

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2005-77522

(P2005-77522A)

(43) 公開日 平成17年3月24日(2005.3.24)

(51) Int.CI.⁷

G09G 5/38
A63F 13/00
G06T 11/00
G09G 5/00
G09G 5/377

F 1

G 09 G 5/38
A 63 F 13/00
A 63 F 13/00
G 06 T 11/00
G 09 G 5/00
G 09 G 5/377

審査請求 有 請求項の数 4 O L (全 14 頁)

テーマコード (参考)

2 C 001
5 B 080
5 C 082

最終頁に続く

(21) 出願番号

特願2003-305273 (P2003-305273)

(22) 出願日

平成15年8月28日 (2003.8.28)

(71) 出願人

ヤマハ株式会社

静岡県浜松市中沢町10番1号

(74) 代理人

100064908

弁理士 志賀 正武

(74) 代理人

100089037

弁理士 渡邊 隆

(72) 発明者

本目 光弘

静岡県浜松市中沢町10番1号 ヤマハ株

式会社内

F ターム (参考) 2C001 BA02 BA04 BA05 BA06 BC06
CB01 CB03 CB08 CC01
5B080 CA01 FA02 FA03 FA08 FA17
5C082 AA06 BA12 BA43 BB15 BB22
CA12 CA56 CA81 DA53 DA63
DA86 DA89 MM05

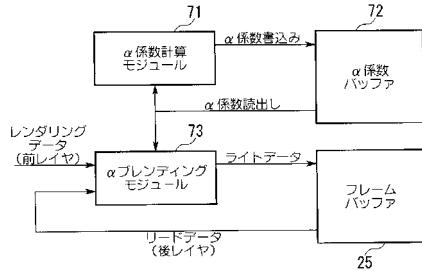
(54) 【発明の名称】画像処理装置および画像処理方法

(57) 【要約】

【課題】 表示優先順位の高いスプライトから描画することができ、かつ複数面についてのカラー演算を行うこともできる画像処理装置および画像処理方法を提供する。

【解決手段】 表示画像の各画素に対応させて 係数を格納する 係数バッファ 72 と、ブレンディング実行時に下位面に対する 係数を計算しその 係数を 係数バッファ 72 に格納させる 係数計算モジュール 71 と、下位面描画時に描画データ、描画済みデータ及び 係数バッファに格納されているデータを用いて ブレンディングを行う ブレンディングモジュール 73 とを有することを特徴とする。

【選択図】 図6



【特許請求の範囲】**【請求項 1】**

表示手段に出力される表示画像データをビットマップ形式で保持するフレームバッファと、

前記表示画像データの各画素に対応させて設定されている係数データであると共に、各画素位置の透過度情報を示す係数データである 係数を格納する 係数バッファと、

前記 係数を使って 2 つの画像を半透明合成する処理である ブレンディングの実行時に、該 2 つの画像における表示優先度が低い方の画像である下位面に対する 係数を計算し、該 係数を前記 係数バッファに格納させる 係数計算モジュールと、

前記 2 つの画像における表示優先度が高い方の画像である上位面又は前記下位面を前記フレームバッファに描画するときに、描画データ、描画済みデータ及び前記 係数バッファに格納されているデータを用いて ブレンディングを行うと共に、該上位面のスプライトを前記フレームバッファに描画した後に、該下位面のスプライトを該フレームバッファに描画する ブレンディングモジュールとを有することを特徴とする画像処理装置。 10

【請求項 2】

表示手段に出力される表示画像データをビットマップ形式で保持するフレームバッファと、

前記表示画像データの各画素に対応させて設定されている係数データであると共に、2 つの画像を重ねてなる合成画像の各画素データを算出するピクセル演算に用いる係数データを格納する 係数バッファと、

前記ピクセル演算を算出するときに、前記 2 つの画像における表示優先度が低い方の画像である下位面に対する前記係数データを計算し、該係数データを前記係数バッファに格納させる 係数計算モジュールと、

前記 2 つの画像における表示優先度が高い方の画像である上位面又は前記下位面を前記フレームバッファに描画するときに、描画データ、描画済みデータ及び前記係数バッファに格納されているデータを用いて前記ピクセル演算を行うと共に、該上位面のスプライトを前記フレームバッファに描画した後に、該下位面のスプライトを該フレームバッファに描画するピクセル演算モジュールとを有することを特徴とする画像処理装置。 20

【請求項 3】

表示画像の各画素に対応させて設定されている係数データであると共に、各画素位置の透過度情報を示す係数データである 係数を格納する 係数バッファを用いて、前記表示画像をフレームバッファに一旦保持させて表示手段に表示させる画像処理方法であって、

前記 係数を使って 2 つの画像を半透明合成する処理である ブレンディング実行時は、該 2 つの画像における表示優先度が低い方の画像である下位面に対する 係数を計算し、該 係数を前記 係数バッファに格納し、

前記 2 つの画像における表示優先度が高い方の画像である上位面又は前記下位面を前記フレームバッファに描画するときは、描画データ、描画済みデータ及び前記 係数バッファに格納されているデータを用いて ブレンディングを行うと共に、該上位面のスプライトを前記フレームバッファに描画した後に、該下位面のスプライトを該フレームバッファに描画することを特徴とする画像処理方法。 40

【請求項 4】

表示画像の各画素に対応させて設定されている係数データであると共に、2 つの画像を重ねてなる合成画像の各画素データを算出するピクセル演算に用いる係数データを格納する 係数バッファを用いて、前記表示画像をフレームバッファに一旦保持させて表示手段に表示させる画像処理方法であって、

前記合成画像を算出するときに、前記 2 つの画像における表示優先度が低い方の画像である下位面に対する前記係数データを計算し、該係数データを前記係数バッファに格納し、

前記 2 つの画像における表示優先度が高い方の画像である上位面又は前記下位面を前記フレームバッファに描画するときは、描画データ、描画済みデータ及び前記係数バッファ 50

に格納されているデータを用いて前記ピクセル演算を行うと共に、該上位面のスプライトを前記フレームバッファに描画した後に、該下位面のスプライトを該フレームバッファに描画することを特徴とする画像処理方法。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、画像処理装置および画像処理方法に関する。

【背景技術】

【0002】

従来、ビデオゲーム装置などに適用される画像表示装置では、スプライト表示方式が用いられている。ここで、スプライト表示方式とは、画面に表示されるキャラクタ毎に表示位置などの属性（スプライト属性）を持ち、そのスプライト属性に従いキャラクタを配置することで画面全体を構成する方式である。そして、表示位置などのスプライト属性を持ったキャラクタをスプライトという。ビデオゲーム装置などのように、インターラクティブに高速にキャラクタを動かす場合、スプライト表示方式では動かすキャラクタのスプライト属性を変更するだけで画面の書き換えをすることができる。

10

【0003】

また、従来の画像表示装置では、2つ以上の画像を係数（アルファ値又は係数）を使って半透明合成するブレンディング（アルファブレンド）機能を備えるものがある。このブレンディング機能では、画像の全画素に共通の係数を指定することもあるが、画像の各画素に対応させて係数の情報を持たせてそれを利用することによって画素ごとの半透明合成を行うこともできる（例えば、特許文献1参照）。

20

【特許文献1】特開2002-341859号公報

【発明の開示】

【発明が解決しようとする課題】

【0004】

しかしながら、特許文献1などに記載されている従来の画像処理装置では、ブレンディングを行うためには、複数のスプライトにおいて表示優先順位の低いスプライトから順番にフレームバッファに描画する必要がある。カラー演算を行う場合もあるが、基本的には、優先順位の高いスプライトは優先順位の低いスプライトの上に上書きされるからである。

30

【0005】

フレームバッファに表示優先順位の低いスプライトから順番に描画する場合は、全てのスプライトをフレームバッファに描画する必要がある。ここで、描画性能は、単位時間に描画できるドット数で表されるが、従来の画像処理装置では、表示優先順位の低いスプライトから全てのスプライトを順番にフレームバッファに描画する必要があるため、描画性能の向上が阻害されているという問題点があった。

【0006】

本発明は、上記問題点を解決するためになされたものであり、表示優先順位の高いスプライトから描画することができ、かつ複数面についてのカラー演算を行うこともできる画像処理装置および画像処理方法を提供するものである。

40

【課題を解決するための手段】

【0007】

上記課題を解決するため、この発明は以下の構成を有する。

即ち、請求項1に記載された発明に係る画像処理装置は、表示手段に出力される表示画像データをビットマップ形式で保持するフレームバッファと、前記表示画像データの各画素に対応させて設定されている係数データであると共に、各画素位置の透過度情報を示す係数データである係数を格納する係数バッファと、前記係数を使って2つの画像を半透明合成する処理であるブレンディングの実行時に、該2つの画像における表示優先度が低い方の画像である下位面に対する係数を計算し、該係数を前記係数バッファ

50

に格納させる係数計算モジュールと、前記2つの画像における表示優先度が高い方の画像である上位面又は前記下位面を前記フレームバッファに描画するときに、描画データ、描画済みデータ及び前記係数バッファに格納されているデータを用いてブレンディングを行うと共に、該上位面のスプライトを前記フレームバッファに描画した後に、該下位面のスプライトを該フレームバッファに描画するブレンディングモジュールとを有することを特徴とする。

【0008】

また、請求項2に記載された発明に係る画像処理装置は、表示手段に出力される表示画像データをピットマップ形式で保持するフレームバッファと、前記表示画像データの各画素に対応させて設定されている係数データであると共に、2つの画像を重ねてなる合成画像の各画素データを算出するピクセル演算に用いる係数データを格納する係数バッファと、前記ピクセル演算を算出するときに、前記2つの画像における表示優先度が低い方の画像である下位面に対する前記係数データを計算し、該係数データを前記係数バッファに格納させる係数計算モジュールと、前記2つの画像における表示優先度が高い方の画像である上位面又は前記下位面を前記フレームバッファに描画するときに、描画データ、描画済みデータ及び前記係数バッファに格納されているデータを用いて前記ピクセル演算を行うと共に、該上位面のスプライトを前記フレームバッファに描画するピクセル演算モジュールとを有することを特徴とする。

【0009】

また、請求項3に記載された発明に係る画像処理方法は、表示画像の各画素に対応させて設定されている係数データであると共に、各画素位置の透過度情報を示す係数データである係数を格納する係数バッファを用いて、前記表示画像をフレームバッファに一旦保持させて表示手段に表示させる画像処理方法であって、前記係数を使って2つの画像を半透明合成する処理であるブレンディング実行時は、該2つの画像における表示優先度が低い方の画像である下位面に対する係数を計算し、該係数を前記係数バッファに格納し、前記2つの画像における表示優先度が高い方の画像である上位面又は前記下位面を前記フレームバッファに描画するときは、描画データ、描画済みデータ及び前記係数バッファに格納されているデータを用いてブレンディングを行うと共に、該上位面のスプライトを前記フレームバッファに描画した後に、該下位面のスプライトを該フレームバッファに描画することを特徴とする。

【0010】

また、請求項4に記載された発明に係る画像処理方法は、表示画像の各画素に対応させて設定されている係数データであると共に、2つの画像を重ねてなる合成画像の各画素データを算出するピクセル演算に用いる係数データを格納する係数バッファを用いて、前記表示画像をフレームバッファに一旦保持させて表示手段に表示させる画像処理方法であって、前記合成画像を算出するときに、前記2つの画像における表示優先度が低い方の画像である下位面に対する前記係数データを計算し、該係数データを前記係数バッファに格納し、前記2つの画像における表示優先度が高い方の画像である上位面又は前記下位面を前記フレームバッファに描画するときは、描画データ、描画済みデータ及び前記係数バッファに格納されているデータを用いて前記ピクセル演算を行うと共に、該上位面のスプライトを前記フレームバッファに描画した後に、該下位面のスプライトを該フレームバッファに描画することを特徴とする。

【発明の効果】

【0011】

本発明によれば、表示優先順位の高いスプライトから描画することができ、かつ複数面についてのカラー演算を行うこともできる。

【発明を実施するための最良の形態】

【0012】

以下、図面を参照し、本発明の実施形態について説明する。

10

20

30

40

50

(第1実施形態)

図1は、本発明の実施形態に係る画像処理装置の構成例を示すブロック図である。本発明の実施形態に係る画像処理装置20は、画像表示装置1の構成要素となる。画像表示装置1は、CGメモリ11と、CPU12と、モニタ13と、本発明の実施形態に係る画像処理装置20とで構成されている。CPU12は、画像処理装置20の動作を制御し、CGメモリ11に格納されているキャラクタデータを用いて、モニタ13に所望の画面を表示させる。モニタ13は、LCDパネル又はCRTモニタなどの表示手段である。

【0013】

画像処理装置20は、CGメモリインターフェース21、リアルタイムデコーダ22、スプライトバッファ23、レンダリングプロセッサ24、フレームバッファ25、フレームデータ制御回路26、D/Aコンバータ27、スプライト・プレーン・ジェネレータ28、CPUインターフェース29、DMA制御回路30、カラーパレット31、汎用データテーブル32、レジスタ33、モニタ制御回路34及びクロック生成回路35で構成されている。

【0014】

まず、画像処理装置20の主要概略動作について説明する。画像処理装置20は、CPU12の指示(汎用データテーブル32内のスプライト属性テーブルへのスプライト属性データの書き込み)にしたがって動作する。すなわち、画像処理装置20は、1フレーム期間内において、汎用データテーブル32内のスプライト属性テーブルからスプライト属性データを順次読み出し、そのスプライト属性データに基づき、CGメモリ11に記憶されたキャラクタを使用してスプライトを生成し、フレームバッファ25に描画する。なお、詳細は後述するが、本発明の画像処理装置では、上記スプライト属性テーブルからのスプライト属性データの読み出し(すなわち、スプライトのフレームバッファ25への描画)は、表示優先順位の高いスプライトから行うこともできる。そのために、描画の際のブレンディング処理が従来とは異なり特徴的になっている。所定フレームにおいてフレームバッファ25に描画された画像データは次のフレームにおいてモニタ13に出力され表示される。

【0015】

次に、画像処理装置20の詳細について説明する。

CGメモリインターフェース21は、CGメモリ11に対する読み出しを制御する回路である。リアルタイムデコーダ22は、CGメモリインターフェース21を介してCGメモリ11から読み出されたキャラクタデータであって圧縮されたキャラクタデータを解凍してRGBデータに変換する回路である。スプライトバッファ23は、1ないし複数のキャラクタデータのテンポラリバッファである。そのキャラクタデータはビットマップ形式で格納されている。

【0016】

レンダリングプロセッサ24は、スプライトの拡大・縮小・変形及びカラー演算を行う回路である。フレームバッファ25は、複数のスプライトで構成される画面データ(フレームデータ)を保持するためのバッファである。フレームデータは、フレームバッファ25上にスプライトを描画することで作られている。このフレームバッファ25は、例えばダブルバッファ構成となっており、表示スキャンにあわせてフレームデータを読み出し中であっても、次のフレームの画面データを描画することができる。

【0017】

フレームデータ制御回路26は、モニタ制御回路34で生成されたモニタのスキャンタイミングにしたがって、フレームバッファ25からフレームデータを読み出し、4倍拡大フィルタ処理又はモザイク処理などを実施してドットデータとしてモニタ13へ送出する回路である。カラーパレット31は、カラーコードをRGBデータに変換する変換テーブルである。したがって、パレットモードではキャラクタを構成する各ドットがカラーコードで記憶されており、通常モード(RGBモード)のときよりもデータ圧縮が可能になる。D/Aコンバータ27は、フレームデータ制御回路26から出力されたデジタル信号を

10

20

30

40

50

アナログ信号に変換してモニタ13へ出力する回路である。

【0018】

スプライト・プレーン・ジェネレータ28は、リアルタイムデコーダ22、スプライトバッファ23、及びレンダリングプロセッサ24の動作を制御する回路である。CPUインターフェース29は、CPU12から、汎用データテーブル32、カラーパレット31、レジスタ33及びCGメモリ11へのアクセスを制御する回路である。DMA制御回路30は、CPUバスを介してのアクセスとは独立して、CGメモリ11及び汎用データテーブル32から画像処理装置20内部の各テーブル及びレジスタへデータを転送するローカルDMA機能を制御する回路である。

【0019】

汎用データテーブル32は、画像処理装置20に内蔵されている書換可能な記憶回路である。汎用データテーブル32は、スプライト属性テーブル、4頂点変形テーブル、スプライト・ラインテーブルなどを格納するものとする。レジスタ33は、スプライト表示を制御する制御データを設定するレジスタである。モニタ制御回路34は、レジスタ33の設定に基づいてモニタ13の走査タイミングを生成する回路である。クロック生成回路35は、画像処理装置20の内部動作クロックを生成する回路である。

【0020】

次に、上記構成の画像処理装置20の動作について説明する。

先ず、CPU12は、画像処理装置20に内蔵されているレジスタ33及び汎用データテーブル32に制御データを設定することにより、モニタ13に表示させる画面(フレーム)を制御する。ここで、制御データは、スプライト属性データなどのスプライト表示するためのデータである。モニタ13に表示される画面(フレーム)は、複数のスプライトが組み合わされて構成されるものとする。そして、フレーム上のどの位置にどのような形でスプライトを表示するかといった表示属性は、汎用データテーブル32にマッピングされるスプライト属性テーブルに設定される。

【0021】

画像処理装置20のスプライト・プレーン・ジェネレータ28は、フレーム毎にスプライト属性テーブルのスプライト属性データを所定の順序で読み出し、そこに設定されたスプライト表示属性にしたがってCGメモリ11に格納されているキャラクタを読み出す。キャラクタは、CGメモリ11に予め格納されている。CGメモリ11上のキャラクタは、通常圧縮された形で格納されているが、非圧縮で格納されていてもよい。CGメモリ11から読み出されたキャラクタは、リアルタイムに解凍されRGBデータに変換された後、スプライトバッファ23に一旦格納される。

【0022】

スプライトバッファ23に格納されたスプライトは、スプライト属性データにしたがい、レンダリングプロセッサ24により、拡大、縮小、反転、変形又はカラー演算などの処理が施され、スプライト属性データにしたがい、フレームバッファ25上のスプライト表示位置に転送される。

これらの処理はモニタ13に表示するスプライトの個数分だけ実施され、フレームバッファ25上に画面データを構成する。

【0023】

フレームバッファ25はダブルバッファ構造となっているので、スプライトにより画面を構築中においても、フレームバッファ25で既に構築された画面データをモニタ13へ表示させることができる。画面データは、モニタ制御回路34で生成されたモニタの走査タイミングに基づいて、フレームバッファ25から読み出され、4倍拡大フィルタ処理又はモザイク処理を施されながら、モニタ13へドットデータとして送出される。モニタ13がアナログ信号を入力信号とするものであるときは、フレームバッファ25の出力はD/Aコンバータ27を介してモニタ13へ送出される。なお、モニタ13がデジタル信号を入力信号とするときは、フレームバッファ25の出力が直接モニタ13へ送出される。

【0024】

10

20

30

40

50

次に、画像処理装置 20 のレンダリングプロセッサ 24 で行われる ブレンディングについて、図 2 から図 4 を参照して説明する。図 2 から図 4 に示す ブレンディングは、従来から用いられている手法である。本発明に係る画像処理装置 20 は、図 2 から図 4 に示す従来の ブレンディングを実現できると共に、さらに従来の画像処理装置を改良して、表示優先順位の高いスプライトから描画することができるようにしたものである。そこで、まず ブレンディングについて説明する。

図 2 は、係数に対応した演算式を示す図である。係数は、汎用データテーブル 32 にマッピングされているスプライト属性テーブルに設定されている。ブレンディングは、複数の画像を 係数を使って半透明合成する処理である。具体的には、次のように行う。
10

【 0 0 2 5 】

フレームバッファ 25 ヘススプライトを描画するときに、フレームバッファ 25 に描画しようとするデータ（前レイヤ：fore layer のカラーデータ）とすでに描画されているデータ（後レイヤ：back layer のカラーデータ）との間で、係数に従い図 2 に示す演算を行う。この演算を行った結果を描画する機能が ブレンディングである。例えば、係数が「0 0 0 0 0」の場合、図 2 に示す複数の演算式における 1 番上の演算式が用いられ、その演算結果を描画する。

【 0 0 2 6 】

次に、上記 ブレンディングを行う回路構成について、図 3 及び図 4 を参照して説明する。図 3 は、画像処理装置 20 内のレンダリングプロセッサ 24 で構成されている ブレンディング回路を示すブロックである。ブレンディング回路は、上記フレームバッファ 25 と、ブレンディングモジュール 41 とで構成されている。ブレンディングモジュール 41 は、上記レンダリングプロセッサ 24 の一部として構成されている。
20

【 0 0 2 7 】

ブレンディングモジュール 41 は、先ず、フレームバッファ 25 における描画しようとする位置にすでに描画されているデータ（後レイヤのカラーデータ = リードデータ）を、そのフレームバッファ 25 から読み出す。次いで、ブレンディングモジュール 41 は、フレームバッファ 25 に描画しようとするデータ（前レイヤのカラーデータ = レンダリングデータ）と、フレームバッファ 25 から読み出されたリードデータとの間で、係数に従い ブレンディングを実行する。そして、ブレンディングモジュール 41 は、ブレンディングされたデータをフレームバッファ 25 に書き込む。
30

ブレンディングモジュール 41 は、ブレンディングとして例えば下記数式（1）の計算を行う。

$$(32 -) / 32 \times (\text{rendering data}) + / 32 \times (\text{read data}) \dots (1)$$

【 0 0 2 8 】

図 4 は、ブレンディングモジュール 41 の具体的な構成例を示す回路図である。ブレンディングモジュール 41 は、上記数式（1）の計算を実行する回路である。ブレンディングモジュール 41 は、セレクタ 51, 52, 53, 54, 55 と、全加算機 56, 57, 58, 59, 60 と、シフター 61, 62, 63, 64, 65 とを有して構成されている。セレクタ 51, 52, 53, 54, 55 は、レンダリングデータとリードデータとのうちで、係数で指定される方を選択して出力する。
40

【 0 0 2 9 】

具体的にはセレクタ 51 は、レンダリングデータとリードデータとのうち、係数の最下位ビットで指定される方を出力する。すなわち、係数の最下位ビットが「0」を示せばレンダリングデータを、「1」を示せばリードデータを選択して出力する。同様に、セレクタ 52, 53, 54, 55 は、レンダリングデータとリードデータとのうち、係数の第 1, 2, 3, 4 ビットで指定される方をそれぞれ出力する。

【 0 0 3 0 】

全加算機 56 は、レンダリングデータとセレクタ 51 の出力とを加算して出力する。シフター 61 は、全加算機 56 の 7 ビットの出力の上位側の 6 ビットを出力する。同様に、
50

全加算機 57, 58, 59, 60 は、シフター 61, 62, 63, 64 の出力とセレクタ 52, 53, 54, 55 の出力を加算してそれぞれ出力する。シフター 65 は、全加算機 60 の 7 ビットの出力の上位側の 6 ビットを出力する。

そして、シフター 65 の出力が ブレンディングモジュール 41 の出力、すなわち上記数式(1)の計算結果となる。

【0031】

例えれば 係数が「00000」の場合は、レンダリングデータのカラーデータをそのまま ブレンディングモジュール 41 が出力する。係数が「00001」の場合は、レンダリングデータ $\times 31 / 32 +$ リードデータ $\times 1 / 32$ が ブレンディングモジュール 41 の出力となる。そして、係数の値が大きくなるほどレンダリングデータが小さくリードデータが大きく重み付けされて ブレンディングモジュール 41 の出力となる。係数が「11111」の場合は、レンダリングデータ $\times 1 / 32 +$ リードデータ $\times 31 / 32$ が ブレンディングモジュール 41 の出力となる。

したがって、ブレンディングモジュール 41 の出力は、レンダリングデータとリードデータとを 係数を重み付けとしてカラー合成し、その合成結果をカラーデータとして出力する。

【0032】

(第1カラー演算方法)

次に、本実施形態に係る画像処理装置 20 で実行される第1カラー演算方法(処理)であって、表示優先順位の高いスプライトから描画した場合の複数スプライトのカラー演算方法(処理)について説明する。

図 5 は、4つのスプライトが重なった状態を示す模式概念図である。図 5 に示すように、4つのスプライトが重なった場合の第1カラー演算方法について説明する。Sprite A は最上位スプライトすなわち優先順位が1番高いスプライトである。Sprite B は、優先順位が2番目のスプライトである。Sprite C は、優先順位が3番目のスプライトである。Sprite D は、最下位スプライトすなわち優先順位が4番目のスプライトである。

まず、第1カラー演算方法を実現するための回路構成について図 6 を参照して説明する。図 6 は、表示優先順位の高いスプライトから描画した場合の複数スプライトのカラー演算処理を行う回路、すなわち第1カラー演算処理実行回路を示すブロック図である。

【0033】

第1カラー演算処理回路は、フレームバッファ 25 と、係数計算モジュール 71 と、係数バッファ 72 と、ブレンディングモジュール 73 とで構成されている。第1カラー演算回路は、図 1 に示す画像処理装置 20 の構成要素として形成されているものである。フレームバッファ 25 は、上述のように複数のキャラクタで構成される画面データ(フレームデータ)を保持するためのバッファである。

【0034】

係数計算モジュール 71 は、ブレンディング実行時に下位面(下位層)に対する係数を計算し、その計算結果の係数を係数バッファ 72 に格納させるモジュールである。係数バッファ 72 は、表示画像の各画素に対応させて係数を格納するバッファである。ブレンディングモジュール 73 は、下位面描画時に描画データ、描画済みデータ及び係数バッファ 72 に格納されているデータを用いてブレンディングを行うモジュールである。ここで、係数計算モジュール 71、係数バッファ 72 及びブレンディングモジュール 73 は、上記レンダリングプロセッサ 24 の構成要素としてもよい。係数バッファ 72 は、汎用データテーブル 32 にマッピングされているものとしてもよい。

【0035】

次に、図 5 に示すように4つのスプライトが重なった場合の第1カラー演算方法について説明する。

Sprite A の RGB データを D_a、Sprite A の 係数を a、Sprite B の RGB データを D_b、Sprite B の 係数を b、Sprite C の RGB データを D_c、Sprite C の 係数を c、Sprite D の RGB データを D_d とする。Sprite D は、最下位スプライトであるため

10

20

30

40

50

ブレンディングは実行しない。

【0036】

そして、図5に示すように4つのスプライトが重なった場合の第1カラー演算方法に係るカラー演算式は下記数式(2)のようになる。

$$\begin{aligned} & (32 - a) / 32 \times D_a + \\ & a / 32 \times \{ (32 - b) / 32 \times D_b + \\ & b / 32 \times [(32 - c) / 32 \times D_c + \\ & c / 32 \times D_d] \} \quad \dots (2) \end{aligned}$$

【0037】

上記数式(2)を展開すると下記数式(3)のようになる。

$$\begin{aligned} & (32 - a) / 32 \times D_a + \\ & a / 32 \times (32 - b) / 32 \times D_b + \\ & a / 32 \times b / 32 \times (32 - c) / 32 \times D_c + \\ & a / 32 \times b / 32 \times c / 32 \times D_d \quad \dots (3) \end{aligned}$$

【0038】

次に、図6に示す第1カラー演算処理実行回路を用いた第1カラー演算方法について具体的に説明する。

最初に、ブレンディングモジュール73は、SpriteAをフレームバッファ25に描画する。第1カラー演算方法では、フレームバッファ25へは表示優先順位の高いスプライトから描画する。このときSpriteAはカラー演算が有効であるので、ブレンディングモジュール73は、SpriteAのデータと係数との間で計算を行いその計算結果をフレームバッファ25に書き込む。このフレームバッファ25に書き込むデータFDaは下記数式(4)により計算される。

$$FDa = (32 - a) / 32 \times Da \quad \dots (4)$$

同時に、係数計算モジュール71は、係数バッファ72にSpriteAの係数(a)を書き込む。

【0039】

次いで、SpriteBを描画する。このとき、SpriteBはカラー演算が有効であるので、ブレンディングモジュール73及び係数計算モジュール71は、SpriteBのデータと、SpriteBの係数と、フレームバッファ25に書き込まれているデータFDaとを用いて、下記数式(5)、数式(6)に示す計算を行い、その計算結果をフレームバッファ25及び係数バッファ72に書き込む。下記数式(5)はブレンディングモジュール73で行われる計算であり、その計算結果であるデータFDabはフレームバッファ25に書き込まれる。

$$FDab = FDA + a / 32 \times (32 - b) / 32 \times Db \quad \dots (5)$$

下記数式(6)は係数計算モジュール71で行われる計算であり、その計算結果である係数は係数バッファ72に書き込まれる。

$$\text{係数} = a \times b \quad \dots (6)$$

【0040】

次いで、SpriteCを描画する。このとき、SpriteCはカラー演算が有効であるので、ブレンディングモジュール73及び係数計算モジュール71は、SpriteCのデータと、SpriteCの係数と、フレームバッファ25に書き込まれているデータFDabと、係数バッファ72に書き込まれているデータ(係数 = a × b)とを用いて、下記数式(7)、数式(8)に示す計算を行い、その計算結果をフレームバッファ25及び係数バッファ72に書き込む。下記数式(7)はブレンディングモジュール73で行われる計算であり、その計算結果であるデータFDabcはフレームバッファ25に書き込まれる。

$$\begin{aligned} & FDA + a / 32 \times b / 32 \times (32 - c) / 32 \times Dc \quad \dots (7) \end{aligned}$$

下記数式(11)は係数計算モジュール71で行われる計算であり、その計算結果で

10

20

30

40

50

ある 係数は 係数バッファ 7 2 に書き込まれる。

$$\text{係数} = a \times b \times c \quad \dots (8)$$

【0041】

最後に、最上位面のSpriteDを描画する。このとき、SpriteDはカラー演算が無効であるので、ブレンディングモジュール73及び係数計算モジュール71は、SpriteDのデータと、SpriteDの係数と、フレームバッファ25に書き込まれているデータF D a b c と、係数バッファ72に書き込まれているデータ(係数 = a × b × c)とを用いて、下記式(9)に示す計算を行い、その計算結果をフレームバッファ25に書き込む。下記式(9)はブレンディングモジュール73で行われる計算であり、その計算結果であるデータF D a b c d はフレームバッファ25に書き込まれる。

10

$$\begin{aligned} F D a b c d &= F D a b c + \\ &\quad a / 32 \times b / 32 \times c / 32 \times D d \quad \dots (9) \end{aligned}$$

また、係数計算モジュール71は、係数バッファ72へ「0」を示すデータを書き込む。

【0042】

これらにより、第1カラー演算方法では、フレームバッファ25に対して表示優先順位の高いスプライトから描画することができ、かつ、カラー演算(ブレンディング)をすることができる。また、カラー演算しない場合には、すでに描画済の領域の描画は不要になるため、第1カラー演算方法によれば従来よりも描画性能を向上させることができる。

20

【0043】

(第2カラー演算方法)

次に、表示優先順位の低いスプライトから描画した場合の複数スプライトのカラー表示方法(第2カラー演算方法)について説明する。第2カラー演算方法は従来の画像処理装置で用いられているカラー演算方法である。本実施形態に係る画像処理装置20は、この表示優先順位の低いスプライトから描画した場合の複数スプライトのカラー表示方法を実行することもできるが、上記第1カラー演算方法によって従来の画像処理装置よりも描画性能を向上させることができる。

【0044】

図5に示すように、SpriteAのRGBデータをD a、SpriteAの係数をa、SpriteBのRGBデータをD b、SpriteBの係数をb、SpriteCのRGBデータをD c、SpriteCの係数をc、SpriteDのRGBデータをD dとする。SpriteDは、最下位スプライトであるためブレンディングは実行しない。

30

最初に、SpriteDをフレームバッファ25に描画する。第2カラー演算方法では、フレームバッファ25へは表示優先順位の低いスプライトから描画する。

【0045】

次いで、SpriteCを描画する。このとき、SpriteCはカラー演算が有効であるので、SpriteDとSpriteCとについてカラー演算を実行した結果をフレームバッファ25に書き込む。このときフレームバッファ25に書き込む値をD c dとする。D c dの演算は下記式(10)のように行う。

$$D c d = (32 - c) / 32 \times D c + c / 32 \times D d \quad \dots (10)$$

40

【0046】

次いで、SpriteBを描画する。このとき、SpriteBはカラー演算が有効であるので、現在フレームバッファ25に描画されている値(D c d)とSpriteBとについてカラー演算した結果をフレームバッファ25に書き込む。このときフレームバッファ25に書き込む値をD b c dとする。D b c dの演算は下記式(11)のように行う。

$$D b c d = (32 - b) / 32 \times D b + b / 32 \times D c d \quad \dots (11)$$

【0047】

最後に、最上位面のSpriteAを描画する。このとき、SpriteAはカラー演算が有効であるので、現在フレームバッファ25に描画されている値(D b c d)とSpriteAとについてカラー演算した結果をフレームバッファ25に書き込む。このときフレームバッファ2

50

5に書き込む値をD a b c dとする。D a b c dの演算は下記式(12)のようを行う。

$$D a b c d = (32 - a) / 32 \times D a + a / 32 \times D b c d \dots (12)$$

【0048】

このように、表示優先順位の低いスプライトから描画することにより、複数面のカラー演算を実現することができる。

上記第2カラー演算方法のように、表示優先順位の低いスプライトから描画すると、必ず全スプライトをフレームバッファ25に描画する必要がある。このため、描画性能が上がらないというデメリットがある。一方、上記第1カラー演算方法のように、表示優先順位の高いスプライトから描画すると、カラー演算を行わない場合には、既に描画済みのドットは描画しなくてもよいため、描画性能を向上させることができるというメリットが生じる。しかし、従来においては、表示優先順位の高いスプライトから描画すると、複数面のカラー演算を実現することができなかった。10

【0049】

そこで、本発明の実施形態に係る画像処理装置20は、上記第1カラー演算方法により、表示優先順位の高いスプライトから描画することができ、かつ複数面についてのカラー演算を実行できるので、上記第2カラー演算方法しか実行できない従来の画像処理装置よりも描画性能を向上させることができる。20

【0050】

以上、本発明の実施形態について図面を参照して詳述してきたが、具体的な構成はこの実施形態に限られるものではなく、本発明の要旨を逸脱しない範囲の設計変更等も含まれる。

【0051】

以上の説明では、本発明に係る画像処理装置20がカラー演算(ブレンディング)を行う実施形態について説明したが、本発明はこれに限定されるものではなく、画像処理装置20がカラー演算の代わりに(又はカラー演算とともに)ピクセル演算を行うものとしてもよい。すなわち、画像処理装置20は、表示優先順位の高いスプライトから描画することができ、かつ、複数面についてのピクセル演算を行うことができるものとしてもよい。なお、ピクセル演算とは、乗算演算、加算演算などである。

また、上記実施形態では、画像処理装置20をすべてハードウェアで構成することとしたが、本発明はこれに限定されるものではなく、本発明に係る画像処理装置の一部構成をソフトウェアで実現してもよい。30

【図面の簡単な説明】

【0052】

【図1】本発明の実施形態に係る画像処理装置の構成例を示すブロック図である。

【図2】係数に対応した演算式を示す図である。

【図3】本発明の実施形態に係る画像処理装置内で構成されているブレンディング回路を示すブロックである。

【図4】本発明の実施形態に係る画像処理装置内で構成されているブレンディングモジュールの構成例を示す回路図である。40

【図5】4つのスプライトが重なった状態を示す模式概念図である。

【図6】本発明の実施形態に係る画像処理装置内で構成されている第1カラー演算処理実行回路を示すブロック図である。

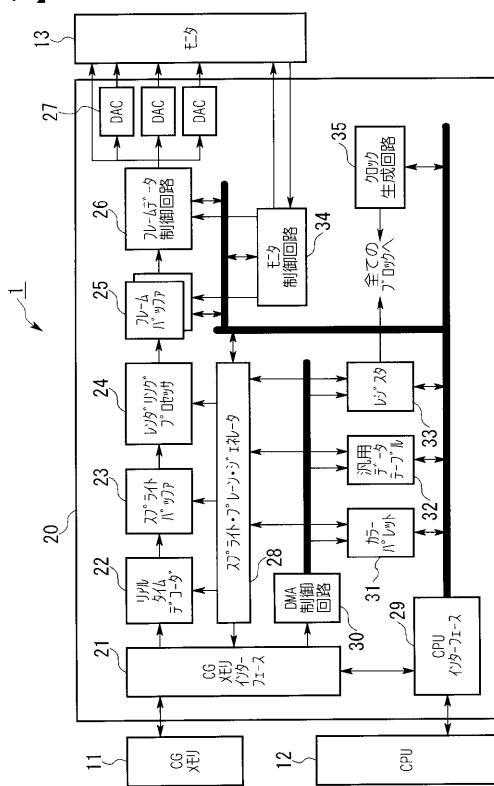
【符号の説明】

【0053】

1...画像表示装置、11...CGメモリ、12...CPU、13...モニタ、20...画像処理装置、21...CGメモリインターフェース、22...リアルタイムデコーダ、23...スプライトバッファ、24...レンダリングプロセッサ、25...フレームバッファ、26...フレームデータ制御回路、27...D/Aコンバータ、28...スプライト・プレーン・ジェネレータ、29...CPUインターフェース、30...DMA制御回路、31...カラーパレット、350

2 ... 汎用データテーブル、33...レジスタ、34...モニタ制御回路、35...クロック生成回路、41...ブレンディングモジュール、51, 52, 53, 54, 55...セレクタ、56, 57, 58, 59, 60...全加算機、61, 62, 63, 64, 65...シフター、71...係数計算モジュール、72...係数バッファ、73...ブレンディングモジュール

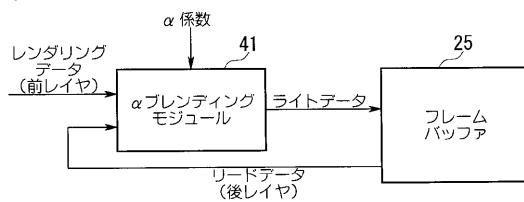
【図1】



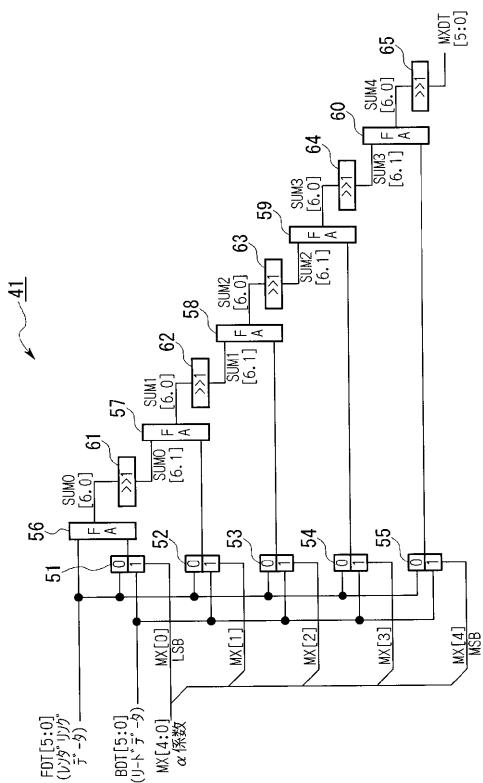
【図2】

MX[4:0]	演算式
5'b00000	(fore layer) × 32/32 + (back layer) × 0/32 が→演算しない場合、この値に設定する。
5'b00001	(fore layer) × 31/32 + (back layer) × 1/32
5'b00010	(fore layer) × 30/32 + (back layer) × 2/32
n	(中略) (fore layer) × (32-n)/32 + (back layer) × n/32 (中略)
5'b11110	(fore layer) × 2/32 + (back layer) × 30/32
5'b11111	(fore layer) × 1/32 + (back layer) × 31/32

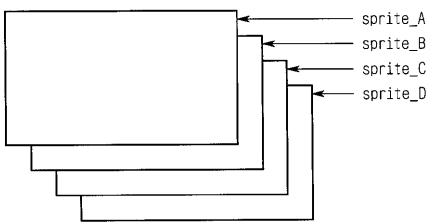
【図3】



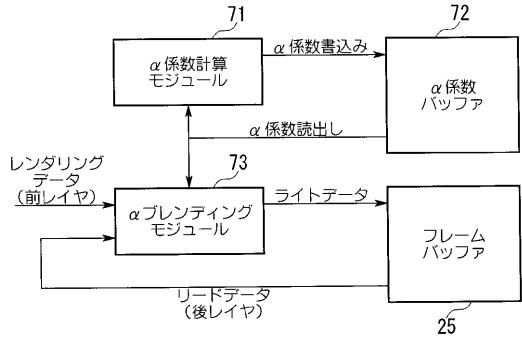
【図4】



【図5】



【図6】



フロントページの続き

(51)Int.Cl.⁷

F I

テーマコード(参考)

G 0 9 G 5/36 5 2 0 M