記光源色データを生成することを特徴とする画像処理装置。

【請求項 9】

重なり合う複数のポリゴンに対する画像処理を行って、表示用画像データを生成する画像処理装置において、

表示画面内の手前側から深さ方向に順に並べた時の n 番目のポリゴンに、ピクセル色データ $TXC0n$ と、光源色データ $INTn$ と、ディスティネーション混合係数 DEn 及びソース混合係数 SRn についてのデータが与えられ、

前記画像処理装置は、

重なり合うピクセルの前記ソース混合係数 SRn の乗算値 Xn を、前記手前側から深さ方向の順に求めるブレンド処理ユニットと、

ピクセルの光源データに従って、深さ方向の各ピクセルに対する光源色データ $INTn$ を求める光源計算ユニットと、

10

20

30

40

50

ポリゴン色データPCOLn及びテクスチャ座標に従って、前記深さ方向の各ピクセルに対するピクセル色データTXCOnを求めるテクスチャ処理ユニットと、

前記深さ方向の各ピクセルに対して、当該ピクセルのピクセル色データTXCOnを、光源色データINTn、ディスティネーション混合係数Dn、及び一つ手前側のピクセルの前記乗算値Xn-1に応じて修正し、当該修正したピクセル色データを累積するカラー合成処理を行うカラー合成ユニットとを有し、

前記光源計算ユニット及びテクスチャ処理ユニットの少なくとも一方が、対応するデータ生成処理を並列に行い、前記カラー合成ユニットが、前記カラー合成処理を深さ方向のピクセルについて順不同で行うことを特徴とする画像処理装置。

【請求項 10】

請求項 9 において、

前記カラー合成処理は、処理中のピクセルのピクセル色データTXCOn、光源色データINTn、ディスティネーション混合係数Dn、及び、一つ手前側のピクセルの前記乗算値Xn-1との乗算値を求め、それらを累積する処理であり、当該累積値が前記表示用画像データとして使用されることを特徴とする画像処理装置。

【請求項 11】

重なり合う複数のポリゴンに対する画像処理を行って、表示用画像データを生成する画像処理方法において、

表示画面内の手前側から深さ方向に順に並べた時のn番目のポリゴンに、ピクセル色データTXCOnと、不透明度nまたは透明度(1 - n)と、光源色データINTnとが与えられ、

前記画像処理方法は、

コンピュータが、重なり合うピクセルの前記透明度(1 - n)の乗算値Xnを、前記手前側から深さ方向の順に求めるブレンド処理工程と、

コンピュータが、ピクセルの光源データに従って、深さ方向の各ピクセルに対する光源色データINTnを求める光源計算処理工程と、

コンピュータが、ポリゴン色データPCOLn及びテクスチャ座標に従って、前記深さ方向の各ピクセルに対するピクセル色データTXCOnを求めるテクスチャ処理工程と、

コンピュータが、前記深さ方向の各ピクセルに対して、順不同で、当該ピクセルのピクセル色データTXCOnを、光源色データINTn、不透明度n、及び一つ手前側のピクセルの前記乗算値Xn-1に応じて修正し、当該修正したピクセル色データを累積するカラー合成処理を行うカラー合成工程とを有し、

前記光源計算処理工程及びテクスチャ処理工程の少なくとも一方で、対応するデータ生成処理を並列に行うことを特徴とする画像処理方法。

【請求項 12】

重なり合う複数のポリゴンに対する画像処理を行って、表示用画像データを生成する画像処理方法において、

表示画面内の手前側から深さ方向に順に並べた時のn番目のポリゴンに、ピクセル色データTXCOnと、光源色データINTnと、ディスティネーション混合係数Dn及びソース混合係数SRnについてのデータが与えられ、

前記画像処理方法は、

コンピュータが、重なり合うピクセルの前記ソース混合係数SRnの乗算値Xnを、前記手前側から深さ方向の順に求めるブレンド処理工程と、

コンピュータが、ピクセルの光源データに従って、深さ方向の各ピクセルに対する光源色データINTnを求める光源計算工程と、

コンピュータが、ポリゴン色データPCOLn及びテクスチャ座標に従って、前記深さ方向のピクセルに対するピクセル色データTXCOnを求めるテクスチャ処理工程と、

コンピュータが、前記深さ方向の各ピクセルに対して、順不同で、当該ピクセルのピクセル色データTXCOnを、光源色データINTn、ディスティネーション混合係数Dn、及び一つ手前側のピクセルの前記乗算値Xn-1に応じて修正し、当該修正したピクセル色データを

10

20

30

40

50

累積するカラー合成処理を行うカラー合成工程とを有し、

前記光源計算工程及びテクスチャ処理工程の少なくとも一方で、対応するデータ生成処理を並列に行うことを特徴とする画像処理方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、半透明ポリゴンを含む複数のポリゴンに対する画像処理装置及びその方法にかかり、特に、半透明ポリゴンに必要な透明度を利用したブレンディング処理や重なり合う複数のポリゴンに対する一般的なブレンディング処理を、より効率的に且つ短時間で処理することができる画像処理装置及びその方法に関する。

10

【0002】

【従来の技術】

ゲーム装置やシミュレーション装置において、操作入力に応じてキャラクタ等を構成する複数のポリゴンの位置を求め、それらのポリゴンのうち表示されるポリゴンに対してレンダリング処理を行って画像データを生成し、それらを表示することが行われる。

【0003】

半透明ポリゴンに対するレンダリング処理は、不透明ポリゴンと異なり、その後方（奥側）に位置する他のポリゴンとのブレンディング処理を伴うので、表示画面内の奥側の半透明ポリゴンから順にブレンディング処理を含むレンダリング処理を行うことが必要である。

20

【0004】

一方、不透明のポリゴンに対するレンダリング処理は、表示画面内の奥行きを示すZ値を利用した陰面消去処理を行うことで、複数のポリゴンのレンダリングの順番に制約を課すことなく、重なり合う部分の処理が行われる。また、不透明ポリゴンと半透明ポリゴンとが混在する場合は、上記のZ値を利用した陰面消去処理によって不透明ポリゴンを任意の順番で処理し、その後、半透明ポリゴンを奥側から順に処理する方法が一般的である。

【0005】

【発明が解決しようとする課題】

ところが、レンダリング処理において、処理すべきポリゴン数が多くなると、決められたフレーム間の時間内でレンダリング処理が完了しない場合がある。その場合、画像処理の質を重視すると全ての半透明ポリゴンのレンダリング処理が終了するまで待たなければならず、表示中のキャラクタの動きが停止することになり、ゲーム装置あるいはシミュレーション装置において不都合である。また、キャラクタの動きを重視するとフレーム間の時間内で処理されないポリゴンが存在することになる。その場合、半透明ポリゴンのレンダリング処理が、より奥側のポリゴンから処理されるので、画像内で最も重要なキャラクタの一つである手前側のポリゴンの処理が切り捨てられることになる。

30

【0006】

更に、不透明ポリゴンと半透明ポリゴンのレンダリング処理は、上記した通り、それぞれの効率を重視すると、それらのポリゴンを分けて別々に処理する必要があり、画像処理の負担を増大させる。かかる負担の増大は、上記と同様にフレーム間の時間内での画像処理の完了を困難にさせる。

40

【0007】

更に、半透明ポリゴンに対する不透明度を利用したブレンディング処理に限定されず、一般的なブレンディング処理においても、表示画面のより奥側のポリゴンの色データを先に求め、その色データと次に処理するポリゴンの色データとに対して、所定のソース混合計数とディスティネーション混合計数とによりブレンディング処理することが提案されている。この場合も、表示画面の奥側のポリゴンから順番に処理することが要求され、半透明ポリゴンの場合と同様の課題を有する。

【0008】

そこで、本出願人は、上記課題を解決する発明を提案した。例えば、特願平11-225426号

50

、1999年8月9日出願の「ブレンディング処理を含む画像処理装置及びその方法」である。本件は、本願出願時点では未公開であり、従って公知ではない。この発明では、半透明ポリゴンのレンダリング処理を、表示画面内の手前側、つまり視点側のポリゴンから深さ方向に順に行うことを特徴とする。そのために、表示画面内の手前側から順に並べた時の n 番目のポリゴンに、色データ T_n と不透明度 α_n とが与えられる場合、ワークデータ X_n として、

$$X_n = (1 - \alpha_n) \cdot X_{n-1}$$

を求め、更に画像データ D_n として、

$$D_n = D_{n-1} + T_n \cdot \alpha_n \cdot X_{n-1}$$

を求め、ワークデータ X_n 及び画像データ D_n を求める上記の演算処理を全てのポリゴンに対して行う。そして、最後の画像データ D_n を、表示用の画像データとして利用する。

10

【0009】

本発明は、上記の画像処理アルゴリズムを利用して、より高速にレンダリング処理を行うことができる新規な画像処理装置及び画像処理方法を提供するものである。

【0010】

本発明の目的は、半透明ポリゴンの画像処理において、フレーム間の時間内により手前側のポリゴンのレンダリング処理を確実に行うことができる画像処理装置、その方法を提供することにある。

【0011】

更に、本発明の目的は、不透明ポリゴンと半透明ポリゴンが混在したポリゴンの画像処理において、両ポリゴンを区別することなくレンダリング処理することができ、更に処理効率を高めた画像処理装置、その方法を提供することにある。

20

【0012】

更に、本発明の目的は、一般的なブレンディング処理において、処理効率を高めた画像処理装置、その画像処理方法を提供することにある。

【0013】

【課題を解決するための手段】

本発明の第1の側面は、半透明ポリゴンのレンダリング処理を、表示画面内の手前側のポリゴンから順にワークデータ X_n を求めると共に、光源からの色データ INT_n を求める光源計算処理や、テクスチャカラーなどを利用して色データ $TXCOn$ を求めるテクスチャ処理を、複数のポリゴンに対して並列に行う。そして、上記のワークデータ X_n 、光源色データ INT_n 、色データ $TXCOn$ 及び不透明度 α_n とから、画像データ $F B_n$ を求めて累積するカラー合成処理を、複数のポリゴンに対して順不同で行うことを特徴とする。

30

【0014】

そのために、表示画面内の手前側から順に並べた時の n 番目のポリゴンに、ピクセル色データ $TXCOn$ と不透明度 α_n と光源色データ INT_n が与えられる場合、ワークデータ X_n として、

$$X_n = (1 - \alpha_n) \cdot X_{n-1}$$

を全てのポリゴンに対して表示画面内の手前側から順番に求める。更に、画像データ $F B_n$ として、乗算値 $INT_n \cdot TXCOn \cdot \alpha_n \cdot X_{n-1}$ を全てのポリゴンに対して求め、累積する。即ち、

40

$$F B_n = (INT_n \cdot TXCOn \cdot \alpha_n \cdot X_{n-1})$$

の演算を行う、これは、重なり合うポリゴンの各画像データ $F B_n$ を加算する演算であるので、各画像データ $F B_n$ は、深さ方向の複数のポリゴンに対して順不同で求めることができる。そして、全て累積した画像データ $F B$ を、表示用の画像データとして利用する。

【0015】

上記において、ワークデータ X_n の初期値 X_0 は、例えば1が好ましい。また、画像データ $F B_n$ の初期値 $F B_0$ は、例えば0が好ましい。

【0016】

上記の画像処理では、画像データ $F B$ は、ピクセル色データ $TXCOn$ と、不透明度 α_n と、光

50

源色データ INT_n 及びワークデータ X_n の積算した値 $INT_n \cdot TXCO_n \cdot \alpha_n \cdot X_{n-1}$ を加算（累積）した値（ $n = 0 \sim j$ 、 j は重なったポリゴンの枚数）である。そして、各ポリゴンに対する積算値 $INT_n \cdot TXCO_n \cdot \alpha_n \cdot X_{n-1}$ を求める演算は、他のポリゴンに影響を受けないので、上記した通り、順不同で行うことができる。そして、光源からポリゴンに与えられる影響を示す光源色データ INT_n の演算は、ポリゴンの種類に応じて簡単な場合と複雑な場合とが混在する。また、テクスチャカラー等を利用してピクセル色データ $TXCO_n$ を求める演算処理も、ポリゴンに応じて簡単な場合と複雑な場合とが混在する。一方、透明度の累積乗算値であるワークデータ $X_n = (1 - \alpha_n) \cdot X_{n-1}$ は、他のポリゴンのワークデータ X_{n-1} の影響を受けるので、スクリーンの手前から深さ方向に順番に行う必要がある。但し、この演算処理自体は、単純であり、処理工数は少ない。

10

【0017】

そこで、本発明では、演算処理工数が比較的高い光源計算処理やテクスチャ処理を、複数のポリゴンに対して並列に行い、これらの処理が終了したポリゴンから順に、積算値 $INT_n \cdot TXCO_n \cdot \alpha_n \cdot X_{n-1}$ を求める処理とそれらを累積加算する処理とからなるカラー合成処理を行う。その結果、従来非常に演算負荷が重かった光源計算処理やテクスチャ処理を短時間で効率的に行うことができる。

【0018】

上記のワークデータ X_n は、手前側（視点側）の n 枚の半透明ポリゴンの透明度（ $1 - \alpha_n$ ）が乗算された値を意味する。従って、重なり合う複数のポリゴンに対して、手前側のポリゴンから順番にワークデータを求める処理をすることで、全てのポリゴンに対するワークデータ X_n を順番に求めることができる。それぞれのポリゴンのレンダリング処理では、ピクセル色データ $TXCO_n$ を、それより手前の $n - 1$ 枚のポリゴンの透明度を乗算したワークデータ値 X_{n-1} と、透明度 α_n と、光源色データ INT_n に応じて、演算処理して画像データ $F B_n$ を求める。そして、それぞれのポリゴンに対して求められた画像データ $F B_n$ が累積（加算）されて、表示用の画像データ $F B$ が求められる。

20

【0019】

本発明の別の側面は、重なり合う複数のポリゴンに対する画像処理を行って、表示用画像データを生成する画像処理装置において、表示画面内の手前側から深さ方向に順に並べた時の n 番目のポリゴンに、ピクセル色データ $TXCO_n$ と、不透明度 α_n または透明度（ $1 - \alpha_n$ ）と、光源色データ INT_n が与えられ、前記画像処理装置は、重なり合うピクセルの前記透明度（ $1 - \alpha_n$ ）の乗算値 X_n を、前記手前側から深さ方向の順に求めるブレンド処理ユニットと、ピクセルの光源データに従って、深さ方向の各ピクセルに対する光源色データ INT_n を求める光源計算ユニットと、ポリゴン色データ $PCOL_n$ 及びテクスチャ座標に従って、前記深さ方向の各ピクセルに対するピクセル色データ $TXCO_n$ を求めるテクスチャ処理ユニットと、前記深さ方向の各ピクセルに対して、当該ピクセルのピクセル色データ $TXCO_n$ を、光源色データ INT_n 、不透明度 α_n 、及び一つ手前側のピクセルの前記乗算値 X_{n-1} に応じて修正し累積するカラー合成処理を行うカラー合成ユニットとを有し、前記光源計算ユニット及びテクスチャ処理ユニットの少なくとも一方が、対応するデータ生成処理を並列に行い、前記カラー合成ユニットが、前記カラー合成処理を深さ方向のピクセルについて順不同で行うことを特徴とする画像処理装置である。または、その画像処理方法である。

30

40

【0020】

更に、本発明の別の側面は、重なり合う複数のポリゴンに対する画像処理を行って、表示用画像データを生成する画像処理装置において、表示画面内の手前側から深さ方向に順に並べた時の n 番目のポリゴンに、ピクセル色データ $TXCO_n$ と、光源色データ INT_n と、ディスティネーション混合係数 DE_n 及びソース混合係数 SR_n についてのデータが与えられ、

50

前記画像処理装置は、
 重なり合うピクセルの前記ソース混合係数 SR_n の乗算値 X_n を、前記手前側から深さ方向の順に求めるブレンド処理ユニットと、
 ピクセルの光源データに従って、深さ方向の各ピクセルに対する光源色データ INT_n を求める光源計算ユニットと、
 ポリゴン色データ $PCOL_n$ 及びテクスチャ座標に従って、前記深さ方向の各ピクセルに対するピクセル色データ $TXCOn$ を求めるテクスチャ処理ユニットと、
 前記深さ方向の各ピクセルに対して、当該ピクセルのピクセル色データ $TXCOn$ を、光源色データ INT_n 、ディスティネーション混合係数 DE_n 、及び一つ手前側のピクセルの前記乗算値 X_{n-1} に応じて修正し累積するカラー合成処理を行うカラー合成ユニットとを有し、
 前記光源計算ユニット及びテクスチャ処理ユニットの少なくとも一方が、対応するデータ生成処理を並列に行い、前記カラー合成ユニットが、前記カラー合成処理を深さ方向のピクセルについて順不同で行うことを特徴とする画像処理装置である。また、その画像処理方法である。

10

【0021】

【発明の実施の形態】

以下、本発明の実施の形態について図面に従って説明する。しかしながら、本発明の技術的範囲がその実施の形態に限定されるものではない。

【0022】

最初に、先願の特願平11-225426号で提案した、ブレンディング処理方法について説明する。

20

【0023】

図1は、半透明ポリゴンの重なり例を説明する図である。この例では、3枚の半透明ポリゴン P_1 、 P_2 、 P_3 と、1枚の不透明ポリゴン P_4 とが、表示画面内の奥行き方向（Z値）について手前から順に重なっている。従って、領域 C_4 の画像データ C_4 は、不透明ポリゴン P_4 のテクスチャデータから環境光等に応じて形成される色データ T_4 そのままである。

【0024】

本明細書において、ポリゴンの色データ T は、例えばポリゴン自体の色データ、模様を含むテクスチャ色データ、更に、テクスチャ色データやポリゴン自体の色データにシェーディング処理した後の色データ等を含む。それらのいずれかの色データを代表して、ポリゴンの色データ T と称する。また、ブレンディング処理などにより求められた色データは、表示に利用される画像データ C である。従って、画像データ C も一種の色データである。

30

【0025】

即ち、 $C_4 = T_4$ (1)

また、領域 C_3 は、不透明ポリゴン P_4 上に半透明ポリゴン P_3 が重なっているので、その画像データは、半透明ポリゴン P_3 の不透明度 α_3 （ $\alpha_3 = 1$ で不透明、 $\alpha_3 = 0$ で透明、 $1 > \alpha_3 > 0$ で半透明）及び色データ T_3 に従って、ポリゴン P_4 、 P_3 の色データをブレンディングした色データとなる。即ち、従来の一般的なレンダリング方法によれば、領域 C_3 の色データは、

40

$C_3 = \alpha_3 T_3 + (1 - \alpha_3) C_4$ (2)

で求められる。

【0026】

同様に、領域 C_2 の画像データ C_2 も、半透明ポリゴン P_2 の不透明度 α_2 及び色データ T_2 と領域 C_3 の色データ C_3 とから、

$C_2 = \alpha_2 T_2 + (1 - \alpha_2) C_3$ (3)

で求められ、更に、領域 C_1 の画像データ C_1 も、

$C_1 = \alpha_1 T_1 + (1 - \alpha_1) C_2$ (4)

で求められる。

【0027】

50

以上の通り、半透明ポリゴンに対するレンダリング処理は、画像データ C_n を求めるために、他のポリゴンとの重なり合う領域において、より奥側の画像データ C_{n+1} を再帰的に参照してブレンディング処理を行う必要がある。従って、表示画面内の奥側から順にポリゴンのレンダリング処理を行うのが一般的である。

【 0 0 2 8 】

そこで、先願で提案した発明は、上記した従来の課題を解決するために、表示画面内の手前側から順にポリゴンのレンダリング処理を行うことを特徴とする。この発明によれば、上記の式 (2)、(3)、(4) から、 C_1 について開くと、

$$C_1 = \alpha_1 T_1 + (1 - \alpha_1) \{ \alpha_2 T_2 + (1 - \alpha_2) [\alpha_3 T_3 + (1 - \alpha_3) C_4] \} \quad (5)$$

となり、更に、 T_n についてまとめると、

$$C_1 = T_1 \alpha_1 + T_2 \alpha_2 (1 - \alpha_1) + T_3 \alpha_3 (1 - \alpha_1) (1 - \alpha_2) + C_4 (1 - \alpha_1) (1 - \alpha_2) (1 - \alpha_3) \quad (5)$$

ここで、ポリゴン P_4 が不透明ポリゴンであり、不透明度データ $\alpha_4 = 1$ とし、更に、任意のワークデータ X_n として、

$$X_n = (1 - \alpha_1) (1 - \alpha_2) \dots (1 - \alpha_n) = X_{n-1} (1 - \alpha_n) \quad (6)$$

と定義するとする。但し、 $X_0 = 1$ とする。

【 0 0 2 9 】

上記式 (6) を式 (5) に挿入すると、

$$C_1 = T_1 \alpha_1 X_0 + T_2 \alpha_2 X_1 + T_3 \alpha_3 X_2 + C_4 X_3 = T_1 \alpha_1 X_0 + T_2 \alpha_2 X_1 + T_3 \alpha_3 X_2 + T_4 \alpha_4 X_3$$

となり、各々の $T_n \alpha_n X_{n-1}$ に着目すれば、

$$F B_n = F B_{n-1} + T_n \alpha_n X_{n-1} \quad (7)$$

とすることができる。但し、 $F B_0 = 0$ である。その結果、 $n = 4$ の場合は、

$$F B_4 = C_1$$

が成立する。

【 0 0 3 0 】

即ち、従来のブレンディング処理と同じ処理を、上記の式 (6) と (7) を利用することで、表示画面内の手前側から順にポリゴンのデータとして、色データ T_n とその不透明度 α_n を与えて、全ての重なり合うポリゴンに対する演算処理を行うことにより、最終的に得られる画像データ $F B_n$ は、図 1 の領域 C_1 の画像データ C_1 と同じになる。従って、新たに定義したワークデータ X_n を利用して、画像データ $F B_n = F B_{n-1} + T_n \alpha_n X_{n-1}$ を求めることができる。これらの数式のインデックスから明らかな通り、これらの演算において、より奥側のポリゴンのデータを再帰的に呼び出す必要はなくなる。

【 0 0 3 1 】

図 2 は、上記の演算を利用する本発明のレンダリング処理を、図 1 の例に適用した時の演算式を示す図表である。図 2 (1) に、先願の発明の場合を、図 2 (2) に従来例の場合をそれぞれ示す。

【 0 0 3 2 】

図 2 の図表から明らかな通り、従来例のレンダリング処理によれば、半透明のポリゴンに対して、奥側のポリゴン P_4 の色データ T_4 と不透明度 α_4 と初期値 C_5 を利用して、領域 C_4 の画像データ C_4 が求められ、更に、一つ手前のポリゴン P_3 の色データ T_3 と不透明度 α_3 と初期値 C_4 を利用して、領域 C_3 の画像データ C_3 が求められる。更に、領域 C_2 の画像データ C_2 及び領域 C_1 の画像データ C_1 が順に求められる。

【 0 0 3 3 】

10

20

30

40

50

それに対して、先願の発明によれば、上記の式(6)、(7)を利用することで、最初に最も手前に位置するポリゴン P_1 の色データ T_1 と不透明度 α_1 と初期値 X_0 と FB_0 を利用して、途中の画像データ FB_1 が求められ、更に、一つ奥側のポリゴン P_2 の色データ T_2 と不透明度 α_2 と上記の途中の画像データ FB_1 を利用して、次の途中の画像データ FB_2 が求められる。更に、同様にして次の途中の画像データ FB_3 、 FB_4 が求められる。そして、全てのポリゴンのデータに対する処理が終了した最終画像データ FB_4 が、領域 C_1 の画像データ C_1 として利用される。

【0034】

従って、この先願発明によれば、半透明ポリゴンに対するレンダリング処理は、手前側のポリゴンに対するデータ(色データ T と不透明度データ)を利用して次々に演算処理することができる。更に、後述する具体例で明らかになるが、より奥側のポリゴンに対するレンダリング処理では、ワークデータ X_{n-1} が小さくなる傾向にあり、そのワークデータ X_{n-1} を利用して求められる途中の画像データ FB_n も変化が少なくなる傾向にある。従って、ある程度ワークデータ X_{n-1} が小さくなると、あるいは0になると、それ以降の奥側のポリゴンのデータに対するレンダリング処理を省略しても、表示される画像への影響は殆どないことが理解される。

【0035】

更に、上記のレンダリング処理の演算において、不透明ポリゴンに対しては、単に不透明度 α_n を1(不透明)とすれば、上記の演算をそのまま適用して処理することができる。即ち、不透明度 $\alpha_n = 1$ の場合は、ワークデータ $X_n = 0$ となり、それより奥側のポリゴンのレンダリング処理の結果に変化はなくなり、実質的にそれらの奥側のポリゴンに対するレンダリングは不要になる。従って、この先願発明のレンダリング処理を利用することにより、半透明ポリゴンと不透明ポリゴンとを混在させても、より手前側のポリゴンから順に、半透明と不透明を区別なく処理することができる。

【0036】

上記したワークデータ X_n は、物理的には、 n 枚のポリゴンを重ねた時の透明度($1 -$)の累積乗算された値を意味する。即ち、上記の式(6)から明らかな通り、ワークデータ X_n は、各ポリゴンの透明度($1 -$)が全て乗算された値である。つまり、透明度の高いポリゴン、例えば($1 -$)が1に近いポリゴンが重ねられる場合に、その奥側に位置する n 枚目のポリゴンの色データ T_n を追加する場合は、それより手前の $n - 1$ 枚の重なったポリゴンの透明度($1 -$)の累積乗算値に応じて、画像データ FB_{n-1} に影響を与える。そして、望ましくは、 n 枚目のポリゴンの不透明度 α_n に応じた色データ T_n が、加えられる。 n 枚目のポリゴン自身が透明なポリゴン($1 - \alpha_n = 1$)の場合は、その色データ T_n は加えられるべきではないからである。

【0037】

以上の様に、上記の式(7)

$$FB_n = FB_{n-1} + T_n \cdot X_{n-1}$$

によるレンダリング処理は、第1に、処理しようとする奥側のポリゴンの色データ T_n を、それより手前の $n - 1$ 枚のポリゴンの透明度を累積した値 X_{n-1} に応じて、手前の $n - 1$ 枚のポリゴンが重ねられた画像データ FB_{n-1} に加える処理である。

【0038】

第2に、上記の式(7)によるレンダリング処理は、処理中のポリゴンの透明度に応じて、その色データ T_n を加えることを行う。即ち、処理中のポリゴンの不透明度 α_n に応じた色データ T_n の成分を、それより手前の $n - 1$ 枚のポリゴンの透明度の累積された値 X_{n-1} に応じて、 $n - 1$ 枚重なったポリゴンの画像データ FB_{n-1} に加えることを意味する。

【0039】

そして、透明度の累積値 X_n の初期値 $X_0 = 1$ の意味は、最初のポリゴンより手前には、別のポリゴンが存在しないので、透明度は最大であることを意味する。従って、最初のポリゴンの色データ T_1 は、そのまま画像データ FB_1 として利用される。但し、最初のポ

10

20

30

40

50

リゴン自身の不透明度 α_1 に応じた色データ T_1 の成分が画像データ $F B_1$ として利用される。尚、画像データ $F B_n$ の初期値 $F B_0 = 0$ の意味は、色の輝度値が 0 であることを意味し、何ら色の成分がない真っ黒の状態を意味する。画像データの初期値 $F B_0$ は、必ずしも真っ黒である必要はない。

【 0 0 4 0 】

以上、先願で提案した画像処理方法を説明した。この画像処理は、次の演算式に従う。

【 0 0 4 1 】

$$F B_n = F B_{n-1} + T_n \cdot X_{n-1} \quad (7)$$

$$X_n = X_{n-1} \cdot (1 - \alpha_n) \quad (6)$$

そこで、上記の (7) 式を書き換えると、

$$F B_n = (T_n \cdot X_{n-1}) \quad (8)$$

とすることができる。これは、表示画面内の深さ方向において、各ポリゴンの不透明度 α_n と、ポリゴン色データ T_n と、ワークデータ X_{n-1} との積を、それぞれ加算した累積値である。

【 0 0 4 2 】

ここで、ポリゴンの色データ T_n は、ポリゴン自体の色やテクスチャカラーに環境光等の光源の色の影響を反映した色データである。この点をより詳細に説明すると、次の通りである。画像データは、ポリゴンの頂点データに対して与えられるポリゴンの自の色 $PCOL$ (以下ポリゴン色データ $PCOL$) と、テクスチャ色データ TX とから得られる色データ $TXCO$ (以下ピクセル色データ $TXCO$) に、光源からの色データ INT の影響を加えて求められる。

【 0 0 4 3 】

光源の色データについては、例えば、黄色い光源からのディフューズド光による影響は、所定の強度を持つ黄色の色データ INT で表すことができる。或いは、周囲の赤い壁紙からのアンビエント光による影響は、所定の強度を持つ赤色の色データ INT で表すことができる。そして、ポリゴンの銀色のメタル生地によるスペキュラー光による影響は、所定の強度を持つ銀色の色データ INT で表すこともできる。従って、それらの光源からの色データとして、ここでは、光源色データ INT と総称する。その結果、式 (7) の色データ T_n は、光源の影響も加えて、

$$T_n = TXCO_n \cdot INT_n$$

と置き換えることができる。即ち、

$$X_n = X_{n-1} \cdot (1 - \alpha_n) \quad (6)$$

$$F B_n = F B_{n-1} + INT_n \cdot TXCO_n \cdot \alpha_n \cdot X_{n-1} \quad (7)$$

$$F B_n = (INT_n \cdot TXCO_n \cdot \alpha_n \cdot X_{n-1}) \quad (8)$$

上記の演算式から明らかな通り、画像データ $F B_n$ は、各ポリゴンについて、1つ手前のポリゴンに対する透明度 $(1 - \alpha_n)$ の積であるワークデータ X_{n-1} と、当該ポリゴンの不透明度 α_n と、光源色データ INT_n と、ピクセル色データ $TXCO_n$ とを求めて、逐次フレームバッファ内に格納される画像データ $F B$ に加算していくことで、求めることができる。しかも、各ポリゴンにおける $INT_n \cdot TXCO_n \cdot \alpha_n \cdot X_{n-1}$ を求める演算は、ワークデータが求まっていれば、他の重なり合うポリゴンの演算結果には依存しない。従って、重なり合うポリゴンに対して、上記のポリゴン毎の $INT_n \cdot TXCO_n \cdot \alpha_n \cdot X_{n-1}$ を求める演算は、順不同で行うことができる。

【 0 0 4 4 】

ゲームやシュミレーションにおいて、リアルタイムに画像処理を行う場合、上記の光源色データ INT_n を求める演算処理は、場合によっては多くの光源に対する演算処理を行う必要があり、非常に工数が高い演算処理になる。同様に、ポリゴンの自の色であるポリゴン色データ $PCOL$ とテクスチャ色データ TX とから得られるピクセル色データ $TXCO$ を求めるテクスチャ処理においても、テクスチャがないポリゴンや、複数のテクスチャデータを有するポリゴンや、更に、細分値 LOD に対応して与えられる複数のテクスチャから、それらの中間の細分値に対応するテクスチャを求めるトライリニアフィルタ処理が必要なポリゴンなどが存在し、それらのポリゴンのテクスチャ処理は、非常に工数がかかる演算処理が必要

な場合がある。

【 0 0 4 5 】

そこで、本発明の好ましい実施の形態例では、光源計算処理やテクスチャ処理を、複数のポリゴンに対して並列に処理を行うようにする。その為に、画像処理装置内に、光源計算回路やテクスチャ処理回路を複数個設ける。そして、複数のポリゴンの演算処理を、空いている処理回路に順次を分配することで処理効率を上げる。これにより、全体の画像処理時間を短くすることができる。

【 0 0 4 6 】

[画像処理装置]

図 3 は、本発明の実施の形態例におけるレンダリング処理プロセッサを有するゲーム装置の構成図である。レンダリング処理プロセッサ 20 は、画像処理装置であり、ゲーム装置やシュミレーション装置内に設けられる。図 3 に示されたゲーム装置には、メイン CPU 10 と、ゲームプログラムが格納されたプログラム ROM と、ゲームプログラム実行時に利用されるワーク RAM 14 と、レンダリング処理プロセッサ 20 とが、内部バス 16 を介して接続される。

10

【 0 0 4 7 】

メイン CPU 10 は、図示しないオペレータからの操作入力に応答して、ゲームプログラムを実行し、必要な画像処理を行う。図 3 の例では、オブジェクトの移動のためのマトリクス演算や透視変換などのジオメトリ処理は、メイン CPU で行われているものとする。もちろん、このジオメトリ処理は、別途設けた専用の処理プロセッサによって行われても良い。

20

【 0 0 4 8 】

レンダリング処理プロセッサ 20 は、メイン CPU 10 により発行されるレンダリング処理を指示する描画コマンドを格納する描画コマンドメモリ 22 と、その描画コマンドを解釈するセットアップブロック 24 とを有する。セットアップブロック 24 は、表示画面内にあるポリゴンデータと、光源データとを、ソータユニット 26 に与える。

【 0 0 4 9 】

ポリゴンデータは、頂点データを有し、各頂点データは、例えば、表示画面内の二次元座標データ (x , y)、奥行き (深さ) を示す Z 値、色データ PCOL、不透明度 (または透明度) p、法線ベクトルデータ (Nx, Ny, Nz)、テクスチャ座標 (Tx, Ty) などを含む。

30

【 0 0 5 0 】

ソータユニット 26 は、供給されるポリゴンデータの視点からの距離である Z 値 (深さの値) に従って、ポリゴンを深さ方向に並び替えて、ユニット内のメモリに記録する。従って、ソータユニット 26 内には、フレーム内の複数のポリゴンについてのデータを一旦保存する。そして、ソータユニット 26 は、表示画面のピクセル毎に、深さ方向の順番で、ポリゴンの不透明度データ p をブレンド処理ユニット 30 に供給する。ブレンド処理ユニットでは、テクスチャの不透明度 t を使用するポリゴンの場合は、ポリゴンの頂点データから得られる不透明度 p と、テクスチャの不透明度 t とを合成して求めたピクセル不透明度 n に従って、ワークデータ Xn を求める処理を行う。上記のテクスチャの不透明度 t をテクスチャメモリ 28 から読み出すために、ピクセルのテクスチャ座標をソータ

40

【 0 0 5 1 】

また、テクスチャの不透明度 t を使用しないポリゴンの場合は、ポリゴンの頂点データから得られる不透明度 p をピクセルの不透明度 n として、ワークデータ Xn を求める。ワークデータを求める処理は、具体的には、上記式 (6) の $X_n = X_{n-1} (1 - n)$ に従う演算を行う処理である。

【 0 0 5 2 】

尚、テクスチャメモリ 28 には、テクスチャの不透明度 t と、テクスチャ色データ T とが格納される。このテクスチャメモリ 28 には、テクスチャデータがプログラム ROM 12 から直接ダウンロードされ、保存される。また、このテクスチャメモリ 28 は、ワーク

50

RAM 14 内の記憶領域に設けることもできる。

【 0 0 5 3 】

ブレンド処理ユニット 30 は、上記の通り、ピクセル毎に且つ深さ方向のワークデータ X_n と不透明度 α_n とを生成し、ピクセルワークメモリユニット 36 に供給する。ワークデータ X_n と不透明度 α_n とは、ピクセルワークメモリユニット 36 内のピクセルワークメモリ内に記録される。

【 0 0 5 4 】

テクスチャ処理ユニット 32 は、ソータユニット 26 から、ピクセル毎に、テクスチャ座標 (Tx, Ty) と、ポリゴン色データ PCOL と、三次元座標 (X, Y, Z) 値などを供給され、テクスチャ座標に従って、テクスチャメモリ 28 からテクスチャ色データ T を読み出し、ポリゴンの地の色であるポリゴン色データ PCOL とテクスチャ色データ T とを合成し、ピクセル色データ TXCOn を求める処理を行う。この処理は、必ずしも表示画面の深さ方向の順に行う必要はない。そして、このテクスチャ処理ユニット内には、処理回路が複数設けられ、複数のピクセルに対する処理が、順次空いている処理回路に分配されて行われる。その結果、ピクセル色データ TXCOn は、順不同で生成される。このピクセル色データ TXCOn もピクセルワークメモリユニット 36 内のメモリに記録される。

【 0 0 5 5 】

光源計算ユニット 34 は、ソータユニット 26 から、光源データと共に、ピクセル毎の三次元座標値 (X, Y, Z) と法線ベクトル (Nx, Ny, Nz) を供給される。そして、光源からの色データ INTn が求められる。この処理も、表示画面の深さ方向の順に行う必要はない。そして、この光源計算ユニット内には、処理回路が複数設けられ、複数のピクセルに対する処理が、順次空いている処理回路に分配されて行われる。その結果、ピクセルの光源色データ INTn は、順不同で生成される。この光源色データ INTn もピクセルワークメモリユニット 36 内のメモリに記録される。

【 0 0 5 6 】

光源色データの生成は、例えばディフーズ光の場合は、光源の輝度ベクトルとピクセルの法線ベクトルの内積値を、光源の色データに乗算して求められる。

【 0 0 5 7 】

ピクセルワークメモリユニット 36 内に、深さ方向で n 番目のピクセルについて、上記のワークデータ X_n 、不透明度 α_n 、ピクセル色データ TXCOn、及び光源色データ INTn が全て格納されると、それらのデータがカラー合成ユニット 38 に供給される。カラー合成ユニット 38 では、それらのデータを乗算して、フレームバッファ 40 内の画像データ FB に加算し、新たに求められた画像データ FB をフレームバッファ 40 内に上書きする。この画像データ FB は、ピクセル色データに対して光源処理され且つブレンド処理された色データである。

【 0 0 5 8 】

全てのピクセルについて、上記のカラー合成処理が行われると、フレームバッファ 40 内には、1 フレームの表示に使用される画像データ FB が格納された状態になる。そこで、このフレームバッファ 40 内の画像データ FB が、表示装置 44 に必要な形態の画像データに変換され、表示装置 44 に供給される。

【 0 0 5 9 】

以上が、レンダリング処理プロセッサ 20 内の処理の流れである。ここでは、ブレンド処理ユニット 30 のみが、深さ方向の順番でピクセルのデータを受け取り、ワークデータ X_n を求める処理を順次行う。また、テクスチャ処理ユニット 32 や光源計算ユニット 34 では、それぞれの演算処理が非常に重たいので、複数の処理回路が設けられ、並列に処理される。しかも、複数の処理回路に対して、空いている処理回路に順次処理が分配されることで、全体の処理スピードを上げることができる。但し、かかる処理をしたことにより、ピクセル色データ TXCOn や光源色データ INTn の生成処理が、深さ方向において前後して完了する。しかし、カラー合成ユニット 38 は、かかるデータ生成の処理が完了したピクセルから順にカラー合成処理を行うことができるので、何ら支障はない。

【0060】

次に、ソーターユニット26、ブレンド処理ユニット30、テクスチャ処理ユニット32、光源計算ユニット34、ピクセルワークメモリユニット36及びカラー合成ユニット38の構成を説明する。

【0061】

図4は、ソーターユニット26の構成図である。ソーターユニット26は、頂点データからなるポリゴンデータPG0~PGjを供給され、各ポリゴン内のピクセルデータを生成するピクセル生成部50と、ピクセル生成部50が生成したピクセルデータPX0,0~PXi,jを、各ピクセル毎に深さ方向の順番で並び替えて、蓄積するピクセルソータ52,54,56とを有する。

10

【0062】

ピクセルソータは、表示画面の画素(ピクセル)の数だけ設けられることが望ましい。しかしながら、画面の画素数は、例えば640×480等と膨大になるので、画面の一部分、例えば8×8=64画素分のピクセルソータを設けるのが、現実的である。その場合は、i=64画素となる。或いは極端な例では、1画素だけのピクセルソータであっても良い。

【0063】

各ピクセルデータPXm,nは、図4に示される通り、例えば座標値(X,Y,Z)値、ポリゴンの地の色であるポリゴン色データPCOL、ポリゴンの頂点データから補間演算により求められる不透明度p、テクスチャ座標(Tx,Ty)、及び法線ベクトル(Nx,Ny,Nz)を有する。頂点データからなるポリゴンデータから、かかるピクセルデータを生成するアルゴリズムは、種々の方法が提案されている。最も一般的には、ラスタスキャン法が利用される。ラスタスキャン法によれば、ポリゴン内のX軸とY軸方向に走査される各ピクセルデータが、ポリゴンの頂点データを補間演算することにより求められる。また、フラクタル法を利用してピクセルデータを求めることもできる。フラクタル法については、本出願人が別途出願した特開平11-144074号公報に開示されている。

20

【0064】

図4のソーターユニットによれば、ピクセル毎のピクセルデータPXm,nが、Z値に基づいて、深さ方向に順番に並べられる。

【0065】

30

図5は、別のソーターユニットの構成図である。このソーターユニット26は、ポリゴンデータPG0~PGjを、Z値に従ってソートし、深さ方向に順番に並び替えて、ポリゴンソーター58に格納する。そして、その並び替えられたポリゴンデータが、より手前側から深さ方向に沿って順に、ピクセル生成部60に供給される。ピクセル生成部60は、ポリゴンデータから、上記したラスタスキャン法やフラクタル法によりポリゴン内のピクセルデータPX0,0~PXi,jを生成する。従って、ピクセルデータは、より手前側から深さ方向にそって順番に、ブレンド処理ユニット30、テクスチャ処理ユニット32、光源計算ユニット34に供給される。

【0066】

図6は、ブレンド処理ユニットの構成図である。ここに示されたブレンド処理ユニット30には、ポリゴンの頂点データから求められた不透明度(透明度)pと、テクスチャ座標とが、深さ方向に沿って順番に供給される。例えば、最も手前側にソートされたピクセルデータPXm,0(m=0~i)の不透明度データpが供給され、そのピクセルに対応するテクスチャの不透明度データtがテクスチャメモリ28から読み出される。透明度処理回路301は、例えば、これらの不透明度データの積算や、その他の所定の処理により合成したピクセルの不透明度nを求める。テクスチャの不透明度データが存在しない場合は、ポリゴンの不透明度データpが、そのまま、ピクセルの不透明度nとなる。

40

【0067】

ブレンド処理ユニット30は、更に、乗算器302と減算器303とXデータワークメモリ304とを有し、Xデータワークメモリ304から1つ手前のピクセルのワークデータ

50

X_{n-1} が読み出されると共に、ピクセルワークメモリユニット36内の対応するピクセルの n 番目の領域に格納される。そして、読み出したワークデータ X_{n-1} と不透明度 n とから、当該処理中の n 番目のピクセルのワークデータ $X_n = X_{n-1} (1 - n)$ が求められる。求められたワークデータ X_n は、ワークデータメモリ304内に更新される。また、不透明度 n は、ピクセルワークメモリユニット36内の対応するピクセルの n 番目の領域に格納される。

【0068】

図17は、ブレンド処理ユニット30の変形例を示す図である。この例は、図6のブレンド処理ユニット30と比較して、ワークデータ X_{n-1} と不透明度 n との乗算を行う乗算器305が追加されている点で異なる。従って、乗算器305で求められた積 $X_{n-1} \cdot n$ が、ピクセルワークメモリユニット36内の対応するピクセルの n 番目の領域に格納される。かかる乗算器305を設けることにより、後述するピクセルワークメモリユニットとカラー合成ユニットの構成を簡略化することができる。

【0069】

このブレンド処理ユニット30では、上記した通り、最も手前にあるピクセルについてのワークデータ X_n が、不透明度 n と1つ前のワークデータ X_{n-1} とから順番に求められる。この演算処理は、乗算器と減算器などにより行われ比較的軽い処理であるので、全ての深さ方向のピクセルに対する演算は、比較的短時間で完了することができる。

【0070】

ブレンド処理ユニット30は、必要であれば、複数ユニット設けて、各ユニットを表示画面(スクリーン)座標に対応付けて並列化することができる。かかる構成の場合でも、各処理ユニットは、対応するピクセルについて手前側から深さ方向に向かって順番にワークデータを求める演算を行う。

【0071】

図7は、光源計算ユニットの構成図である。光源計算ユニット34は、光源計算を行う複数の光源計算回路342, 343, 344を有する。図7の例では、3つの光源計算回路が設けられ、光源計算を並列に処理することができる。光源計算ユニット34は、更に、ソーターユニットからのピクセルの光源データを供給され、空き状態の光源計算回路342, 343, 344に順次光源計算処理を分配する演算指示分配回路341が設けられる。更に、光源計算ユニット34は、光源計算回路が生成した光源色データINTを、生成順にピクセルワークメモリユニット36に供給する演算結果出力回路345が設けられる。

【0072】

図8は、光源計算回路の動作タイミングチャート図である。ここでは、6つのピクセルデータPX1~PX6に対する光源計算回路の動作のタイミングが示される。図8の例では、ピクセルデータPX1に対する光源計算処理に6クロックを要するとする。同様に、ピクセルデータPX2~PX6の処理クロック数が、2, 3, 7, 4, 2と仮定する。そして、ソーターユニット26から、光源計算に必要なピクセルデータとして、光源データと座標値(X, Y, Z)と法線ベクトル(N_x, N_y, N_z)とが順番に供給されたとする。

【0073】

上記の場合、演算指示分配回路341は、空き状態にある光源計算回路に順次その処理を分配する。図8に示される通り、最初にクロック1のサイクルでピクセルデータPX1に対する演算処理が、計算回路1に割り当てられる。ここでは、処理を割り当てるために、ピクセルデータをユニット内のバス346を経由して供給するので、1クロックサイクルを要するとする。そして、ピクセルデータPX1に対する計算は、計算回路1で6クロックサイクルを要して完了する。

【0074】

次に供給されたピクセルデータPX2の処理は、光源計算回路2に割り当てられる。バス346を利用する割り当て処理は、ピクセルデータPX1と重なることができないので、クロック2のサイクルで割り当てが行われ、計算回路2で2サイクルを要して計算が完了する。また、次のピクセルデータPX3は、光源計算回路3に割り当てられる。そして、3クロ

10

20

30

40

50

ックサイクルで処理が完了する。

【 0 0 7 5 】

その後供給されるピクセルデータPX4の処理は、クロック 4 で空き状態にある光源計算回路 2 にアサインされる。そして、7クロックサイクルで処理が完了する。同様に、ピクセルデータPX5は、クロック 6 で空き状態になる光源計算回路 3 に供給される。ピクセルデータPX6は、クロック 7 で空き状態になる光源計算回路 1 に供給される。

【 0 0 7 6 】

図 8 から明らかな通り、6つのピクセルデータに対する光源計算は、クロック 10 で全ての計算が終了する。このピクセルデータに対する光源計算をシリアルに実行した場合は、合計で、 $6+2+3+7+4+2=24$ クロックサイクルを要する。それに比べて、2倍以上早く処理を完了することができる。これは、複数の光源計算回路を設け、複数のピクセルデータに対する光源計算を、並列に且つ空き状態の計算回路に順次処理を分配したことで、実現できたのである。但し、それぞれのピクセルデータに対する計算結果は、順不同で完了する。

10

【 0 0 7 7 】

図 9 は、テクスチャ処理ユニットの構成図である。この構成は、光源計算ユニットと同等であり、複数のテクスチャ処理回路 3 2 2 , 3 2 3 , 3 2 4 が設けられ、光源計算ユニットの場合と同様に、空き状態にある処理回路に順次テクスチャ処理が割り当てられる。演算指示分配回路 3 2 1 は、テクスチャ処理に必要なピクセルデータを、3つの処理回路に分配する。各テクスチャ処理回路は、テクスチャ座標 (Tx,Ty) に従ってテクスチャメモリ 2 8 からテクスチャ色データ T を読み出し、供給されたポリゴンの地肌のポリゴン色データ PCOL と合成して、ピクセル色データ TXCOn を生成する。

20

【 0 0 7 8 】

このテクスチャ処理は、ピクセル毎に処理サイクルにバラツキがあるので、複数の処理回路 3 2 2 , 3 2 3 , 3 2 4 のうち空き状態にある回路が、複数のピクセルに対するテクスチャ処理を行う。それらの動作タイミングに関しては、光源計算において図 8 で説明したアルゴリズムと同じである。従って、ピクセル色データ TXCOn も、順不同で生成され、ピクセルワークメモリユニット 3 6 に格納される。

【 0 0 7 9 】

上記の光源計算ユニット 3 4 やテクスチャ処理ユニット 3 2 には、共通のポリゴンに対応するピクセルデータが連続して供給されることが望ましい。同じポリゴンに対するピクセルデータであれば、共通の処理が含まれる場合があり、それぞれのユニットの処理効率を高めることができるからである。

30

【 0 0 8 0 】

図 10 は、ピクセルワークメモリユニットの構成図である。ピクセルワークメモリユニット 3 6 は、ブレンド処理ユニット 3 0 から供給されるワークデータ Xn、不透明度 n、テクスチャ処理ユニット 3 2 から供給されるポリゴン内のピクセル色データ TXCOn、光源計算ユニット 3 4 から供給される光源色データ INTn とを、ピクセル毎に記録するピクセルワークメモリ 3 6 2 を有する。図 10 には、ピクセルワークメモリ 3 6 2 が、演算済みのピクセルデータ EPX0,0 ~ EPXi,j を格納するマトリクス状のメモリ領域で構成されることが示されている。ピクセルデータ EPXm,n は、図 10 中に示される通り、上記のワークデータ Xn-1、不透明度 n、ピクセル色データ TXCOn、光源色データ INTn と、それらのデータが格納されたことを示す 4 つのフラグ F 1 ~ F 4 を有する。ピクセルデータ EPXm,n の、m は表示画面上のピクセルを区別し、n は表示画面の深さ方向のピクセルを区別する。但し、n は必ずしも深さ方向の順番である必要はない。

40

【 0 0 8 1 】

ピクセルワークメモリユニット 3 6 は、更に、供給される演算済みのピクセルデータを、それぞれのメモリ領域に記録するピクセルワークメモリコントローラ 3 6 1 を有する。ピクセルワークメモリコントローラ 3 6 1 は、演算済みのピクセルデータが格納されると、格納されたデータに対応するフラグ F 1 ~ F 4 を、未格納状態の 0 から、格納状態の 1 に

50

変更する。更に、ピクセルワークメモリコントローラ 361 は、それらの 4 つのフラグが全て 1 になったピクセルデータを、順次、カラー合成ユニット 38 に供給する。

【0082】

ブレンド処理ユニット 30 は、ワークデータ X_n 、不透明度 n を、手前側から深さ方向の順で求め、ピクセルワークメモリユニット 36 に供給する。従って、ピクセルデータの上記 2 つのデータに対応するフラグ F_1 、 F_2 は、最も手前から深さ方向の順番に、即ち $EPX_{m,0}$ 、 $EPX_{m,1}$ 、 $EPX_{m,2}$. . . $EPX_{m,j}$ の順番で、格納状態に変更される。また、ピクセル色データ TXC_{0n} 及び光源色データ INT_n は、それぞれテクスチャ処理ユニット 32 と光源計算ユニット 34 で求められるが、複数の回路のうち、空き状態にある回路を順次利用して求められるので、生成されるタイミングは必ずしも深さ方向の順番にはならない。その結果、4 つのデータが全てピクセルワークメモリ 362 に格納されるタイミングも、ピクセルによって順不同となる。

10

【0083】

そこで、ピクセルワークメモリコントローラ 361 は、各ピクセルデータの 4 つのフラグを監視し、すべて格納済みになったピクセルデータ EPX から、カラー合成ユニット 38 に供給される。

【0084】

図 11 は、カラー合成ユニットの構成図である。カラー合成器 38 は、乗算器 381 と加算器 382 とを有する。乗算器 381 は、ピクセルワークメモリコントローラ 361 から供給されるピクセルデータ $EPX_{m,n}$ のワークデータ X_{n-1} 、不透明度 n 、ピクセル色データ TXC_{0n} 、光源色データ INT_n を乗算する。加算器 382 は、フレームバッファ 40 内の処理中のピクセルに対応する画像データ FB と、上記乗算結果とを加算し、新たに生成された画像データ FB を、フレームバッファ 40 の同じピクセル領域に格納する。この演算処理は、必ずしも深さ方向に順番に行われるわけではなく、順不同で行われる。カラー合成ユニット 38 は、同じピクセルにおいて、深さ方向の全てのピクセルデータ $EPX_{m,n}$ について上記演算を行うことにより、表示用画像データ FB を、フレームバッファ 40 内に格納することができる。

20

【0085】

図 16 に示したブレンド処理ユニットの変形例が利用される場合は、カラー合成ユニット 38 の乗算器 381 には、ワークデータ X_n と不透明度 n との乗算値が供給される。従って、その分、乗算器 381 の構成を簡略化することができる。

30

【0086】

図 3 に戻り、全てのピクセルに対する最終的な画像データ FB が、フレームバッファ 40 に格納されると、その画像データ FB が表示制御部 42 を経由して、表示装置 44 に供給され、表示される。

【0087】

図 12 は、上記したレンダリング処理プロセッサの処理フローチャート図である。セットアップブロック 24 が、描画コマンドを解読して、ポリゴンデータをソーターユニット 26 に供給する (S10)。ソーターユニット 26 は、ポリゴンをピクセルに分解して、ピクセルソータ 52, 54, 56 に供給する (S12)。ピクセルソータは、視点からの距離 (深さ、Z 値) に応じて、ピクセルデータを並び替える (S14)。その後、ソーターユニット 26 は、手前のピクセルから順に、ピクセルデータを、ブレンド処理ユニット 30、テクスチャ処理ユニット 32、光源計算ユニット 34 に供給する (S16)。

40

【0088】

各処理ユニットでは、上記した通り、ワークデータ X_n の生成 (S18)、ピクセル色データ TXC_{0n} の生成 (S20)、及び光源色データ INT_n の生成 (S22) を行う。ブレンド処理ユニット 30 は、手前のピクセルから順番にワークデータを求める必要があるが、テクスチャ処理ユニット 32 や光源計算ユニット 34 は、複数の処理回路の空き状態の回路を順次利用して、順不同でデータを生成する。ブレンド処理ユニット 30 は、手前のピクセルから順番にワークデータを求めるが、深さ方向において、不透明度 n が 1 である不

50

透明ピクセルまでワークデータを求めたら、それ以降のワークデータ X_n は 0 であるので、ブレンド処理は終了する。従って、ブレンド処理部は、必ずしも深さ方向の全てのピクセルに対して処理を繰り返すとは限らない。

【 0 0 8 9 】

そして、各ピクセルのデータの生成が完了すると、カラー合成ユニットは、演算済みのピクセルデータから、各ピクセルの光源処理とブレンド処理された色データ $INT_n \cdot TXCO_n \cdot X_n \cdot X_{n-1}$ を演算で求め、フレームバッファ内の画像データに加算して、再度格納する (S 2 4)。

【 0 0 9 0 】

上記の工程 S 1 6 ~ S 2 4 が、各ピクセルの深さ方向の全てのピクセルデータについて終了し、且つ全てのピクセルについて終了するまで、繰り返される。そして、全てのピクセルについて処理が終了すると、フレームバッファ内に格納された画像データにより、表示が行われる (S 2 6)。工程 S 1 0 ~ S 2 6 が、フレーム毎に繰り返される。

【 0 0 9 1 】

[汎用的なブレンド処理への適用]

上記の実施の形態例では、半透明ポリゴンの色とその奥側に位置するポリゴンの色とを不透明度 で混合ブレンディング処理する場合に、手前側のポリゴンから順番に処理する例を説明した。しかしながら、一般的なコンピュータを利用したグラフィックライブラリには、例えばオープン G L (Graphic Library) で規格化されているように、多様なブレンディング処理ができるようになっている。その一般化されたブレンディング処理は、以前処理して求めたフレームバッファ内の色データ $F B_{m-1}$ と今回処理中のピクセルの色データ PIX_m とを、それぞれソース混合計数 $S R_m$ とディスティネーション混合計数 $D E_m$ とでブレンディングすることにある。即ち、演算式で表すと、今回の処理で求められてフレームバッファ内に格納される色データ $F B_m$ は、

$$F B_m = PIX_m * D E_m + F B_{m-1} * S R_m \quad (8)$$

となる。但し、 m は整数であり、小さいほど先に処理される (* は積を表す。以下同様)。

【 0 0 9 2 】

上記のブレンディング演算式において、ソース混合計数 $S R_m$ とディスティネーション混合計数 $D E_m$ とは、次の 1 1 種類のうちから適宜選択される。

【 0 0 9 3 】

0、 $F B_{n-1}$ 、 PIX_n 、 PIX_n^* 、 $F B_{n-1}^*$ 、1、 $1 - F B_{n-1}$ 、 $1 - PIX_n$ 、 $1 - PIX_n^*$ 、 $1 - F B_{n-1}^*$ 、 $(F B_{1-n}^*$ 、 $1 - PIX_n^*$) のうちいずれか小さい方上記の式 (8) から理解される通り、一般的なブレンディング処理は、描画前のフレームバッファの色データを必要とし、また演算結果も描画順序に依存している。従って、一般的なコンピュータ・グラフィックにおいては、ブレンディング処理の必要なポリゴンは奥側より順に描画するのが通例である。

【 0 0 9 4 】

図 1 3 は、半透明ポリゴンの混合ブレンディング処理を示す図である。手前側の半透明ポリゴン $P G_m$ と、奥側のポリゴン $P G_{m-1}$ が一部重なっていて、手前側の半透明ポリゴン $P G_m$ の不透明度が の場合である。この場合は、従来例で説明した通り、奥側のポリゴン $P G_{m-1}$ の色データ $F B_{m-1}$ が先に求められ、それに手前側の半透明ポリゴン $P G_m$ の色データが、不透明度 m と透明度 ($1 - m$) によって、混合ブレンディング処理される。即ち、その演算式は、図示される通り、

$$F B_m = PIX_m * m + F B_{m-1} * (1 - m) \quad (9)$$

即ち、上記式 (9) は、一般式 (8) において、ディスティネーション混合係数 $D E_m = m$ で、ソース混合係数 $S R_m = (1 - m)$ にした場合に対応する。

【 0 0 9 5 】

図 1 4 は、補助フィルタ処理の具体例を示す図である。汎用的なグラフィックライブラリには、ブレンド処理として補助フィルタ処理がある。この処理は、図 1 3 の半透明ポリゴ

ンの混合ブレンディング処理と異なり、奥側のポリゴン P G_{m-1}の色データがその手前側のポリゴン P G_mの混合係数 α_m にかかわらず、そのまま減衰することなく残される処理である。

【 0 0 9 6 】

例えば、手前側のポリゴン P G_mが炎に対応するポリゴンであり、奥側のポリゴン P G_{m-1}が所定の不透明ポリゴンの場合を想定する。この場合は、視点からは炎のポリゴン P G_mを通して不透明ポリゴン P G_{m-1}を見ていることになる。その場合は、炎ポリゴン P G_mの色は加算されるが、奥側のポリゴン P G_{m-1}の色データが減衰することはない。

【 0 0 9 7 】

その場合は、炎ポリゴン P G_mの色データ PIX_mに対してディスティネーション混合係数 D E_m = α_m 、ソース混合係数 S R_m = 1 にすることにより、上記の一般式 (8) を利用することができる。その結果、補助フィルタ処理の演算式は、図示される通り、

$$F B_m = PIX_m * \alpha_m + F B_{m-1} * 1 \quad (1 0)$$

となる。即ち、前回の処理で求めたポリゴン P G_{m-1}の色データ F B_{m-1}をそのまま残し、今回処理中のポリゴン P G_mの色データ PIX_mに混合係数 α_m 分だけ加算する処理である。

【 0 0 9 8 】

この補助フィルタ処理は、上記の炎ポリゴンの例以外に、例えば赤い光源からの光がポリゴン P G_{m-1}に照射されている場合も、同様の処理が適用される。

【 0 0 9 9 】

図 1 5 は、更に、カラーフィルタ処理を示す図である。カラーフィルタ処理では、例えば、背景にあるポリゴン P G_{m-1}の特定の色成分だけを抽出するために、手前側にそれ自身は色成分を持たないカラーフィルタポリゴン P G_mを配置させる。そして、このブレンディング処理演算では、特定の色成分だけを抽出させるために、背景のポリゴン P G_{m-1}の色データ F B_{m-1}にカラーフィルタポリゴン P G_mの色抽出データ PIX_mを乗算する。

【 0 1 0 0 】

即ち、その演算式は、図示される通り、

$$F B_m = PIX_m * 0 + F B_{m-1} * PIX_m \quad (1 1)$$

となる。この演算式は、一般式 (8) に対して、カラーフィルタポリゴン P G_mの色抽出データ PIX_mに対してディスティネーション混合係数 D E_m = 0、前回求めた色データ F B_{m-1}に対してソース混合係数 S R_m = PIX_mにすることにより得られる。

【 0 1 0 1 】

以上の通り、半透明ポリゴンの混合ブレンディング処理以外に、補色フィルタ処理やカラーフィルタ処理などの様々なブレンディング処理は、一般式 (8) に、適宜選択したディスティネーション混合係数 D E_mとソース混合係数 S R_mと適用して演算処理することができる。通常のコンピュータ・グラフィック処理装置は、上記の一般式 (8) に適用できるように構成され、その結果、ディスティネーション混合係数 D E_mとソース混合係数 S R_mとを適宜選択して行うことにより、様々なブレンディング処理が可能になる。

【 0 1 0 2 】

そこで、上記の一般的なブレンディング処理に対して、視点側から (手前側) ポリゴンを順番に処理することができる方法、処理装置について、以下に説明する。

【 0 1 0 3 】

図 1 の場合と同様に、4枚のポリゴンが手前側から重なっていると仮定する。その場合、一般式 (8) に対して、 $m = 1 \sim 4$ により、次の4つの演算式が導かれる。

【 0 1 0 4 】

$$F B_1 = PIX_1 * D E_1 + F B_0 * S R_1 \quad (2 1)$$

$$F B_2 = PIX_2 * D E_2 + F B_1 * S R_2 \quad (2 2)$$

$$F B_3 = PIX_3 * D E_3 + F B_2 * S R_3 \quad (2 3)$$

$$F B_4 = PIX_4 * D E_4 + F B_3 * S R_4 \quad (2 4)$$

ここで、留意すべき点は、上記ブレンディング処理は、奥側のポリゴンを先に処理してそ

の色データがフレームバッファに記録されているので、 m が大きいほど手前のポリゴンを意味し、 m が小さいほど奥側のポリゴンを意味することである。

【 0 1 0 5 】

そこで、上記の(21)～(24)について、式(5)(6)(7)で行ったように、最も手前のポリゴンの色データを $F B_4$ から D_1 に、最も奥側のポリゴンの色データを $F B_1$ から D_4 にそれぞれ置き換え、半透明ポリゴンの場合と同様に、処理中のピクセルの色データPIXをピクセル色データTXCO $_n$ と光源色データINT $_n$ の乗算値TXCO $_n$ * INT $_n$ に置き換えると、次の通りである。つまり、 $D_4 = F B_1 \sim D_1 = F B_4$ とし、 $PIX_4 = TXCO_1 * INT_1 \sim PIX_1 = TXCO_4 * INT_4$ すると、

$$D_4 = TXCO_4 * INT_4 * DE_4 + D_5 * SR_4 \quad (31)$$

$$D_3 = TXCO_3 * INT_3 * DE_3 + D_4 * SR_3 \quad (32)$$

$$D_2 = TXCO_2 * INT_2 * DE_2 + D_3 * SR_2 \quad (33)$$

$$D_1 = TXCO_1 * INT_1 * DE_1 + D_2 * SR_1 \quad (34)$$

そこで、ワークデータX $_n$ (但しX $_0 = 1$)を、

$$X_n = SR_n * X_{n-1} = SR_n * SR_{n-1} * SR_{n-2} * \dots * SR_1 \quad (35)$$

とすると、式(34)は、次の通り展開される。

【 0 1 0 6 】

$$D_1 = TXCO_1 * INT_1 * DE_1 * X_0 + TXCO_2 * INT_2 * DE_2 * X_1 + TXCO_3 * INT_3 * DE_3 * X_2 + TXCO_4 * INT_4 * DE_4 * X_3$$

であるから、各TXCO $_n$ * INT $_n$ * DE $_n$ * X $_{n-1}$ に注目すると、

$$\begin{aligned} D_n &= D_{n-1} + TXCO_n * INT_n * DE_n * X_{n-1} \\ &= \Sigma (TXCO_n * INT_n * DE_n * X_{n-1}) \end{aligned} \quad (36)$$

と一般化することができる。この場合上記の置き換えにより、 n が小さいほど手前のポリゴンを意味しているの、より手前のポリゴンから順番にブレンディング処理を行うことができることを意味している。即ち、より手前に位置するポリゴンの処理結果D $_{n-1}$ を利用して、次の奥側に位置するポリゴンのデータD $_n$ を求めることができるのである。そして、上記の式(35)と(36)において、SR $_n = (1 - \alpha_n)$ 、DE $_n = \alpha_n$ と置き換え、更にポリゴンの色データD $_n$ をフレームバッファのデータF B $_n$ に置き換えると、半透明ポリゴンの処理で示した式(6)(7)と同じになることが理解される。

【 0 1 0 7 】

上記の式(35)によれば、ワークデータX $_n$ は、ソース混合係数SR $_n$ の累積値である。また、上記の式(36)によれば、視点から n 枚目のポリゴン进行处理した結果の色データD $_n$ (F B $_n$)は、 n 枚目のポリゴンの色データTXCO $_n$ を、光源色データINT $_n$ 、ディスティネーション混合係数DE $_n$ と、手前に位置する $n - 1$ 枚目のポリゴンのソース混合係数SR $_1 \sim SR_{n-1}$ の累積乗算値(ワークデータ)X $_{n-1}$ とに応じて処理し、全ての画像データD $_n$ を加算するレンダリング処理により求められる。しかも、ワークデータX $_n$ が所定の最低値よりも小さくなると、その後のポリゴンの処理を省略しても表示される色データに大きな影響を与えないことは、半透明ポリゴンのブレンド処理と同じである。

【 0 1 0 8 】

図16は、上記の汎用レンダリング処理を行う場合の、レンダリング処理プロセッサ内に設けられるブレンド処理ユニットの構成図である。上記の通り、汎用レンダリング処理に適用する場合は、ブレンド処理ユニット30が、ソース係数SR $_n$ を乗算したワークデータX $_n$ を、視点側から奥行き方向に沿って順に求める必要がある。その場合、ソース係数SR $_n$ は、ブレンディング処理の種類によって、不透明度 α_n を使用する場合や、テクスチャ色データT $_n$ を使用する場合に、それぞれ区別して処理しなければならない。

【 0 1 0 9 】

図16に示したブレンド処理ユニット30は、ソース、ディスティネーション係数生成回路301と、乗算器302と、Xデータワークメモリ304を有する。係数生成回路301は、テクスチャメモリ28A、28Bから、テクスチャの不透明度 α とテクスチャの

10

20

30

40

50

色データTを読み出すことができる。また、係数生成回路301に、ソーターユニットから、ピクセルの不透明度pとテクスチャ座標(Tx,Ty)が供給される。

【0110】

ブレンディング処理の種類に応じて、係数生成回路301は、ソース混合係数SRnとして、 $1 - \alpha_n$ 、 1 、 T_n をそれぞれ生成し、乗算器302に供給する。乗算器302は、そのソース混合係数SRnをXデータワークメモリ304から読み出されたワークデータ X_{n-1} と乗算する。また、係数生成回路301は、ディステーション混合係数Dnも同時に出力し、ピクセルワークメモリ36に供給し、格納する。更に、Xデータワークメモリ304から読み出されたワークデータ X_{n-1} も、ピクセルワークメモリ36に供給され、格納される。上記の3つのブレンディング処理では、ディステーション混合係数Dnは、

10

【0111】

即ち、ブレンド処理ユニット30では、次の様に混合係数を選択することで、上記した3つの処理を行うことができる。

【0112】

半透明ポリゴンの混合ブレンディング処理： $D_n = \alpha_n$ 、 $SR_n = 1 - \alpha_n$

補助フィルタ処理： $D_n = \alpha_n$ 、 $SR_n = 1$

カラーフィルタ処理： $D_n = 0$ 、 $SR_n = T_n$

上記の例以外にも、種々の混合係数を選択することができる。例えば、 0 、 $T_n * \alpha_n$ 、 $1 - T_n$ 、 $1 - T_n * \alpha_n$ 等を適宜選択することができる。但し、データFBn-1、即ち、後で処理して求められる色データDn+1を含む混合係数を選択することはできない。

20

【0113】

図18は、ブレンド処理ユニットの変形例を示す図である。図16に比較して、図18の変形例は、更にディステーション係数DnとXワークデータ X_{n-1} とを乗算する乗算器305を有する。そして、この乗算器305の乗算値 $X_{n-1} * D_n$ が、ピクセルワークメモリユニット36に格納される。かかるブレンド処理ユニットを利用することで、カラー合成ユニットの乗算器381の構成を簡略化することができる。

【0114】

テクスチャ処理ユニット32、光源計算ユニット34の構成は、汎用ブレンディング処理の場合でも前述と同じである。即ち、比較的大きな処理工数を必要とするピクセル色データTXCOnの生成(テクスチャ処理)や、光源色データを求める光源計算は、並列回路により順不同で求め、逐次ピクセルワークメモリユニット36に格納する。そして、ピクセルデータの処理が完了したピクセルから、カラー合成ユニット38により合成処理が行われる。

30

【0115】

カラー合成ユニット38では、一つ手前のワークデータ X_{n-1} 、ディステーション混合係数Dn、ピクセルの色データTXCOn及び光源色データINTnが、乗算器381で乗算され、フレームバッファ40内の画像データFBに加算され、フレームバッファ40に書き込みされる。

40

【0116】

図16に示したブレンド処理ユニット30は、カラーフィルタ処理の場合に、ソース混合係数SRnとして、テクスチャの色データTをそのまま利用した。しかしながら、より厳密にカラーフィルタの色データを求めるためには、テクスチャ処理ユニット32により生成されるピクセル色データTXCOnと、光源計算ユニット34により生成される光源色データINTnとを乗算して光源処理する必要がある。従って、その光源処理されたピクセル色データをブレンド処理ユニット30に供給してソース混合係数として利用する必要がある。その為に、カラーフィルタ処理において、ブレンド処理ユニット30によるワークデータの生成処理が、しばらく停止する必要がある。そして、それぞれ生成されてピクセルワークメモリに記録されたピクセルの色データTXCOnと光源色データINTnを、ブレンディング処

50

理ユニット 30 に供給しながら、ワークデータ Xn を生成する。

【0117】

上記の処理は、カラーフィルタ処理の場合に限定されるので、全体のレンダリング処理に与える影響は、それほど大きくない。

【0118】

以上の実施の形態例において、レンダリング処理プロセッサ 20 内のブレンド処理ユニット 30、テクスチャ処理ユニット 32、及び光源計算ユニット 34 は、それぞれマイクロコンピュータによって実現されても良い。レンダリング処理プロセッサ 20 が、1 チップのプロセッサで実現される場合は、それらの処理ユニットとしてマイクロコンピュータの構成を有するマクロ回路が埋め込まれる。その場合、テクスチャ処理ユニット 32、光源計算ユニット 34 は、複数のマイクロコンピュータを有し、それぞれの処理をプログラムを実行することにより行う。複数ユニット設けられたマイクロコンピュータは、空き状態になりしだい、テクスチャ処理と光源計算が分配されるので、各ピクセルの処理時間が異なっても、全体の処理時間を短くすることができる。

10

【0119】

以上、本発明の保護範囲は、上記の実施の形態例に限定されるものではなく、特許請求の範囲に記載された発明とその均等物にまで及ぶものである。

【0120】

【発明の効果】

以上説明した通り、本発明によれば、半透明ポリゴンの画像処理において、表示画面の視点側のポリゴンから奥行き方向に沿って順に処理するアルゴリズムを採用することで、奥行き方向について順不同でカラー合成処理を行うことができる。その結果、より高速に処理可能なレンダリングプロセッサを実現することができる。

20

【0121】

また、本発明によれば、一般的なブレンディング処理を行うことができる汎用レンダリングプロセッサにおいても、同様により高速に処理する構成を実現することができる。

【図面の簡単な説明】

【図 1】半透明ポリゴンの重なり例を説明する図である。

【図 2】本発明及び従来例のレンダリング処理を、図 1 の例に適用した時の演算式を示す図表である。

30

【図 3】本発明の実施の形態例におけるレンダリング処理プロセッサを有するゲーム装置の構成図である。

【図 4】ソーターユニットの構成図である。

【図 5】別のソーターユニットの構成図である。

【図 6】ブレンド処理ユニットの構成図である。

【図 7】光源計算ユニットの構成図である。

【図 8】光源計算回路の動作タイミングチャート図である。

【図 9】テクスチャ処理ユニットの構成図である。

【図 10】ピクセルワークメモリユニットの構成図である。

【図 11】カラー合成ユニットの構成図である。

40

【図 12】レンダリングプロセッサの処理フローチャート図である。

【図 13】半透明ポリゴンの混合ブレンディング処理を示す図である。

【図 14】補助フィルタ処理を示す図である。

【図 15】カラーフィルタ処理を示す図である。

【図 16】汎用レンダリング処理を行う場合のブレンド処理ユニットの構成図である。

【図 17】ブレンド処理ユニットの変形例の構成図である。

【図 18】汎用レンダリング処理を行う場合のブレンド処理ユニットの変形例の構成図である。

【符号の説明】

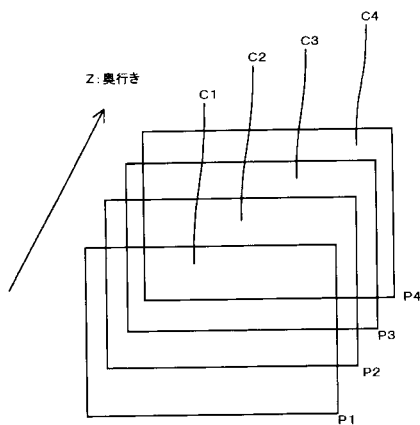
34 レンダラ、レンダリング処理部

50

3 4 2 Xバッファ、ワークデータバッファ
 3 4 3 Dバッファ、画像データバッファ
 T_n 色データ、テクスチャデータ
 D_n 画像データ
 n 不透明度データ
 P 1 ~ P 4 ポリゴン

【図 1】

半透明ポリゴンの重なり



【図 2】

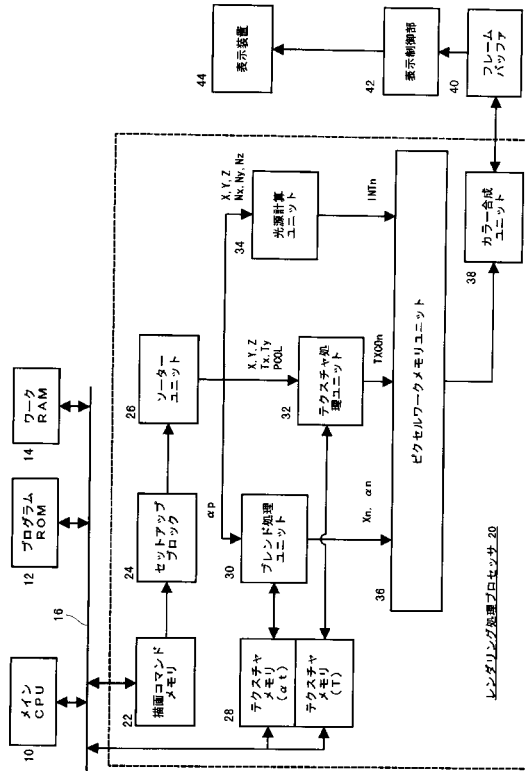
(1) 本発明

	n	X_n	D_n
初期値	0	1	0
	1	$X_0(1-\alpha_1)$	$FB_0+T_1\alpha_1X_0$
	2	$X_1(1-\alpha_2)$	$FB_1+T_2\alpha_2X_1$
	3	$X_2(1-\alpha_3)$	$FB_2+T_3\alpha_3X_2$
結果	4	$X_3(1-\alpha_4)$	$FB_3+T_4\alpha_4X_3$

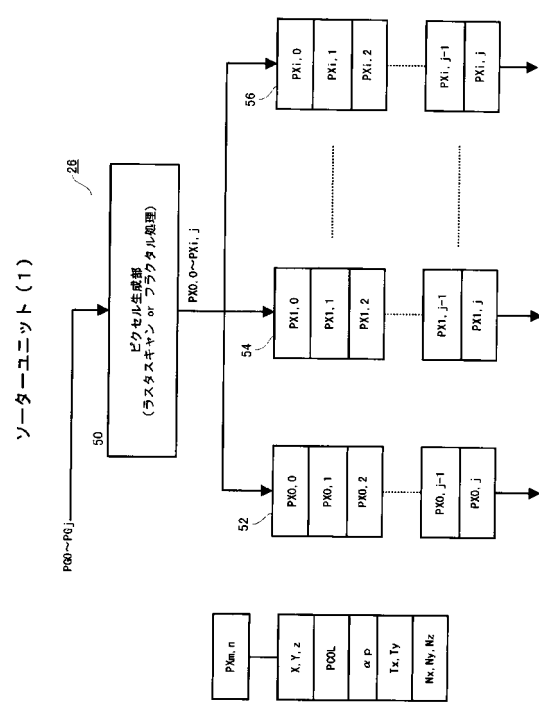
(2) 従来例

	n	C_n
初期値	-	C_5
	4	$\alpha_4T_4+(1-\alpha_4)C_5$
	3	$\alpha_3T_3+(1-\alpha_3)C_4$
	2	$\alpha_2T_2+(1-\alpha_2)C_3$
結果	1	$\alpha_1T_1+(1-\alpha_1)C_2$

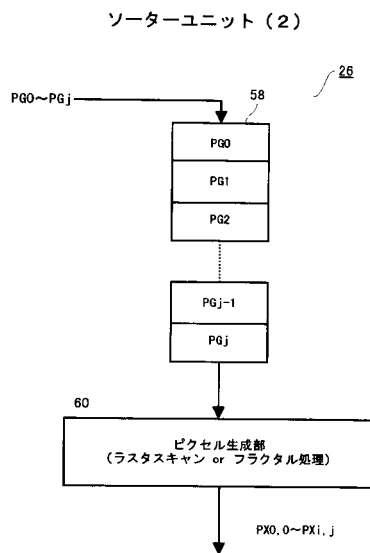
【 図 3 】



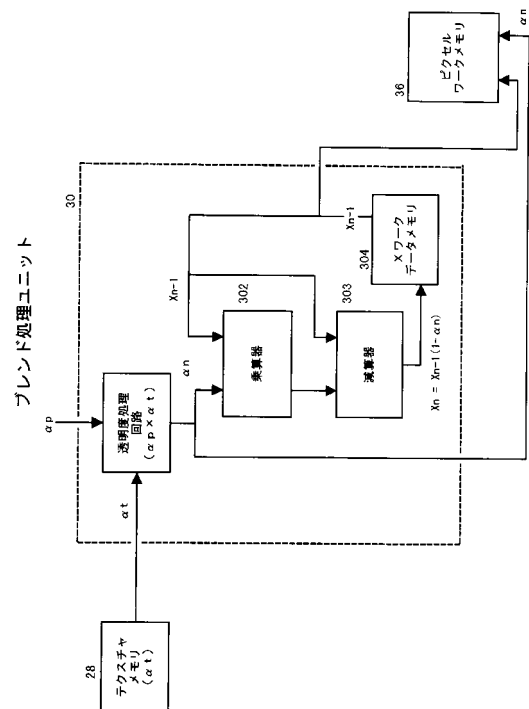
【 図 4 】



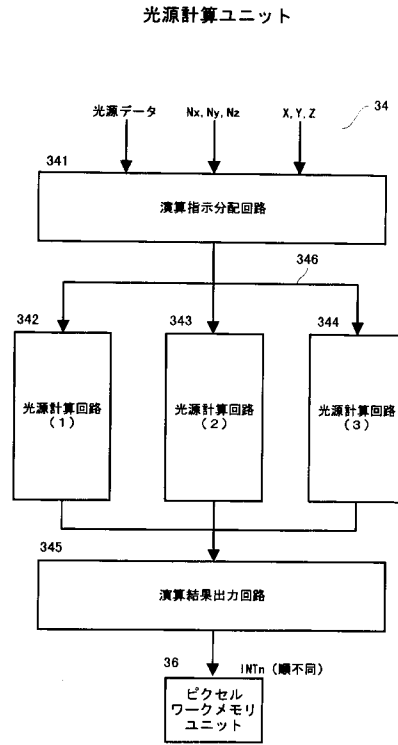
【 図 5 】



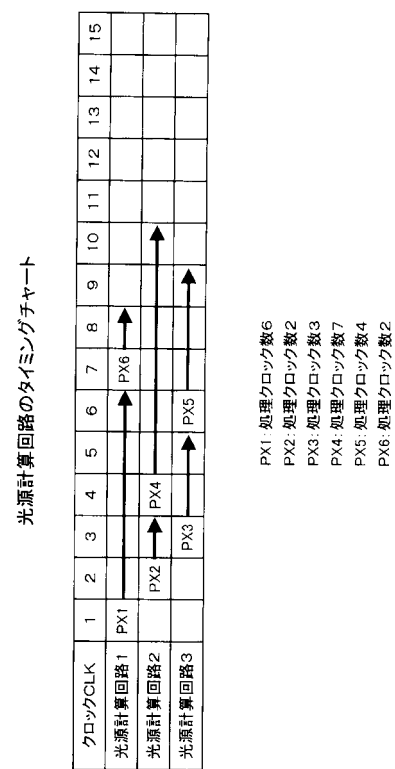
【 図 6 】



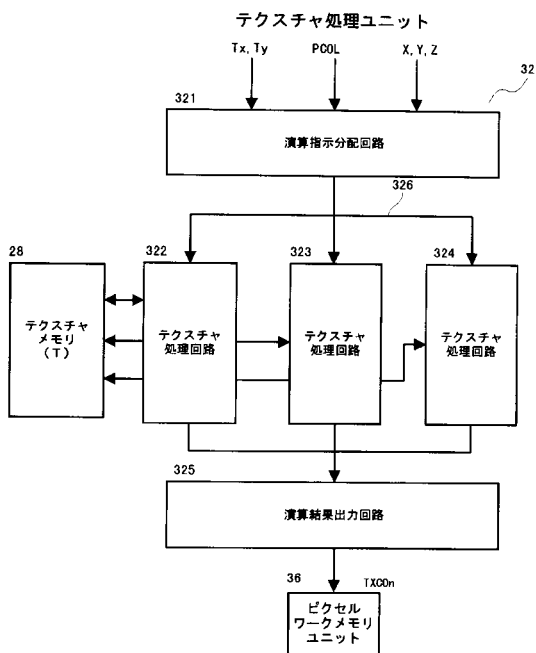
【図 7】



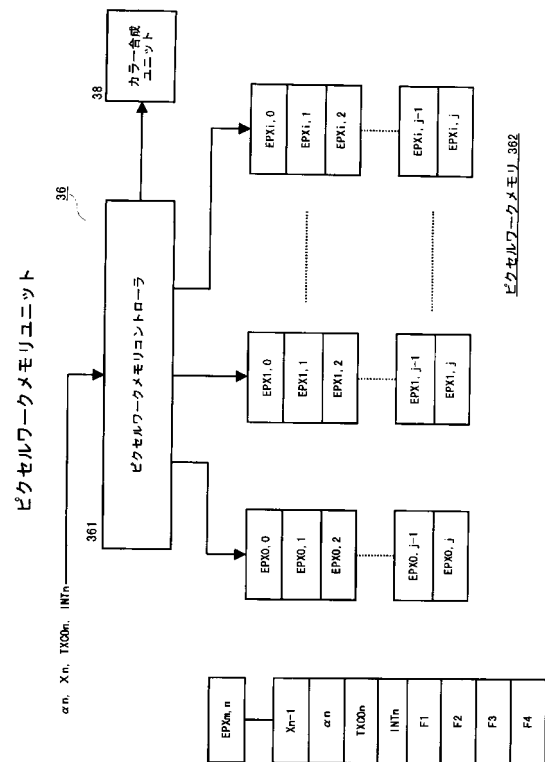
【図 8】



【図 9】

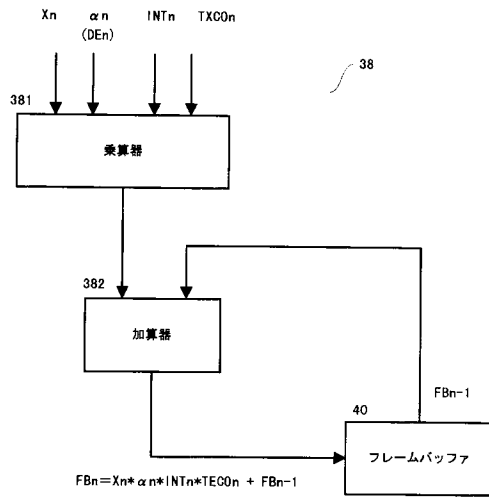


【図 10】

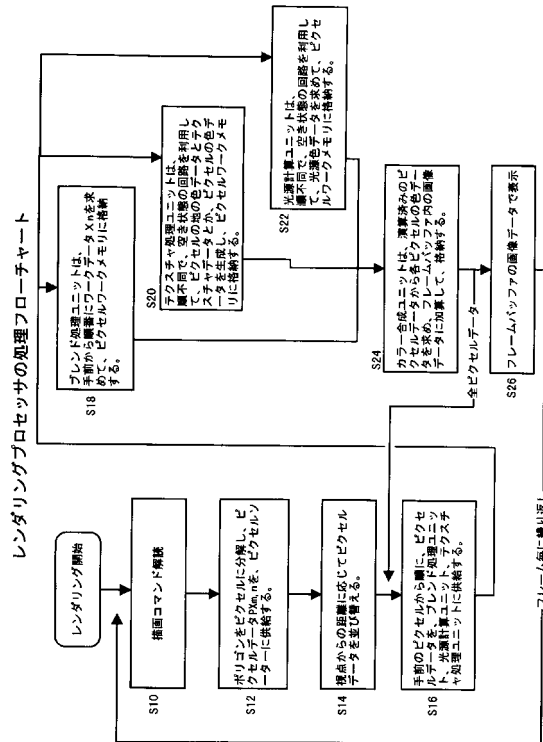


【 ㄨ 1 1 】

カラー合成ユニット

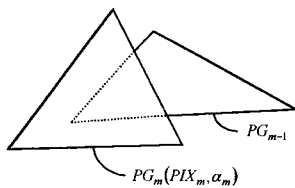


【 図 1 2 】



【 図 1 3 】

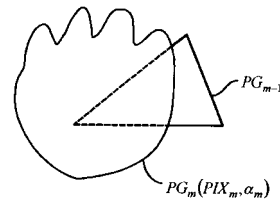
半透明ポリゴンのブレンディング処理



$$FB_m = \underset{\substack{\parallel \\ (DE_m)}}{PLX_m} * \alpha_m + FB_{m-1} * \underset{\substack{\parallel \\ (SR_m)}}{(1 - \alpha_m)}$$

【 図 1 4 】

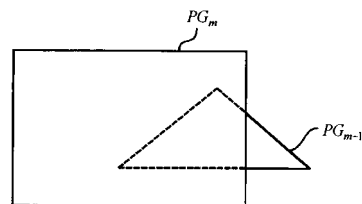
補助フィルタ処理の具体例



$$FB_m = PIX_m * \underset{\substack{\parallel \\ (DE_m)}}{\alpha_m} + FB_{m-1} * \underset{\substack{\parallel \\ (SR_m)}}{1}$$

【 図 1 5 】

カラーフィルタ処理の具体例

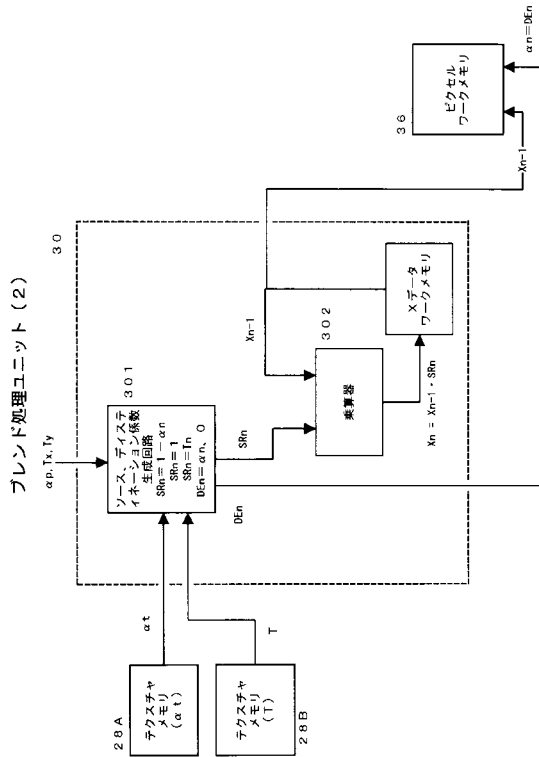


$$FB_m = PIX_m * 0 + FB_{m-1} * PIX_m$$

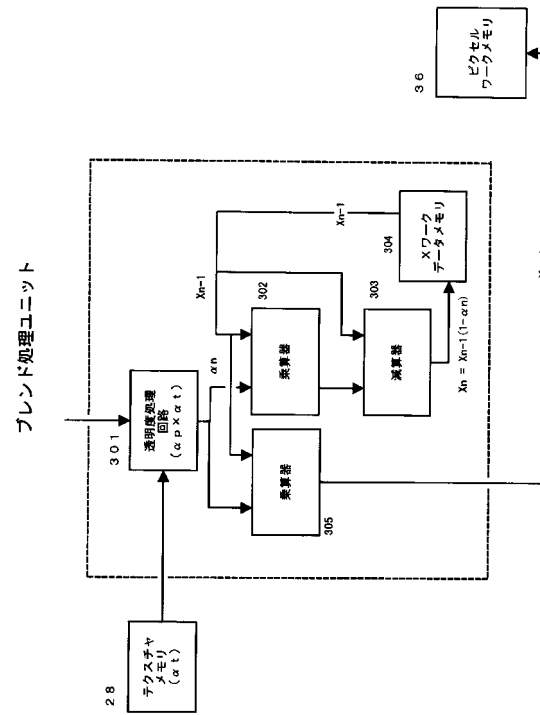
$$\quad \quad \quad \parallel \quad \quad \quad \parallel$$

$$\quad \quad \quad (DE_m) \quad \quad \quad (SR_m)$$

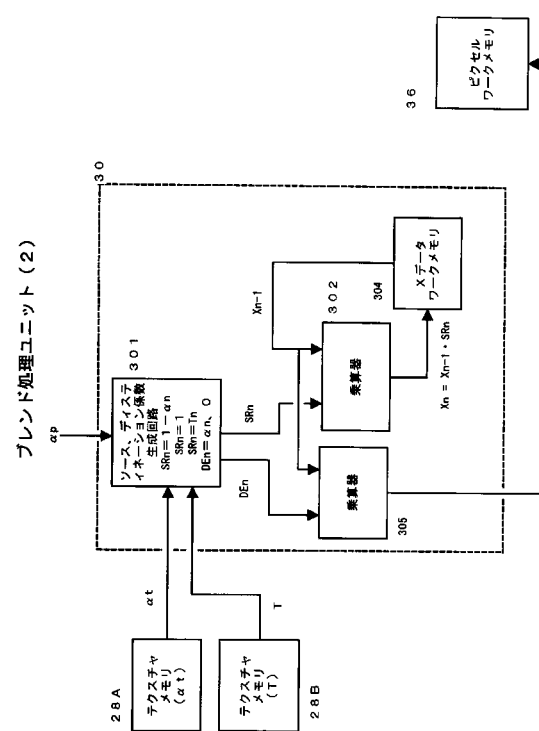
【図 16】



【図 17】



【図 18】



フロントページの続き

(56)参考文献 特開平 1 1 - 0 6 6 3 4 0 (J P , A)
特開平 0 9 - 3 3 0 4 2 2 (J P , A)
特開平 0 8 - 2 2 1 5 9 3 (J P , A)
特開平 0 6 - 2 7 4 6 4 7 (J P , A)
特開平 0 4 - 5 0 2 3 6 8 (J P , A)
国際公開第 9 9 / 0 5 4 8 4 7 (WO , A 1)

(58)調査した分野(Int.Cl. , D B 名)

G06T 15/00