

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第4349733号
(P4349733)

(45) 発行日 平成21年10月21日(2009.10.21)

(24) 登録日 平成21年7月31日(2009.7.31)

(51) Int.Cl. F I
G06T 15/00 (2006.01) G O 6 T 15/00 I O O A
G06T 15/70 (2006.01) G O 6 T 15/70 A

請求項の数 12 (全 22 頁)

(21) 出願番号	特願2000-289625 (P2000-289625)	(73) 特許権者	000233778
(22) 出願日	平成12年9月22日 (2000.9.22)		任天堂株式会社
(65) 公開番号	特開2001-134779 (P2001-134779A)		京都府京都市南区上鳥羽鉾立町11番地1
(43) 公開日	平成13年5月18日 (2001.5.18)	(74) 代理人	100098291
審査請求日	平成18年7月24日 (2006.7.24)		弁理士 小笠原 史朗
(31) 優先権主張番号	60/155660	(72) 発明者	安本 吉孝
(32) 優先日	平成11年9月24日 (1999.9.24)		京都府京都市東山区福稲上高松町60番地
(33) 優先権主張国	米国 (US)		任天堂株式会社内
(31) 優先権主張番号	09/468109	審査官	伊知地 和之
(32) 優先日	平成11年12月21日 (1999.12.21)		
(33) 優先権主張国	米国 (US)	(56) 参考文献	特開平09-311954 (JP, A)
			特開平08-131252 (JP, A)
			特開平09-185699 (JP, A)

最終頁に続く

(54) 【発明の名称】 三次元ビデオグラフィックスシステムにおける非写実的な漫画的な輪郭線提供方法およびその装置

(57) 【特許請求の範囲】

【請求項1】

対話型の三次元画像処理装置であって、
 ユーザによるリアルタイム操作入力が可能で、少なくとも1つの手動制御器と、
 少なくとも1つの複数の部位から成るキャラクタを描画するための三次元データを記憶する記憶媒体と、

画素ごとにカラーデータと深度データとを含む画像データを記憶するバッファメモリと、
 前記バッファメモリに接続される三次元グラフィックスパイプラインと、

前記バッファメモリに接続され、輪郭線処理を前記画像データに適用して、前記キャラクタの周囲に境界線を生成するフィルタとを備え、

前記三次元グラフィックスパイプラインは、少なくとも前記ユーザによるリアルタイム操作入力および前記キャラクタを表現する前記三次元データに基づいて、前記キャラクタに対応する画像データを前記バッファメモリにおいて描画し、

前記記憶媒体は、前記複数の部位それぞれに異なる識別値を記憶し、

前記フィルタは、前記バッファメモリに記憶されている画素の深度データと、前記部位の識別値とに基づいて、前記カラーデータを選択的に変更して境界線を生成する、三次元画像処理装置。

【請求項2】

前記フィルタは、前記画素に対応する深度データと、前記画素周辺に或る1つの画素に対応する深度データとの差を求め、これにより前記差が所定の閾値を上回った場合には、

前記画素に対応する前記カラーデータを変更することを特徴とする、請求項 1 に記載の三次元画像処理装置。

【請求項 3】

前記バッファメモリは、前記画素ごとにアルファ値を更に含み、

前記フィルタは、前記画素に対応するアルファ値を、前記画素の深度データに対応する値に設定することを特徴とする、請求項 1 に記載の三次元画像処理装置。

【請求項 4】

前記フィルタは、前記画素に対応するアルファ値を、前記画素の深度と、その周辺の画素の深度との差に対応する値に設定することにより、前記バッファメモリに記憶されるカラーデータを選択的に変更することを特徴とする、請求項 3 に記載の三次元画像処理装置

10

【請求項 5】

前記画素は、前記画像において $P[x, y]$ であると指定され、前記フィルタは、前記画素の深度を $P[x - 1, y]$ であると指定される第 1 画素と、 $P[x, y - 1]$ であると指定される第 2 画素の深度と比較することを特徴とする、請求項 1 に記載の三次元画像処理装置。

【請求項 6】

複数の部位から成り、複数の部位ごとに識別値を記憶した三次元画像を描画するためのデータと、画像の画素に対応するカラーデータおよび深度データを記憶する画素メモリを含むコンピュータグラフィックスシステムにおける三次元画像処理方法であって、

20

(a) 少なくとも 1 つの手動制御器からユーザのリアルタイム操作入力を取得するステップ、

(b) 前記画素メモリから画素データを読み出し、当該画素に対応する深度データと、部位の識別値とに基づいて、当該画素のカラーデータを選択的に変更するステップ、

(c) 前記三次元画像を描画するためのデータと、前記変更されたカラーデータと、前記リアルタイム操作入力とに基づいて、三次元画像を描画するステップ、
からなる三次元画像処理方法。

【請求項 7】

前記三次元画像を描画するステップは、前記画素に対応する深度データと、前記画素の少なくとも 1 つの隣接画素に対応する深度データとの差を求め、これにより前記差が所定の閾値を上回った場合には、前記画素に対応する前記カラーデータを変更することを特徴とする、請求項 6 に記載の方法。

30

【請求項 8】

前記画像メモリは、前記画素ごとにアルファ値を更に含み、

前記三次元画像を描画するステップは、前記画素に対応するアルファ値を、前期画素の深度データに対応する値に設定することを特徴とする、請求項 6 に記載の方法。

【請求項 9】

前記三次元画像を描画するステップは、前記画素に対応するアルファ値を、前記画素の深度と少なくとも 1 つの周辺画素の深度との差に対応する値に設定することを特徴とする、請求項 8 に記載の方法。

40

【請求項 10】

前記画素は、前記画像において $P[x, y]$ であると指定され、前記三次元画像を描画するステップは、前記画素の深度を、 $P[x - 1, y]$ であると指定される第 1 画素と、 $P[x, y - 1]$ であると指定される第 2 画素の深度と比較することを特徴とする、請求項 6 に記載の方法。

【請求項 11】

画像に境界線を引く処理をコンピュータに実行させるためのプログラムを記録したコンピュータ読み取り可能な記憶媒体であって、

前記画像の各画素の深度値をメモリから読み出す読み出しステップと、

読み出された深度値を用いて各画素がオブジェクトの縁であるかどうかを判別する判別

50

ステップと、

縁上にある画素を境界線の色に変更する変更ステップとを前記コンピュータに実行させ

前記判別ステップにおいて、前記コンピュータは、画素の深度値と、その画素の周辺にある画素の深度値との差を算出し、算出された差に基づいてオブジェクトの縁であるかどうかを判別し、

前記変更ステップにおいて、前記コンピュータは、変更前の画像の色と、前記境界線の色とを前記差に応じた割合でブレンドすることによって、縁上にある画素を境界線の色に変更する、プログラムを記録した記憶媒体。

【請求項 12】

画像に境界線を引く処理を実行する画像処理装置であって、

前記画像の各画素の深度値をメモリから読み出す読み出し手段と、

読み出された深度値を用いて各画素がオブジェクトの縁であるかどうかを判別する判別手段と、

縁上にある画素を境界線の色に変更する変更手段とを備え、

前記判別手段は、画素の深度値と、その画素の周辺にある画素の深度値との差を算出し、算出された差に基づいてオブジェクトの縁であるかどうかを判別し、

前記変更手段は、変更前の画像の色と、前記境界線の色とを前記差に応じた割合でブレンドすることによって、縁上にある画素を境界線の色に変更する、画像処理装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、三次元コンピュータグラフィックスに関し、より特定的には、非写実的な三次元画像化に関する。さらに特定すれば、本発明は、コンピュータによる三次元対象物の陰影上の線およびその他その縁に見られる境界の生成・表示方法および装置に関する。

【0002】

【従来の技術】

コンピュータグラフィックスに関する研究の多くは、リアルな画像の生成に注目する傾向にあり、これまでかなりの成功を収めてきている。今では、コンピュータの作り出す画像は非常にリアルなものとなってきており、写真と見分けることができないほどである。たとえば、コンピュータによって作り出された特殊効果による、非常に写実的で本物であるかのような恐竜、異星人などを映画やテレビで見かけることも多い。新人のパイロットが訓練を受けているコンピュータによるフライトシミュレータは、非常にリアルで、実際の飛行状態をほぼ再現している。低価格な家庭用ビデオゲームシステムにおいても、驚くべきレベルのリアリズムが実現されており、ゲームのプレーヤは、たとえば、本物のレーサーでトラックを走行しているような、雪や氷に覆われたスロープをスキーで滑っているような、中世の城の中を歩いているような錯覚を覚えるほどである。大抵のゲームでは、このリアリズムによる錯覚がゲームプレイ体験を非常にすばらしいものになっている。

【0003】

しかし、場合によってはリアルでない画像の方が好ましいこともある。たとえば、対話型ビデオやコンピュータゲームなどにおいて、現実の（または空想上の）世界をリアルにシミュレートするのではなく、わざと非現実的なパロディー調の漫画キャラクタが、たくさん登場する漫画の世界を創造して表示すれば、娯楽的価値を持たせることができる。例えば、このようなゲームでは、動き、音声、および対話性をもたせながら、手描きの漫画本を再現するように努めている。

【0004】

このような三次元コンピュータシステムにおいて、望ましい視覚効果の1つとしては、表示対象物の陰影の線やその他その縁にみられる境界をはっきりと、確かなものにするのである。このような境界線によって、ある種の画像は鮮明になる。たとえば、漫画キャラクタの輪郭、風景の丘や山、壁の縁など、それぞれ異なる面をユーザがよりはっきり区別

10

20

30

40

50

しやすくなる。さらに、このような境界線によって、漫画家による手描きであるかのような望ましい印象が生まれやすくなる。

【0005】

【発明が解決しようとする課題】

ここで、キャラクタなどの対象物の縁周りの境界線のような、リアルでない効果を付与する方法の一つに、ある対象物を形成するポリゴンの縁に隣接した別の対象物と区別するラインをはっきり示す方法がある。

【0006】

しかし、別の対象物とのボーダをはっきり示すと、画像処理が極めて複雑化する。その結果、リソースの限られたシステムにおいて、この方法は画像生成の遅延や効率の低下を招く。対話型三次元ゲームシステムなどの低価格の三次元グラフィックスシステムにおいては、効率は極めて重要である。より効率のよい技術を用いて視覚効果を生成する方が、乏しいシステムリソースにかかる負担を軽減し、さらには、速度やその他の性能にそれほど影響を及ぼすことなく、視覚効果を総合的に向上させることができる。

【0007】

【課題を解決するための手段および発明の効果】

本発明は、家庭用ビデオゲームコンソールなどの、三次元グラフィックスシステムにおいて、陰影の線やその縁における境界線を表示するための効率的な技術を提供することによって、この問題を解決する。

【0008】

第1の発明によれば、画像がフレームバッファにおいて描画された後、境界線を生成する。本発明では、フレームバッファに記憶された数値を用いて、対象物の輪郭またはその他、縁にどの画素が位置しているかを求め、ボーダカラーをこれら画素に選択的にブレンドして表示する。

【0009】

また、或る実施例によれば、輪郭の縁は、ある画素の深度値とその周辺画素の深度値とを比較して配置される。たとえば、「距離」値が或る画素の深度とその周辺画素の深度との間の距離の絶対値を基に計算される。その後、この距離値を基に、望ましい境界線のカラーがその画素のカラー値にブレンドされる。たとえば、距離値は画素のアルファ値を計算するために用いられ、このアルファ値は画素のカラーにブレンドされるボーダカラーの量を制御するのに用いられる。

【0010】

別の例では、さらに深度変化を利用する。たとえば、距離値と、画素深度の関数であるさらに設けられた値とを基に画素のアルファ値を計算して、画素の深度値によって変化したアルファ値を求める。この深度によって変化したアルファ値を基に、ボーダカラーを画素カラーにブレンドする。

【0011】

さらなる実施例では、対象物の輪郭上ではなく、漫画家が境界線を引くであろう或る内部の縁の上に境界線を引く。例えば、漫画キャラクタが身体の前で腕を曲げている状態を考える。この場合、漫画家は、キャラクタの腕の輪郭の周りに境界線を引くと思われる。たとえ、このキャラクタが、腕の輪郭の縁が全体の輪郭の内部に位置するように胴体の前部で腕を曲げていても、である。本発明のさらなる特徴によれば、画像の画素にはそれぞれ異なる識別値が割り当てられる。この識別値は、典型的にはアルファ値のために用意されたフレームバッファメモリ領域等に格納される。これらの識別値は、境界線のある縁と、境界線が付与されない縁とを識別するのに用いられる。

【0012】

【発明の実施の形態】

以下、本発明における実施の形態について、図面を参照しながら説明する。図1は、本発明に係る、境界線を表示するための三次元ビデオグラフィックスシステム1005の例を示す。

10

20

30

40

50

【 0 0 1 3 】

システム 1 0 0 5 は、たとえば、ゲームコントローラ 1 0 0 7 および / または他の手入力装置からの対話型リアルタイム入力に応答し、表示装置 1 0 0 9 (たとえば、家庭用カラーテレビ受像機、ビデオモニタ、または他のディスプレイなど) に映像表示を行う。システム 1 0 0 5 は、外部記憶媒体 1 0 1 1 (たとえば、交換可能なビデオゲームカートリッジ、CD-ROM、または他の光ディスクなど) に記憶されたビデオゲームプログラムなどのコンピュータプログラムの制御によって動作する。

【 0 0 1 4 】

本例においては、システム 1 0 0 5 は、プロセッサ 1 0 1 0 , 三次元グラフィックスコプロセッサ 1 0 2 0 およびメモリ 1 0 3 0 を含む。プロセッサ 1 0 1 0 は、三次元グラフィックスコマンドをグラフィックスコプロセッサ 1 0 2 0 に与える。グラフィックスコプロセッサ 1 0 2 0 は、三次元グラフィックスコマンドに従って、三次元の世界における二次元ビューをインタラクティブに生成する。たとえば、グラフィックスコプロセッサ 1 0 2 0 は、ハードウェアによる三次元グラフィックスパイプラインを含む。このパイプラインは、三次元空間において規定されたポリゴンなどのグラフィックスプリミティブを操作して、任意に選ばれた視点に基づく視野面に投影された三次元空間の二次元画像を表す画素を生成する。本例において、ユーザは、ゲームコントローラ 1 0 0 7 を用いて、リアルタイムでインタラクティブに視点を変更することができる。

【 0 0 1 5 】

グラフィックスコプロセッサ 1 0 2 0 は、生成した表示画素を、メモリ装置 1 0 3 0 内のフレームバッファ 1 0 4 0 に記憶する。フレームバッファ 1 0 4 0 は、カラーフレームバッファ 1 0 4 2 と、深度 (Z) バッファ 1 0 4 4 とを備える。本例において、カラーフレームバッファ 1 0 4 2 は、赤、緑、青 (RGB) のカラー値の二次元アレイ、および対応するアルファ (A) 値を記憶する。記憶された RGB カラー値と表示装置 1 0 0 9 上に表示される画素との間是一对一の関係であってもよい。または、フレームバッファ 1 0 4 2 は、副サンプルを記憶してもよい。Z バッファ 1 0 4 4 は、カラーフレームバッファに記憶されている各画素または副画素についての深度値 (たとえば、視点に対する z 方向の距離) を記憶する。よく知られているように、Z バッファ 1 0 4 4 は、グラフィックスパイプラインが画像を「構築」する際に、さまざまな目的に用いられる (たとえば、見えない面の除去など)。

【 0 0 1 6 】

好ましい実施例において、カラーフレームバッファ 1 0 4 2 および / または z バッファ 1 0 4 4 の内容は、ボーダカラーを、カラーフレームバッファ 4 2 に記憶されている画素カラー値に選択的にブレンドして、対象物の境界線を規定する役割も果たす。具体的には、システム 1 0 0 5 は、画素フィルタ 5 0 を備えていても良い。画素フィルタ 5 0 は、三次元グラフィックスコプロセッサ 1 0 2 0 によって描画される画像の画素値を処理して、境界線カラーを表示する対象物の輪郭線、および / または他の縁の上に選択的に付与する。好ましい実施例における画素フィルタ 5 0 は、画像データがフレームバッファ 1 0 4 0 または他の画素メモリにおいて描画された後に、画素後処理の段階で動作する。たとえば、画素フィルタ 5 0 は、グラフィックスコプロセッサからの画像データと、メモリ 1 0 3 0 への画像データとの結合処理中に、三次元グラフィックスコプロセッサ 1 0 2 0 で描画されてバッファされた画像データに、境界線などの非写実的な画像要素を付与してもよい。

【 0 0 1 7 】

図 2 は、本発明を実施しうるインタラクティブ三次元コンピュータグラフィックスシステム 1 0 0 5 全体の詳細な模式図である。システム 1 0 0 5 は、魅力的なステレオ音声を伴うインタラクティブ三次元ビデオゲームを行うために用いることができる。光ディスク 1 0 1 1 などの適切な記憶媒体を光ディスク再生装置 1 1 3 4 に挿入することによって、いろいろなゲームを行うことができる。ゲームを行う者は、手動コントローラ 1 0 0 7 などの入力装置を操作することによって、システム 1 0 0 5 とリアルタイムに対話することができる。このコントローラ 1 0 0 7 は、ジョイスティック、ボタン、スイッチ、キー

10

20

30

40

50

ボード、キーパッドなど、さまざまな操作器が含まれる。

【0018】

本例において、メインプロセッサ1010が、手動コントローラ1007（および/または他の入力装置）からの入力を受ける。メインプロセッサ1010は、インタラクティブにこのようなユーザ入力に回答し、たとえば外部記憶装置1011から提供される、ビデオゲームや他のグラフィックスプログラムを実行する。たとえば、メインプロセッサ1010は、衝突検出や動画処理のほかにも、様々なリアルタイムでインタラクティブな制御機能を行うことが可能である。

【0019】

メインプロセッサ1010は、三次元グラフィックスコマンドや音声コマンドを生成し、これらをグラフィックスおよび音声コプロセッサ1020に送信する。グラフィックスおよび音声コプロセッサ1020は、これらのコマンドを処理して、魅力的な画像を表示装置1009に表示し、ステレオ音声をステレオスピーカ1137R、1137L、または適切な音声生成装置において生成する。

10

【0020】

システム1005は、TVエンコーダ1140を含む。TVエンコーダは、画像信号をコプロセッサ1020から受信し、標準的な表示装置1009（たとえば、コンピュータモニタや家庭用カラーテレビ受像機）での表示に適するように合成ビデオ信号に変換する。また、システム1005は、オーディオコーデック（圧縮器/伸長器）1138を含む。

【0021】

オーディオコーデック1138は、デジタル化された音声信号を圧縮/伸長する（さらに、デジタルアナログ音声信号形式間の変換を行ってもよい）。オーディオコーデック1138は、バッファ1141を介して音声入力を受信し、コプロセッサ1020に送信する（コプロセッサ1020における処理には、コプロセッサが生成する他の音声信号、および/または、光ディスク装置1134のストリーミング音声出力を介して受信した他の音声信号とのミキシングなどがある）。

20

【0022】

コプロセッサ1020は、音声関連情報を音声タスク専用のメモリ1144に記憶させる。コプロセッサ1020は、処理後の音声出力信号をオーディオコーデック1138に与えて、圧縮および（たとえば、バッファ増幅器1142Lおよび1142Rを介して）アナログ信号への変換を行い、スピーカ1137Lおよび1137Rによって再生できるようにする。

30

【0023】

コプロセッサ1020は、システム1005内の様々な周辺機器と通信することができる。たとえば、パラレルデジタルバス1146は、光ディスク装置1134との通信に用いられてもよい。シリアル周辺バス1148は、様々な周辺機器との通信に用いられてもよい。この周辺機器の例としては、ROMおよび/またはリアルタイムクロック1150、モデム1152、フラッシュメモリ1154などが挙げられる。さらなる外部シリアルバス1156は、拡張メモリ1158（たとえば、メモリカード）との通信に用いられてもよい。

40

【0024】

図3は、コプロセッサ1020内の構成要素例のブロック図である。コプロセッサ1020は、1つの集積回路であって、三次元グラフィックスプロセッサ1107と、プロセッサインターフェース1108と、メモリアンターフェース1110と、オーディオデジタル信号プロセッサ（DSP）1162と、オーディオメモリアンターフェース（I/F）1164と、オーディオインターフェースおよびミキサ1166と、周辺制御器1168と、表示制御器1128とを含む。

【0025】

三次元グラフィックスプロセッサ1107は、グラフィックス処理タスクを行い、オーディオデジタル信号プロセッサ1162は、オーディオ処理タスクを行う。表示制御器1

50

128は、メモリ1030から画像情報をアクセスして取り出し、表示装置1009に表示するためにTVエンコーダ1140へ与える。オーディオインターフェースおよびミキサ1166は、オーディオコーデック1138とのインターフェースとなり、発生源の異なる音声(たとえば、ディスク1011からのストリーミングオーディオ入力、オーディオDSP1162の出力、オーディオコーデック1138を介して受信した外部オーディオ入力など)を混合することも可能である。プロセッサインターフェース1108は、メインプロセッサ1010およびコプロセッサ1020間のデータや制御に関するインターフェースとなる。メモリインターフェース1110は、コプロセッサ1020およびメモリ1030間のデータや制御に関するインターフェースとなる。本例において、メインプロセッサ1010は、プロセッサインターフェース1108とコプロセッサ1020の一部であるメモリインターフェース1110とを介して、メインメモリ1108にアクセスする。周辺制御器1168は、コプロセッサ1020および上述の様々な周辺機器(たとえば、光ディスク装置1134、コントローラ1007、ROMおよび/またはリアルタイムクロック1150、モデム1152、フラッシュメモリ1154、およびメモリカード1158)間のデータや制御に関するインターフェースとなる。オーディオメモリインターフェース1164は、オーディオメモリ1144とのインターフェースを提供する。

【0026】

図4は、三次元グラフィックスプロセッサ1107と、それに関連するコプロセッサ1020内の構成要素のより詳細な図である。三次元グラフィックスプロセッサ1107は、コマンドプロセッサ1114と、三次元グラフィックスパイプライン1116を含む。メインプロセッサ1010は、グラフィックスデータのストリームをコマンドプロセッサ1114とやり取りする。コマンドプロセッサ1114は、このような表示コマンドを受信し、(メモリ1030から処理に必要な追加のデータを取得して)解析を行い、三次元処理および描画のために頂点コマンドのストリームをグラフィックスパイプライン1116に与える。グラフィックスパイプライン1116は、これらのコマンドに基づいた三次元画像を生成する。処理後の画像情報は、メインメモリ1030に転送され、表示制御器1128によってアクセスできるようにしてもよい。表示制御器1128は、表示装置1009にパイプライン1116のフレームバッファ出力を表示する。メモリ要求アービトレーション回路130は、グラフィックスパイプライン1116と、コマンドプロセッサ1114と、表示部128とにおけるメモリアccessを調停する。

【0027】

図4に示すように、グラフィックスパイプライン1116は、変換部1118と、セットアップ/ラスタライザ1120と、テクスチャ部1122と、テクスチャ環境部1124と、画素エンジン1126とを含んでもよい。グラフィックスパイプライン1116において、変換部1118は、様々な三次元変換動作を行い、さらに、照明効果やテクスチャ効果を行ってもよい。変換部1118の行う処理は、たとえば、頂点毎に入力される形状のオブジェクト空間からスクリーン空間への変換、入力されるテクスチャ座標の変換、射影テクスチャ座標の計算、ポリゴンクリッピング、頂点毎の照明度計算、バンプマッピングテクスチャ座標の生成などがある。セットアップ/ラスタライザ1120は、セットアップ部を含む。セットアップ部は、変換部1118から頂点データを受信し、三角形のセットアップ情報を複数のラスタライザに送信する。ラスタライザは、縁のラスタ化、テクスチャ座標のラスタ化、カラーのラスタ化を行う。テクスチャ部1122は、テクスチャリングに関連して様々なタスクを行う。このタスクには、マルチテクスチャ処理、キャッシュ後のテクスチャ伸長処理、テクスチャフィルタリング、浮き出しバンプマッピング、射影テクスチャを用いた陰影付け、およびアルファ透明度および深度を用いたBLITなどがある。テクスチャ部1122は、フィルタリングされたテクスチャ値をテクスチャ環境部1124に出力する。テクスチャ環境部1124は、ポリゴンカラーとテクスチャカラーとをブレンドし、テクスチャフォグやその他の環境関連機能を行う。画素エンジン1126は、zバッファリングやブレンドを行い、データを集積回路上のフレームバッファメモリに記憶する。

10

20

30

40

50

【 0 0 2 8 】

図5に示すように、グラフィックスパイプライン1116は、フレームバッファ情報を局所的に記憶するDRAMメモリ1126aを搭載してもよい。集積回路上のフレームバッファ1126aは、集積回路上には備えられていないメインメモリ1030に定期的に書き込まれ、集積回路上の表示部1128がアクセスできるようにする。たとえば、画素フィルタ50は、この書き込み処理中に境界線などの輪郭を漫画に付与するよう動作する。グラフィックスパイプライン1116のフレームバッファ出力（最終的にはメインメモリ1030に記憶される）は、表示部1128によってフレーム毎に読み出される。表示部1128は、表示装置1009における表示のために、デジタルRGB画素値を提供する。

10

【 0 0 2 9 】

上述したビデオゲームシステムの構成要素の或るものは、上述の家庭用ビデオゲームコンソール構成以外のものとして実現することも可能である。一般的に、家庭用ビデオゲームソフトウェアは、特定の家庭用ビデオゲームシステム上で実行されるように記述されている。このビデオゲームソフトウェアは、異なるシステムのハードウェアやソフトウェア構成を考慮に入れて、その異なるシステムに「移植される」（変換される）ことも、場合によっては可能である。このような「移植」には、ビデオゲームのソースコードにアクセスする必要があることが多い。この他に、構成の異なるシステムにおいてゲームソフトウェアを実行させるための方法としては、第2のシステムに第1のシステムをエミュレートさせる方法がある。第2のシステムが第1のシステムのハードウェアおよびソフトウェア資源をうまくエミュレートまたはシミュレートすることができれば、第2のシステムはビデオゲームソフトウェアのバイナリで実行可能な画像をうまく実行することができ、ゲームソフトウェアのソースコードにアクセスする必要はなくなる。

20

【 0 0 3 0 】

家庭用ビデオゲームにおいて、エミュレータは、ゲームプログラム用のシステム、すなわちゲームプログラムを実行することができるように設計されているシステムとは異なるシステムである。一例として、エミュレータシステムは、ゲームソフトウェア用のシステムのものとは異なるハードウェアおよび/またはソフトウェア構成（プラットフォーム）を提供する。エミュレータシステムは、ゲームソフトウェア用のシステムのソフトウェアおよび/またはハードウェア構成要素を模倣するソフトウェアおよび/またはハードウェア構成要素を含む。たとえば、エミュレータシステムは、当該ゲームソフトウェア用システムのハードウェアおよび/またはファームウェアをシミュレートするソフトウェアエミュレータプログラムを実行するパーソナルコンピュータなどの汎用デジタルコンピュータを備えてもよい。

30

【 0 0 3 1 】

エミュレータソフトウェアは、コンソールを用いた図1ないし図4に示すような家庭用ビデオゲームシステム用のゲームをパーソナルコンピュータやその他の汎用デジタルコンピュータ上で実行できるように開発されてもよい。このような汎用デジタルコンピュータ（たとえば、IBMやマッキントッシュのパーソナルコンピュータやその互換機）には、DirectXや他の標準三次元グラフィックスコマンドアプリケーションプログラムインターフェース（API）に対応する三次元グラフィックスパイプラインを提供する三次元グラフィックスカードを搭載しているものがある。また、サウンドコマンドの標準セットに基づいて、高品質のステレオサウンドを提供するステレオサウンドカードを搭載しているものもある。エミュレータソフトウェアを実行するこのようなマルチメディアハードウェア搭載パーソナルコンピュータは、専用の家庭用ビデオゲームコンソールハードウェア構成におけるグラフィックス性能やサウンド性能に近づくのに十分な性能を有することもある。エミュレータソフトウェアは、パーソナルコンピュータプラットフォーム上のハードウェア資源を制御して、家庭用ビデオゲームコンソールプラットフォームが行う処理（このプラットフォームのためにゲームプログラムはゲームソフトウェアを書いたのであるが）、三次元グラフィックス、サウンド、周辺機器、および他の機能をシミュレート

40

50

する。

【0032】

図6は、エミュレーション処理全体の一例を示す。本エミュレーション処理の例は、ホストプラットフォーム1201と、エミュレータ要素1303と、ROMまたは光ディスク1305などの記憶媒体や他の記憶装置上のバイナリ画像を実行可能なゲームソフトウェアとを用いる。ホスト1201は、たとえばパーソナルコンピュータや他の型のゲームコンソールのような汎用または専用のデジタルコンピューティング装置であってもよい。

【0033】

エミュレータ1303は、ホストプラットフォーム1201において実行されて、記憶媒体1305からのコマンド、データおよび他の情報をホスト1201によって処理可能な形式へリアルタイムで変換する。たとえば、エミュレータ1303は、図1ないし図4に示すような家庭用ビデオゲームプラットフォーム用のプログラム命令を取り出し、このプログラム命令をホスト1201が実行または処理できるような形式に変換する。一例として、ゲームプログラムがZ-80、MIPS、IBM PowerPCを用いたプラットフォームまたは他の特定の処理装置上で実行されるように書かれたものであって、かつホスト1201が異なる(たとえば、インテル社)の処理装置を用いたパーソナルコンピュータである場合に、エミュレータ1303は、記憶媒体1305から1つまたは一連のプログラム命令を取り出し、1つまたはそれ以上のインテル社のプログラム命令に変換する。

10

【0034】

同様に、エミュレータ1303は、図3に示すグラフィックスおよびオーディオコプロセッサによる処理用のグラフィックスコマンドやオーディオコマンドを取り出し、ホスト1201において利用可能なグラフィックスハードウェアおよび/またはソフトウェアならびにオーディオ処理資源が処理できるような形式に変換する。一例として、エミュレータ1303は、このようなコマンドを、ホスト1201の特定のグラフィックスおよび/またはサウンドカードが(たとえば、標準のDirectXやサウンドアプリケーションプログラムインターフェース(API)を用いて)処理できるようなコマンドに変換してもよい。

20

【0035】

上述のようなビデオゲームシステムの機能の一部または全部を提供するために用いられるエミュレータ1303は、エミュレータを用いて実行されるゲーム用の様々な選択肢や画面モードの選択を簡略化または自動化するグラフィックユーザインターフェース(GUI)を備えていてもよい。一例として、このようなエミュレータ1303は、当該ビデオゲームソフトウェアが元来想定していたホストプラットフォームに比して高い機能性をさらに有していてもよい。

30

【0036】

図7は、エミュレータ1303と共に用いるのに適した、パーソナルコンピュータに基づくホスト1201を示す。パーソナルコンピュータシステム1201は、処理部1203と、システムメモリ1205とを含む。システムバス1207は、システムメモリ1205などの様々なシステム要素と、処理部1203とを接続する。システムバス1207は、メモリバスまたはメモリコントローラ、周辺バス、ローカルバスなど、数種のバス構造(各々のバス構造には様々なバスアーキテクチャのうちのいずれかが用いられる)のいずれであってもよい。

40

【0037】

システムメモリ1207は、読み出し専用メモリ(ROM)1252と、ランダムアクセスメモリ(RAM)1254とを含む。ROM1252には、ベーシック入出力システム(BIOS)1256が記憶されており、これには、立ち上げ時などにパーソナルコンピュータシステム1201内の要素間の情報伝達を助ける基本ルーチンが含まれている。

【0038】

パーソナルコンピュータシステム1201は、様々なドライブおよび関連するコンピュー

50

タが読み出し可能な媒体をさらに含む。ハードディスクドライブ1209は、(典型的には固定の)磁気ハードドライブ1211に対して、読み出しや書き込みを行う。磁気ディスクドライブ1213は、着脱可能な「フロッピー」や他の磁気ディスク1215に対して、読み出しや書き込みを行う。光ディスクドライブ1217は、CD-ROMや他の光学媒体などの着脱可能な光ディスク1219に対して読み出しを行い、構成によっては書き込みも行う。ハードディスクドライブ1209、磁気ディスクドライブ1213、および光ディスクドライブ1217は、それぞれ、ハードディスクドライブインターフェース1221、磁気ディスクドライブインターフェース1223、および光ドライブインターフェース1225によって、システムバス1207に接続されている。これらドライブおよび関連するコンピュータが読み出し可能な媒体は、コンピュータが読み出し可能な命令、データ構造、プログラムモジュール、ゲームプログラム、およびパーソナルコンピュータシステム1201のための他のデータを不揮発的に記憶する。

10

【0039】

また、他の構成においては、コンピュータがアクセス可能なデータを保存することができる、コンピュータが読み出し可能な他の媒体(たとえば、磁気カセット、フラッシュメモ리카ード、デジタルビデオディスク、ベルヌーイ(b e r n o u l l i)カートリッジ、ランダムアクセスメモリ(RAM)、読み出し専用メモリ(ROM)など)が用いられてもよい。

【0040】

エミュレータ1303を含む多くのプログラムモジュールが、システムメモリ1205のハードディスク1211、着脱可能な磁気ディスク1215、光ディスク1219および/またはROM1252および/またはRAM1254に記憶されてもよい。このようなプログラムモジュールは、グラフィックスおよびサウンドAPI、1つ以上のアプリケーションプログラム、他のプログラムモジュール、プログラムデータ、およびゲームデータを提供するオペレーティングシステムを含んでもよい。ユーザは、キーボード1227やポインティングデバイス1229などの入力装置を用いて、パーソナルコンピュータシステム1201に対して、コマンドや情報を入力する。他の入力装置としては、マイクロフォン、ジョイスティック、ゲームコントローラ、衛星放送用アンテナ、スキャナなどがある。このような入力装置は、システムバス1207に接続しているシリアルポートインターフェース1231を介して、処理装置1203に接続される場合が多いが、たとえばパラレルポート、ゲームポート、ユニバーサルシリアルバス(USB)などの他のインターフェースによって接続されてもよい。また、システムバス1207には、モニタ1233や他の型の表示装置が、ビデオアダプタ1235などのインターフェースを介して接続されている。

20

30

【0041】

パーソナルコンピュータ1201は、インターネットなどのワイドエリアネットワーク1152と通信を行うためのモデム1154などの手段をさらに含んでもよい。モデム1154は、内蔵でも外付けでもよく、シリアルポートインターフェース1231を介してシステムバス1207に接続される。パーソナルコンピュータ1201は、ネットワークインターフェース1156をさらに備えていてもよく、これによって遠隔計算装置1150(たとえば、他のパーソナルコンピュータ)とローカルエリアネットワーク1158を介して通信することができる(このような通信は、ワイドエリアネットワーク1152、または他の通信路、たとえばダイアルアップ方式や他の通信手段などを介して行われてもよい)。パーソナルコンピュータシステム1201は、典型的には、プリンタなどの標準周辺機器のような他の周辺出力装置を含む。

40

【0042】

たとえば、ビデオアダプタ1235は、三次元グラフィックスパイプラインチップセットを含んでもよい。このチップセットは、マイクロソフト社のDirectXのような標準三次元グラフィックスアプリケーションプログラマインターフェースに基づいて出された三次元グラフィックスコマンドに回答して、三次元グラフィックス描画を高速に行う。さ

50

らに、一連のステレオスピーカ1237が、システムバス1207に接続されており、この接続は、バス1207によって与えられるサウンドコマンドに基づく高品質ステレオサウンド生成に対応したハードウェアおよび内蔵ソフトウェアを提供する従来の「サウンドカード」などのサウンド生成インターフェースを介して行われる。このようなハードウェアの性能により、ホスト1201は、記憶媒体1305に記憶されたビデオゲームを行うのに十分な速度のグラフィックスおよびサウンド性能を提供することができる。

【0043】

(本発明による非写実的な漫画的な輪郭線処理法の例)

本発明による漫画の輪郭線/境界線処理は、ハードウェアおよび/またはソフトウェアで有利に実行され得る。時間が充分にあると仮定すると、グラフィックスコプロセッサ1020が画像をフレームバッファにおいて描画した後であってその画像が表示される前に、プロセッサ1010は、フレームバッファ1040にアクセスして画素後処理を行い、境界線カラーをカラーフレームバッファ1042にブレンドする。あるいは、グラフィックスコプロセッサ1020内にハードウェアおよび/またはソフトウェアを備えて、この機能を行わせる。上述のように、コプロセッサ1020は、グラフィックスパイプライン内のアルファブレンディングおよびzバッファ演算回路などの既存のハードウェアサポートを利用して、画素後処理/フィルタリングを行い、対象物の縁周りに境界線を付与してもよい。あるいは、プロセッサ1010および/または表示部1128が、フレームバッファ1040の内容を基に画素フィルタリング処理を行うこともできる。

【0044】

コプロセッサ1020がこの画素後処理を完了すると、変更された画素値はフレームバッファメモリ1030に書き戻される。このフレームバッファメモリ1030から、コプロセッサ内のビデオ生成回路によって内容がアクセスされて、表示装置1009上に映像が生成される。あるいは、変更された値は他の場所へ(たとえば、直接、表示装置1005へ)送られる。この伝送効率を高めるために、フレームバッファ1040のデータをダブルバッファしてもよく、DMAを使用してもよい。フレームバッファ1040が1ラインずつ読み出され、かつ処理される場合、前に位置するラインのZ値を保持するためにラインバッファを使用してもよい。

【0045】

図8は、好ましい実施例に係る画素フィルタリングルーチン100の一例を示すフローチャートであって、画素フィルタ50に対して画素後処理境界線を行う。図8の画素フィルタルーチン100は、フレームバッファ1040(ブロック102)内に記憶された画素 $[x][y]$ のそれぞれに対して行われる。この画素フィルタ100は画素のカラー値(画素 $R[x][y]$ 、画素 $G[x][y]$ 、画素 $B[x][y]$)を読み出し、さらに画素の深度値($Z[x][y]$)を読み出す(ブロック104)。加えて、この画素フィルタルーチン100は、画素 $[x][y]$ 周辺にある画素の深度値も読み出す(同じくブロック104)。本例において、画素フィルタルーチン100は隣接画素2つの深度値を読み出す。つまり、 $x-y$ アレイ内の所定の画素の「左」隣にある周辺画素における深度値 $Z[x][y]$ 、および $x-y$ アレイ内の所定の画素の真「下」にある周辺画素における深度値 $Z[x][y-1]$ である。以上は、図9を参照のこと。フレームバッファ1040が1ラインずつ読み出され、かつ処理された場合、ラインバッファは前に位置するラインのZ値を保持するのに使用され得る。

【0046】

画素フィルタルーチン100は、たとえば以下のDz演算(ブロック106参照)によって、画素 $[x][y]$ が対象物の縁であるかどうかを判別する、たとえば、画素の深度値($Z[x][y]$)と1つ以上の周辺の(本例においては隣接する)画素の深度値との差を判別する。

$$\begin{aligned} D_z D_x &= |Z[x][y] - Z[x-1][y]| \\ D_z D_y &= |Z[x][y] - Z[x][y-1]| \\ D_z &= \max(D_z D_x, D_z D_y) \end{aligned}$$

10

20

30

40

50

上記演算を行う際、画素フィルタ100は画素の深度と、2つの隣接画素それぞれとの差を判別する、つまりこれらの差の絶対値をとり（ある対象物は別の対象に近いか遠いかのどちらかであることを利用する）、得られた2つの絶対値のうち大きい方を選択する。この結果が距離値Dzであって、ある画素がz方向において、ある周辺画素からどれだけ離れているかをはかる値である。この演算は、画素が対象物の縁上にあるかどうかをテストするために行われる。たとえば、画素が対象物の縁上にある場合、Dzは一般に大きくなり（典型的に、周辺画素はそれぞれ異なる深度を有しているため）、画素が対象物の縁上にない場合は、Dzは一般に小さくなる（周辺画素は深度が相似するためである）。

【0047】

別の例においては、2つの周辺画素の距離値の総計をとることで距離値Dzを演算することができる。たとえば、下式が例である。

$$Dz = Dz Dx + Dz Dy$$

また、その他の方法でも可能である（たとえば、演算方法を変えたり、使用する周辺画素を変更したり、または周辺画素数を変更するなど）。

【0048】

本例における画素フィルタルーチン100は、上述のように演算された距離値Dzに比例してカラーを変化させる。たとえば、画素フィルタ100は、得られた距離値Dzの係数をスケール/修正して変更し、その結果得られた値をクランプして、たとえば以下のような画素ブレンドファクタを成立させる（ブロック108参照）。

$$\text{アルファ} = \text{クランプ}(\text{スケール係数} * Dz + \text{ベース係数}, 0, 1)$$

ここで、クランプは、クランプ(x, 0, 1)とした場合において、xが0以下のときには0を返し、xが1以上のときには1を返し、 $0 < x < 1$ のときにはxを返す関数である。

【0049】

スケール係数が正の値であると仮定すると、得られたアルファ値はDzが大きいと大きくなり（たとえば、ほぼ1、または1になるようにクランプされる）、Dzが小さいとアルファ値も小さくなる（たとえば、ほぼ0、または0になるようにクランプされる）。画素ブレンドファクタを成立させるには、方法は問わない。たとえば、画素ブレンドファクタを演算する別の方法を以下に示す。

$$\text{グレイ} = \text{クランプ}((Dz - \text{係数} A) * \text{係数} B, 0, 1)$$

【0050】

本例における画素フィルタ100は、ブレンド処理（たとえば、アルファブレンディング）で得られた画素ブレンド係数を用いて、選択的に所定の境界線カラー（たとえばラインR、ラインG、ラインB）を、カラーフレームバッファ1042から得られた画素のカラー値（画素R[x][y]、画素G[x][y]、画素B[x][y]）とブレンドする。このブレンドを行うための演算の例は、以下の通りである。

$$\text{新画素} \cdot R[x][y] = \text{旧画素} \cdot R[x][y] * (1 - \text{アルファ}) + \text{ライン} \cdot R * \text{アルファ}$$

$$\text{新画素} \cdot G[x][y] = \text{旧画素} \cdot R[x][y] * (1 - \text{アルファ}) + \text{ライン} \cdot G * \text{アルファ}$$

$$\text{新画素} \cdot B[x][y] = \text{旧画素} \cdot R[x][y] * (1 - \text{アルファ}) + \text{ライン} \cdot B * \text{アルファ}$$

これら演算に従って、ブレンドされた（「新」）画素カラー値は、距離値Dzが小さいと、ほとんどの場合、初期値になり、距離値Dzが大きいと境界線カラーになる。

【0051】

上記のブレンド演算は、境界線カラーを黒（たとえばライン・R = 0、ライン・G = 0、ライン・B = 0）に限定し、および/またはホワイトキャンパスのカラー（ライン・R = 255、ライン・G = 255、ライン・B = 255）だけを用いることで簡略化が可能である（たとえば、さらなる最適化のために行われる）。より最適化するためには、RGBカラーでないフォーマット、たとえばYUV（この場合、境界線が鮮明に見えるよう、確

10

20

30

40

50

実に付与するために明るさYを制御する)などを用いる。

【0052】

ブレンド後、ブレンドされた画素をバッファして表示装置1009(ブロック112)に表示する。たとえば、画素フィルタルーチン100を、フレームバッファ1040内の画素すべてに対して行い、その結果得られるブレンドされたフレームバッファの内容を表示装置1009に表示する。

【0053】

遠近処理は、場合によっては上記処理で得られる結果に影響を及ぼす。図10に示すように、遠くに画素P i x A、近くに画素P i x Bがある場合について考える。変換により、遠くにある画素P i x AのD z D yは、近くにある画素P i x BのD z D yよりも大きくなる。この結果、未処理のD z D xおよびD z D y値を用いた画像全体における、境界線の検知がさらに難しくなる。この問題を克服するためには、Zレンジに応じてZ値に修正を加えるとよい。修正には、たとえばレンジ変換するため、以下のように $\log_2(n)$ 関数を用いる。

$$D z D x = | \log_2 (Z[x][y]) - \log_2 (Z[x - 1][y]) |$$

$$D z D y = | \log_2 (Z[x][y]) - \log_2 (Z[x][y - 1]) |$$

プロセッサ1010に浮動小数点計算機能があり、Z値が例えば以下のような浮動小数点フォーマットで記憶されている場合、 $\log_2(n)$ 関数の計算は非常に都合がよい。

[3ビット指数][11ビット仮数]

【0054】

図11は、図8に示すフローチャートを変更して、この $\log_2(n)$ のレンジ変換式を加えたものである。

【0055】

図12は、ブレンディングを制御するのに用いられる、アルファ値の深度変調を行う後処理画素フィルタ200のもう一つの例である。図12に示す、ブロック204、206、および208は、図8(レンジ修正が適用される場合は図11)に示す対応するブロック104、106、および108と同一である。図12のブロック210は、ブロック208で計算された値、および画素深度zの関数であるさらなる値の関数(たとえば、積)としてアルファAを計算するために付加されたものである。たとえば、

$$\text{アルファ} = \text{アルファ} D z * \text{アルファ} Z$$

この結果得られたアルファ値、つまり図8のブロック210に関連して説明した、ブレンディングの制御(ブロック212参照)に用いられるアルファ値も、画素の深度に左右される。

【0058】

(漫画的な輪郭線処理のさらなる実施例)

演算方法によっては、上記境界線の応用アルゴリズムでは、漫画的な輪郭線処理において、満足のいく結果が得られないことがある。図13においてそのような例を示す。図13における漫画キャラクタ300には、上記のように陰影の縁302に境界線が付与されている。図13から、この漫画キャラクタ300は右手、手首を有しており、腕を前部で曲げている様子がわかる。上記の手法では、場合によって(キャラクタの腕が胴体部からどれだけ離れているかによるが)、右の手、手首、および腕の周りの縁は陰影の縁ではなく内部だと判断されるため、それらの箇所には境界線を付与しない。図13では、漫画的な輪郭線処理がキャラクタの陰影の縁302だけに施された場合、漫画キャラクタ300が部分的に消滅する、あるいはやや鮮明でなくなる様子を示す。ところが、ユーザは(塗り絵、手描きのアニメ漫画、および/または漫画本などの記憶から)手、手首および腕がはっきり示されるような境界線を期待している。

【0059】

漫画キャラクタ300が手描きであるかのように見せるには、内部の縁304にも境界線を付すとよい。本例においては、内部の縁は、キャラクタが自身の前部で曲げている手、手首および腕を規定している。図14では、このキャラクタ300の内部の縁304に

10

20

30

40

50

境界線が付与されている。この内部の縁304は、キャラクタ300が腕を外に向けて延ばしている場合は輪郭線の縁になるが、図14に示す腕の位置では内部の縁になる。

【0060】

本発明に係る別の局面によれば、図14に示すような内部の縁においては、漫画的な輪郭線処理は自動的に行われる。つまり、対象物の部分によって、画素にはそれぞれ異なる識別値が付与される。たとえば、通常はアルファ情報をエンコードするのに用いられるフレームバッファ1040および/またはDRAM1126a内に、ビット割り当てを行うことにより、この識別値を指定する。この指定された識別値を用いて、その画素の位置に境界線を付すかどうかを決定してもよい。たとえば、ある画素の識別値と、その画素の近辺にある（隣接した）画素の識別値とを比較する。これら2つの画素の識別値間にある所定の関係が認められた場合、境界線は付与されない。たとえば、識別値が同一の場合、これら2つの画素は同一面上にあるため、境界線は付さない。しかし、これら2つの画素の識別値が、別の所定の関係にある場合は、境界線を付与する。

10

【0061】

図15(A)~(C)では一例を示す。図15(A)は、3つの部分320、322および324に大別される対象物319の斜視図である。図15(B)は、同じ対象物319の平面図である。対象部分320は四角形であり、対象部分322は円形であり、対象部分324は円錐である。ここで、グラフィックスアーティストが、円錐324が四角形320に視覚的に接触するが、円形322には接触しない部分、または円形が四角形に接触する部分に境界線330(図15(A)参照)を引こうとしていると仮定する。この場合、四角形320、円形322および円錐324内の画素は、それぞれ異なる識別値でコーディングされる。たとえば、四角形320内の画素は識別値「1」でコーディングされ、円形322内の画素は識別値「2」でコーディングされ、さらに円錐324内の画素は識別値「3」でコーディングされる。図15(C)は、コーディング情報を記憶するフレームバッファ1040および/または1126aのアルファ部の一例を示す(斜線を付したセルが境界線カラーを付与するセルである)。

20

【0062】

画素後処理の段階においては、フレームバッファ内の様々な識別値に対してテストを行う。隣接する画素のように同一の識別値を有する画素(このような画素はすべて同一面上に存在する)に対しては、境界線を付与しない。また、或る画素の識別値が隣接画素のそれと、ある基準を下回って異なっている場合は境界線を付与しない(たとえば、画素kの識別値が画素k+1の識別値から2以上変わらない場合、境界線は付与しない)。しかし、ある画素の識別値が隣接画素のそれと、ある基準を上回って異なっている場合は境界線を付与する(たとえば、画素kの識別値が画素k+1の識別値から2以上違う場合、境界線を画素kに付与する)。

30

【0063】

図16は、本実施例において、境界線を付与するための画素後処理ルーチンの一例を示すフローチャートである。ルーチン350は、ループ(ブロック352-362)を含み、このループはフレームバッファ1040および/または1126aに記憶される画像における画素[i][j]それぞれに対して行われる。上述のように、画像生成処理においては、フレームバッファで画像を描画する際、通常はアルファ値を記憶するためにビットが割り当てられているフレームバッファ内にある対象物の部分毎に識別値を設定する。ルーチン350において、これらアルファ値(ここではID値)をテストして境界線を引くかどうかを決定する。本例において、ルーチン350では、画素[i][j]のアルファ(ID)値および隣接画素(たとえば、画素[i-1][j]および画素[i][j-1])のアルファ(ID)値を検索する(ブロック352)。その後、ルーチン352では、以下の演算を行い(ブロック354)画素[i][j]のアルファ(ID)値と隣接画素のアルファ(ID)値との差を求める。

40

$$diffX = |(\text{アルファ}[i][j] - \text{アルファ}[i-1][j])|$$

$$diffY = |(\text{アルファ}[i][j] - \text{アルファ}[i][j-1])|$$

50

【 0 0 6 4 】

その後ルーチン 3 5 0 では、演算して得られた差の値、 $d i f f X$ と $d i f f Y$ とをテストして、どちらかが所定の差の値を超えているかどうかを判別する（たとえば任意に固定の、または 1 のようなプログラム可能閾値）（ブロック 3 5 6）。差の値のうち、少なくとも 1 つが所定の差の値を超えている場合、ルーチン 3 5 0 は画素 $[i][j]$ のカラーを境界線カラーに設定する（ブロック 3 5 8）。これにより本例においては、アルファスロープが - 1 から + 1 の場合、画素は同一面上にあるとされる。ステップ 3 5 2 から 3 5 8 は、画像内にある画素それぞれに対して順に行われる（ブロック 3 6 0、3 6 2）。

【 0 0 6 5 】

ルーチン 3 5 0 の変形例として、対象物のある特別なアルファ識別値（たとえば 0×0 ）でコーディングして、境界線を付与するために無視すべき対象物内の画素を特定する（図 1 7 参照）。境界線が付与されていない対象物をビットマップとして描画する際に有用である（例えば、爆発のアニメーション等）。

【 0 0 6 6 】

図 1 8 は、上記ルーチン 3 5 0 を用いて、図 1 3 および 図 1 4 に示す対象物 3 0 0 にかに効率良く境界線を引くかを示す。本例において、対象物は部分毎にそれぞれ異なるアルファ（ID）値でコーディングされている。たとえば、対象物 3 0 0 は 2 本の腕 3 1 1 a、3 1 1 b および胴体 3 0 9 を備える。腕はそれぞれ手 3 1 3、手首部 3 1 5、下腕部 3 1 2、肘部 3 0 8、上腕部 3 1 0、および肩部 3 1 7 を備える。これら部位は、以下のようにそれぞれ異なるアルファ ID でコーディングされる。

身体部 アルファ ID

左手 3 1 3 a 1

左手首 3 1 5 a 2

左下腕 3 1 2 a 3

左肘 3 0 8 a 4

左上腕 3 1 0 a 5

左肩 3 1 7 a 6

胴体 3 0 9 7

右肩 3 1 7 b 8

右上腕 3 1 0 b 9

右肘 3 1 1 b 1 0

右下腕 3 1 2 b 1 1

右手首 3 1 5 b 1 2

右手 3 1 3 b 1 3

上記アルファ ID コーディングの例によれば、ルーチン 3 5 0 では、図 1 8 に太線で示されているような境界線を引く。しかし、部位の区切り部分（点線で示す）には境界線を引かない。

【 0 0 6 7 】

上記コーディングは、同じ対象物 3 0 0 の部位の区切り部分に境界線を引くのにも用いることができる。従来の塗り絵や手描きアニメ漫画などでは、このような区切り部分にも漫画的な輪郭線処理を施して、さらに関節を規定したり、筋肉の発達を想像させたりしている。図 1 9（A）～（C）では、キャラクタ 3 0 0 の上腕 3 1 0 と下腕とをつなぐ関節 3 0 8（つまり肘）を特に示す。上記コーディングを用いて、図 1 9（B）および（C）に示すように、関節 3 0 8 を曲げることで上腕 3 1 0 と 3 1 2 とが互いに隣接（接触）して位置した場合、ルーチン 3 5 0 では（下腕 3 1 0 のアルファ ID と、上腕 3 1 2 のアルファ ID との差が 1 より大きいかどうかに基づいて）、部位 3 1 0 と 3 1 2 との間に境界線セグメント 3 1 6 を付与する。

【 0 0 6 8 】

これにより、本発明は、三次元コンピュータグラフィックスにおいて、漫画的な輪郭線のような非写実的な効果を付与するための効果的な手法を提供する。これら手法は、画素後

10

20

30

40

50

処理の段階で（すなわち、三次元グラフィックスパイプラインが画像をフレームバッファにおいて描画されてから）用いられ、三次元グラフィックスパイプラインによって与えられる以外の画素情報を必要としない（すなわち、色、深度およびアルファ画素情報）。このように、本発明における画素後処理の手法は、フレームバッファ（たとえば、回路上のバッファの一部から回路外のバッファへ、エリアシングまたは他の後処理動作）の「ダウンストリーム」で効果的に、かつ効率良く行われ、これによって三次元グラフィックスパイプラインまたはシステムの他の構成要素が実質的に複雑化することもない。

【0069】

本発明は、現在最も現実的で好ましい実施例と思われるものに関連して説明されてきたが、本発明は、開示された実施例に限定されるものではなく、様々な変形例や均等な構成を含むことを想定していると解釈される。

10

【図面の簡単な説明】

【図1】本発明を具体化するための三次元ビデオグラフィックスシステムの例を示す図である。

【図2】本発明を実施しうるインタラクティブ三次元コンピュータグラフィックスシステム1005全体の詳細な模式図である。

【図3】コプロセッサ1020内の構成要素例のブロック図である。

【図4】三次元グラフィックスプロセッサ1107と、それに関連するコプロセッサ1020内の構成要素のより詳細な図である。

【図5】本発明におけるスクリーン効果を例示的に示す図である。

20

【図6】エミュレーション処理全体の一例を示した図である。

【図7】エミュレータ1303と共に用いるのに適した、パーソナルコンピュータに基づくホストを示した図である。

【図8】好ましい実施例に係る画素フィルタリングルーチン100の一例を示すフローチャートである。

【図9】画素フィルタルーチン100が隣接画素2つの深度値を読み出す例を示した模式図である。

【図10】遠くに画素 $P_{ix A}$ 、近くに画素 $P_{ix B}$ がある場合の遠近処理を示した模式図である。

【図11】図8に示すフローチャートを変更して、 $\log_2(n)$ のレンジ変換式を加えたフローチャートである。

30

【図12】ブレンディングを制御するのに用いられる、アルファ値の深度変調を行う後処理画素フィルタ200の一例である。

【図13】漫画的な輪郭線処理による漫画キャラクターの一例を示す図である。

【図14】漫画的な輪郭線処理による漫画キャラクターの別例を示す図である。

【図15】漫画的な輪郭線処理の境界線の箇所を特定するための対象物識別値を用いた、本発明のさらなる実施形態を示す図である。

【図16】本実施例において、境界線を付与するための画素後処理ルーチンの一例を示すフローチャートである。

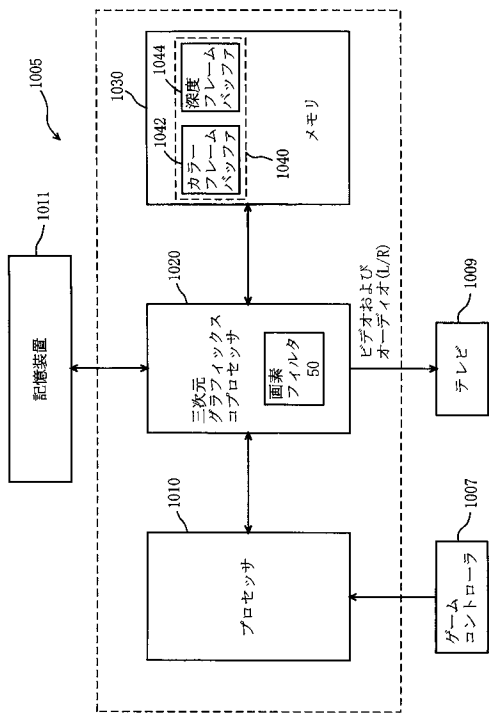
【図17】ルーチン350の変形例を示した模式図である。

40

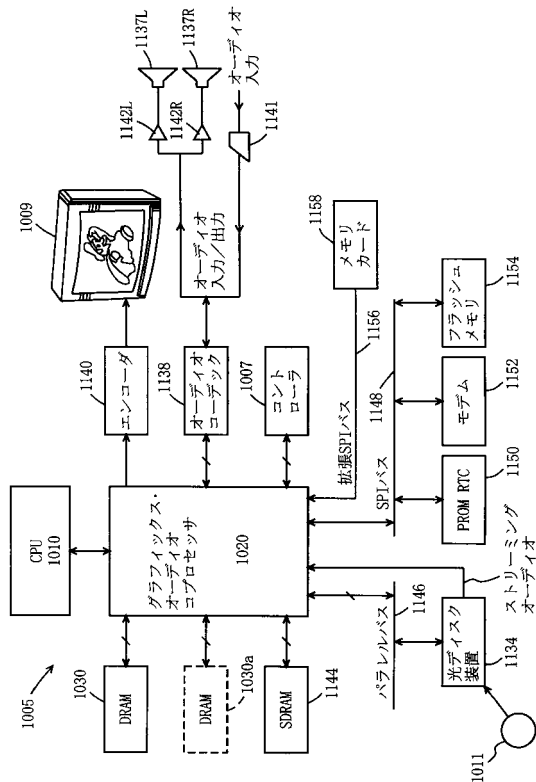
【図18】漫画的な輪郭線効果を得るために行う、対象物識別値コーディングの一例を示す図である。

【図19】漫画的な輪郭線効果を得るために行う、対象物識別値コーディングの別例を示す図である。

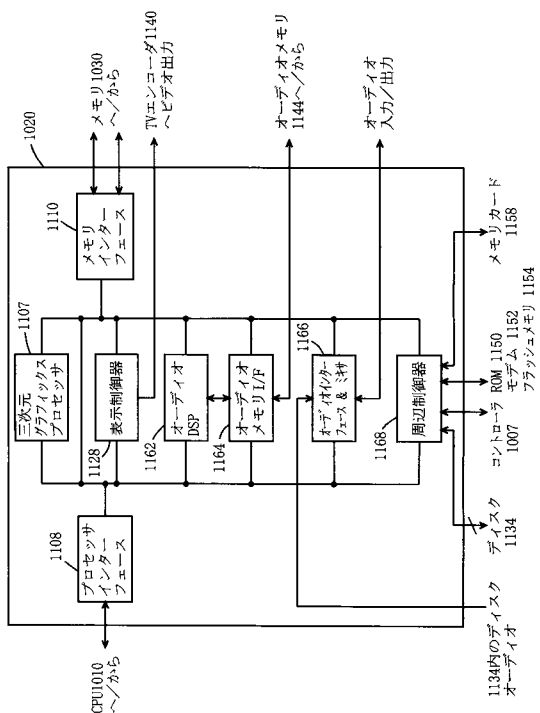
【図1】



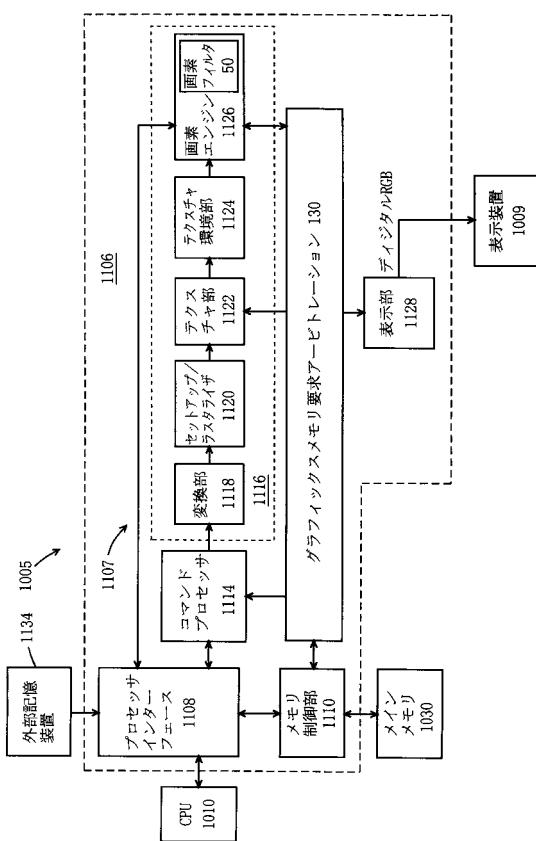
【図2】



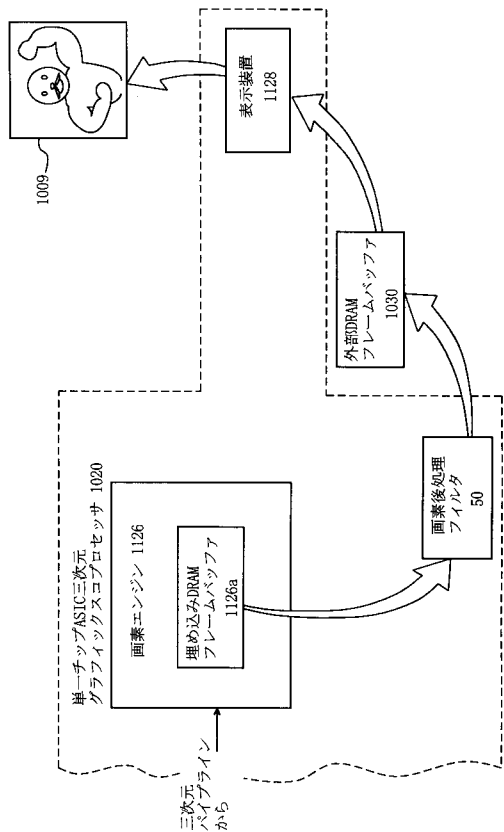
【図3】



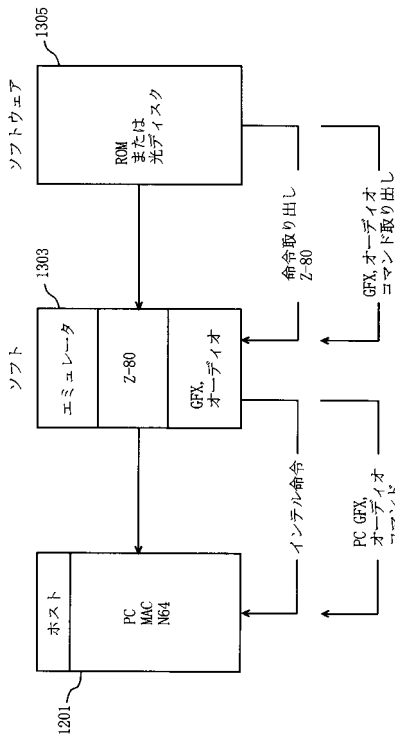
【図4】



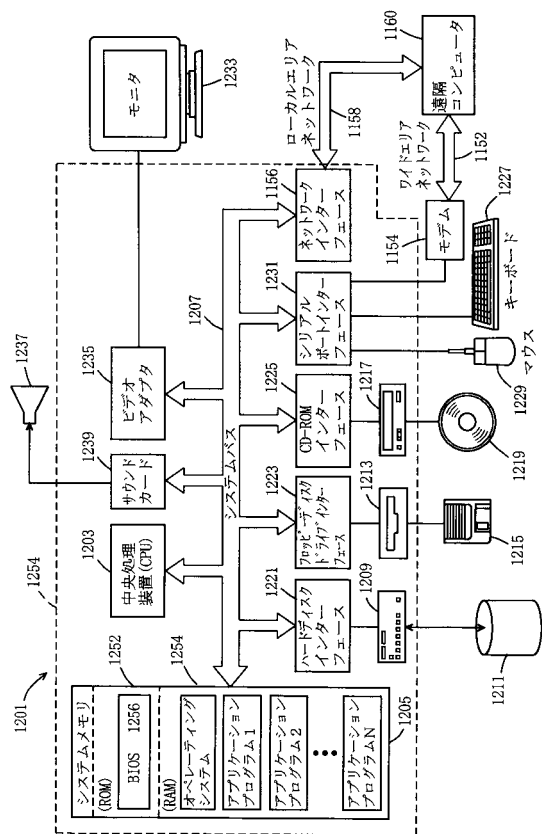
【図5】



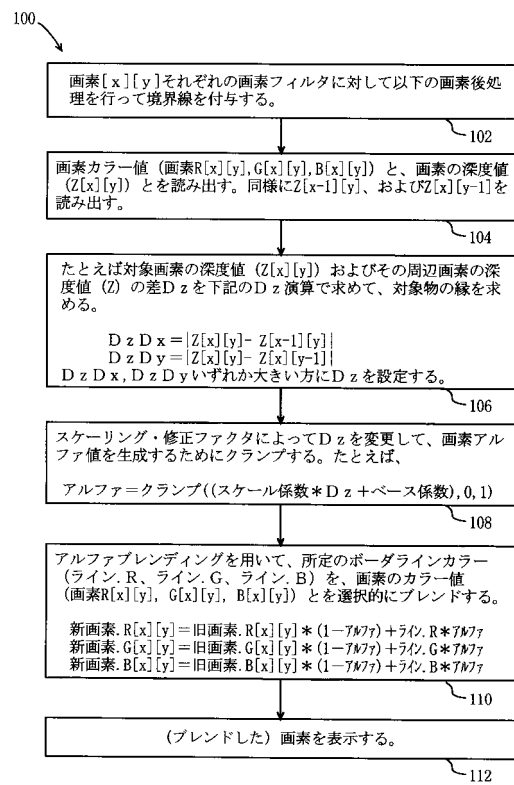
【図6】



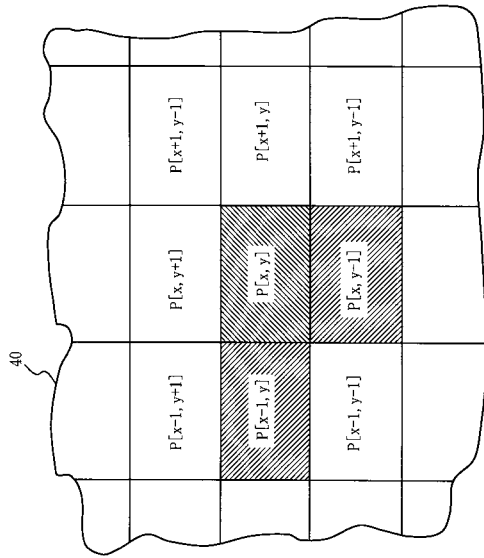
【図7】



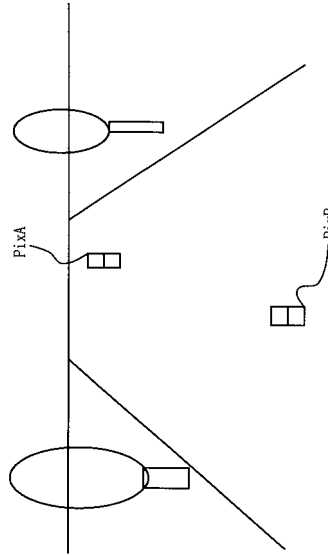
【図8】



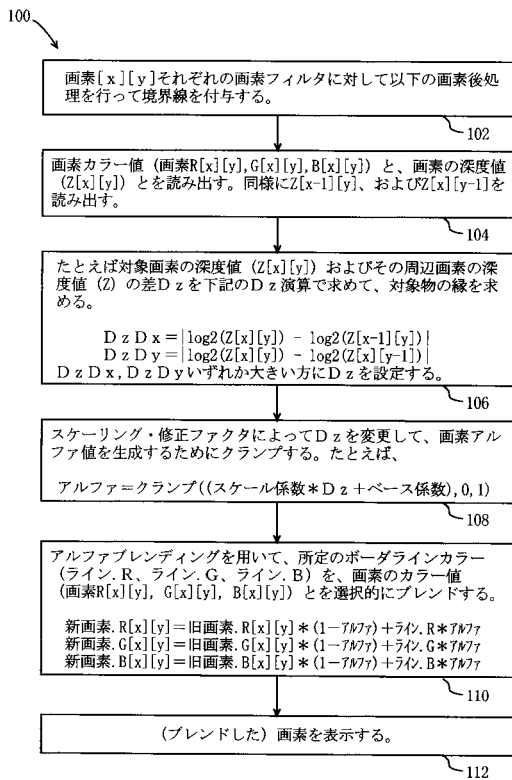
【図9】



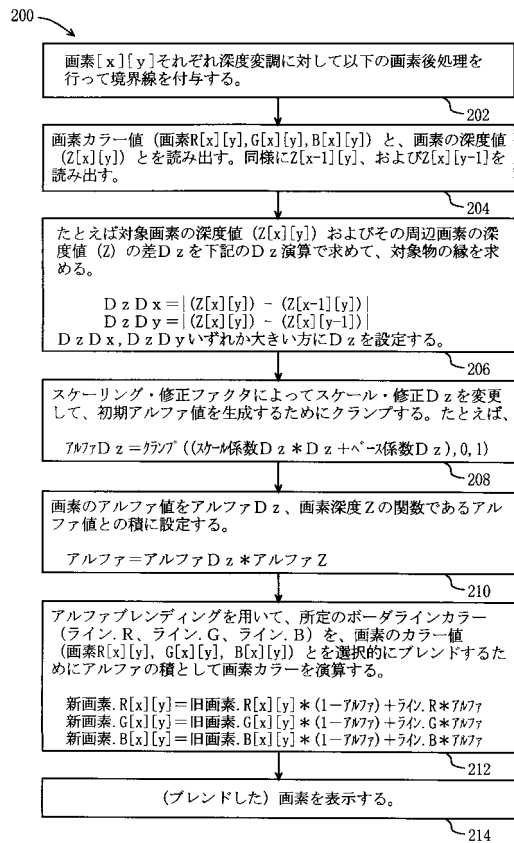
【図10】



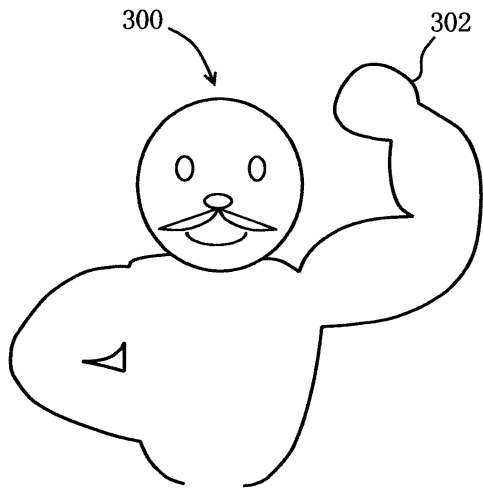
【図11】



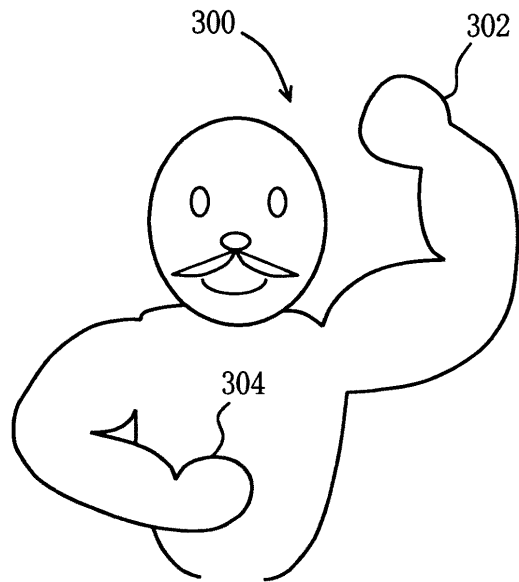
【図12】



【図13】

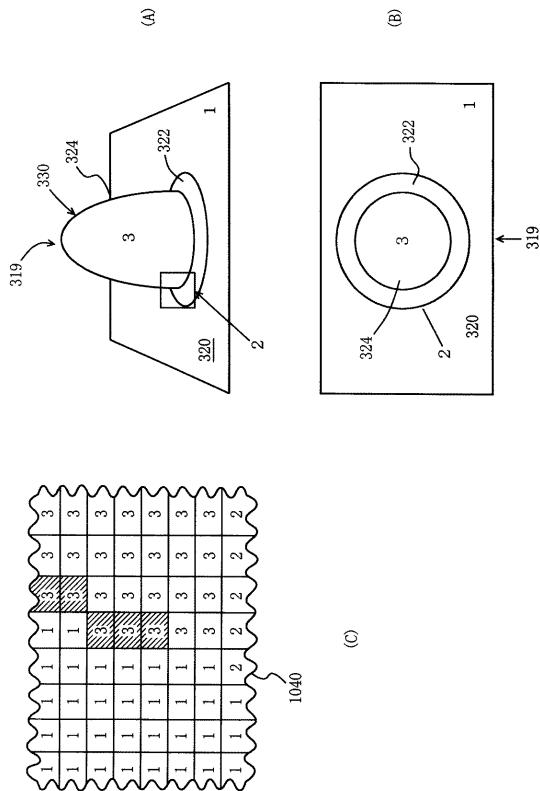


【図14】

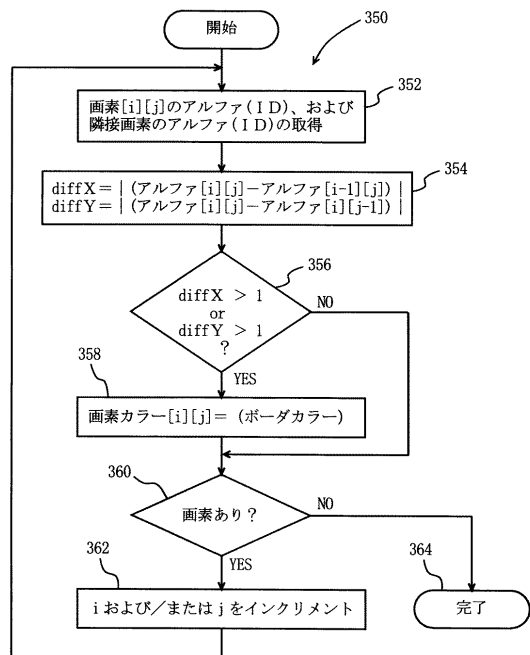


【図14】

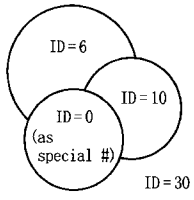
【図15】



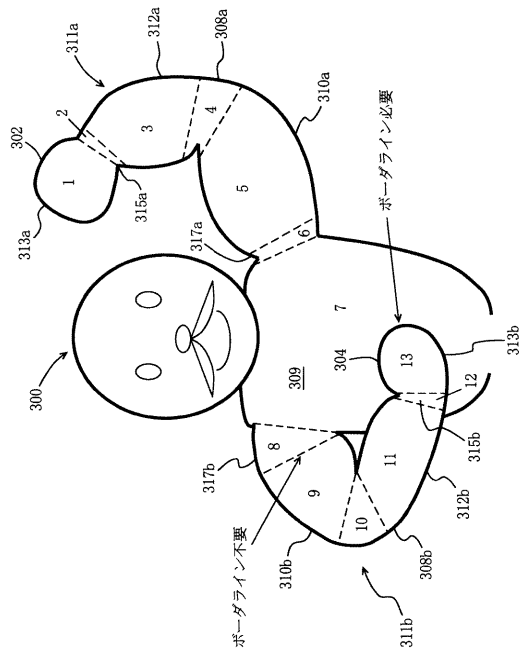
【図16】



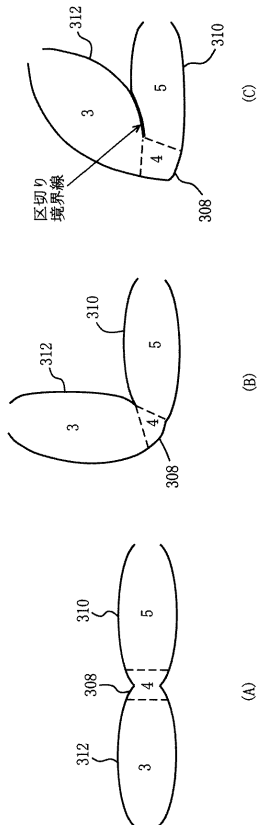
【 図 17 】



【 図 18 】



【 図 19 】



フロントページの続き

(58)調査した分野(Int.Cl. , D B名)

G06T 15/00 - 17/50

G09G 5/00 - 5/36

A63F 13/00

CSDB(日本国特許庁)