

(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2008-9897

(P2008-9897A)

(43) 公開日 平成20年1月17日(2008.1.17)

(51) Int. Cl.

G06T 11/40 (2006.01)

F I

G06T 11/40 200D

テーマコード (参考)

5B080

審査請求 未請求 請求項の数 5 O L (全 18 頁)

(21) 出願番号 特願2006-181921 (P2006-181921)

(22) 出願日 平成18年6月30日 (2006.6.30)

(71) 出願人 000006013

三菱電機株式会社

東京都千代田区丸の内二丁目7番3号

(74) 代理人 100123434

弁理士 田澤 英昭

(74) 代理人 100088605

弁理士 加藤 公延

(74) 代理人 100101133

弁理士 濱田 初音

(72) 発明者 濱田 雅樹

東京都千代田区丸の内二丁目7番3号 三

菱電機株式会社内

(72) 発明者 加藤 義幸

東京都千代田区丸の内二丁目7番3号 三

菱電機株式会社内

最終頁に続く

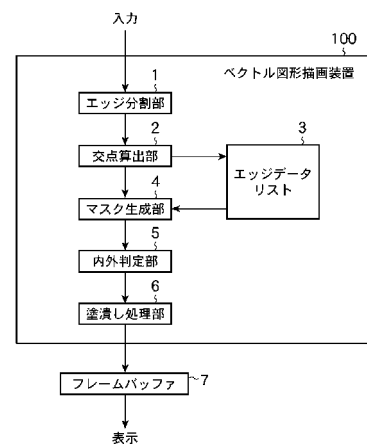
(54) 【発明の名称】 ベクトル図形描画装置及びそのプログラム

(57) 【要約】

【課題】多大な計算量を要することなく高速な描画を実行することのできるベクトル図形描画装置を得る。

【解決手段】交点算出部2は、エッジ分割部1が生成したエッジとスキャンラインとの交点座標をサブピクセル精度で求めた値をエッジデータとして算出し、エッジデータリスト3に登録する。マスク生成部4は、エッジデータリスト3からエッジデータをサブスキャンライン単位で取得し、図形のマスク情報を生成する。内外判定部5は、マスク生成部4が生成したマスク情報を元に、図形の内外判定を行ってカバレッジ情報に変換する。塗潰し処理部6は、カバレッジ情報に基づき、図形の内部と判定された領域に対して塗潰し処理を実行する。

【選択図】図1



【特許請求の範囲】

【請求項 1】

ベクトル図形の少なくとも頂点座標を含むベクトル図形のデータを入力し、前記入力データが曲線である場合には、当該曲線を微小な線分であるエッジに分割するエッジ分割部と、

前記エッジ分割部が生成したエッジとスキャンラインとの交点座標をサブピクセル精度で求めた値をエッジデータとして算出する交点算出部と、

前記算出されたエッジデータをサブスキャンライン単位で取得し、図形のマスク情報を生成するマスク生成部と、

前記マスク生成部が生成したマスク情報を元に、図形の内外判定を行ってカバレッジ情報に変換する内外判定部と、 10

前記内外判定部のカバレッジ情報に基づき、前記図形の内部と判定された領域に対して塗潰し処理を実行する塗潰し処理部とを備えたベクトル図形描画装置。

【請求項 2】

交点算出部は、交点座標の算出時にピクセル内部におけるエッジとサンプル点との位置関係を判定し、当該判定結果と交点座標の整数部およびエッジの向きの情報をエッジデータとして算出することを特徴とする請求項 1 記載の図形描画装置。

【請求項 3】

マスク生成部は、サブピクセル毎に、エッジの向きに基づいたスキャンラインとの交差回数を保持する 1 ライン分のスキャンラインバッファを備えたことを特徴とする請求項 2 記載のベクトル図形描画装置。 20

【請求項 4】

内外判定部は、サブスキャンライン単位で 1 ピクセル分のカバレッジをカウントしたそれぞれのピクセルの値を保持するための 1 ライン分のカバレッジバッファを備え、当該カバレッジバッファを用いてカバレッジ情報を保持することを特徴とする請求項 3 記載のベクトル図形描画装置。

【請求項 5】

コンピュータを、請求項 1 から請求項 4 のうちのいずれか 1 項記載のベクトル描画装置として機能させるためのベクトル図形描画装置のプログラム。

【発明の詳細な説明】 30

【技術分野】

【0001】

この発明は、ベクトル図形を描画するベクトル図形描画装置及びこの機能をコンピュータに実現させるベクトル図形描画装置のプログラムに関するものである。

【背景技術】

【0002】

従来、ベクトル図形を描画する図形描画装置においては、輪郭線のデータが曲線（ベジェ曲線等）で与えられた場合には、曲線を微小な線分（エッジ）に分割した後、多角形の描画と同様の手法で描画されるのが一般的である。エッジの分割数を増加させるほど、ベクトル図形の輪郭を曲線に近いものに近似できるが、頂点数が増大し複雑な図形となってしまう場合が多い。 40

【0003】

また、ベクトル図形の輪郭線が途中で交差するような場合には、図形の内外を区別するため一般に、イーブンオッド規則、ノンゼロ規則の 2 つの塗潰し規則が定められている。これらの規則に対応する方法の例としては、オーダードエッジリスト方式があり、走査線に交差する輪郭線を抽出し、この交差する位置を走査方向にソートして、交差する順番に走査線を辿り、交差位置に辿り着いた場合に、輪郭線のベクトル方向によって加算器に + 1（または - 1）し、加算結果に従って走査線方向に向かって塗潰しを行うものが提案されている。この方式では、X 座標ソートに多大な負荷を必要とするため、例えば特許文献 1 で示されているように、傾き情報の比較を行うことでソート処理の負荷を削減する方式 50

が開示されている。

【 0 0 0 4 】

また、ベクトル図形の輪郭部分にジャギーが発生せず滑らかに表示されるようなベクトル図形を得るためには、アンチエイリアス処理が必要である。このようなアンチエイリアス処理に関する技術として、例えば特許文献 2 では、ベクトル図形の輪郭線の、サブピクセル分割法によるアンチエイリアス処理の方法として、エッジ部分にある判定されたピクセルのみ濃度を算出し、エッジに挟まれた部分は最高濃度値で設定することにより、アンチエイリアス処理を効率化する手法について開示されている。

【 0 0 0 5 】

【特許文献 1】特開平 1 0 - 1 4 3 6 7 6 号公報

10

【特許文献 2】特開平 6 - 6 8 2 7 1 号公報

【発明の開示】

【発明が解決しようとする課題】

【 0 0 0 6 】

オーダードエッジリスト方式による多角形描画方式では、輪郭線との交点を走査線毎に毎回抽出し、その交点を走査方向にソートしなければならず、この処理に大きく処理時間を割いてしまうという課題があった。このような課題に対して、例えば、特許文献 1 に開示される技術においても、ソート回数は削減されても、複雑な図形を描画する場合には、依然として処理負荷を要するという問題点があった。

【 0 0 0 7 】

20

また、特許文献 2 に開示される技術においても、交点 X 座標値の小さい順に配列する必要がある、1 ライン上の交点数が増大した場合には描画処理の度に多大なソート処理を必要とするという課題がある。更に、ノンゼロ規則の塗潰しへの対応については開示されていない。

このように、従来のベクトル図形描画技術では、多大な計算量を要することなく高速な描画を実行することが困難であった。

【 0 0 0 8 】

この発明は上記のような課題を解決するためになされたもので、多大な計算量を要することなく高速な描画を実行することのできるベクトル図形描画装置及びそのプログラムを得ることを目的とする。

30

【課題を解決するための手段】

【 0 0 0 9 】

この発明に係るベクトル図形描画装置は、ベクトル図形のデータにおける曲線を微小な線分であるエッジに分割するエッジ分割部と、エッジ分割部が生成したエッジとスキャンラインとの交点座標をサブピクセル精度で求めた値をエッジデータとして算出する交点算出部と、算出されたエッジデータをサブスキャンライン単位で取得し、図形のマスク情報を生成するマスク生成部と、マスク生成部が生成したマスク情報を元に、図形の内外判定を行ってカバレッジ情報に変換する内外判定部と、内外判定部のカバレッジ情報に基づき、図形の内部と判定された領域に対して塗潰し処理を実行する塗潰し処理部とを備えたものである。

40

【発明の効果】

【 0 0 1 0 】

この発明のベクトル図形描画装置は、エッジデータをサブスキャンライン単位で取得して図形のマスク情報を生成し、このマスク情報に基づいて図形の内外判定を行いカバレッジ情報に変換するようにしたので、多大な計算量を要することなく高速な描画を実行することができる。

【発明を実施するための最良の形態】

【 0 0 1 1 】

実施の形態 1 .

図 1 は、この発明の実施の形態 1 によるベクトル図形描画装置を示す構成図である。

50

図において、ベクトル図形描画装置 100 は、エッジ分割部 1、交点算出部 2、エッジデータリスト 3、マスク生成部 4、内外判定部 5、塗潰し処理部 6 を備えている。

【0012】

エッジ分割部 1 は、ベクトル図形の少なくとも頂点座標と描画色を含むベクトル図形のデータを入力し、その入力データが曲線であるか否かを判定して、曲線である場合には、微小な線分であるエッジに分割する手段である。即ち、エッジ分割部 1 は、電子データとして入力されたベクトル図形の輪郭データが曲線で入力された場合には、曲線を微小な線分に近似する。以後、この分割された線分のことをエッジと呼ぶ。

【0013】

交点算出部 2 は、エッジ分割部 1 が生成したエッジとスキャンラインとの交点 X 座標をサブピクセル精度で求めてエッジデータをエッジデータリスト 3 に登録する手段である。即ち、交点算出部 2 は、算出した交点座標をエッジの向きの情報と合わせて、エッジデータリスト 3 に保存する。この時、エッジと走査線との交差回数をライトポイントに保存する。また、交点算出部 2 は、ベクトル図形の輪郭線の始点から開始して終点に達するまで同様の処理を繰り返し行う。

エッジデータリスト 3 は、交点算出部 2 が算出したベクトル図形の始点から終点までエッジデータのリストである。

【0014】

マスク生成部 4 は、交点算出部 2 が算出したエッジデータをエッジデータリスト 3 から走査線（サブスキャンライン）単位で読み出し、図形のマスク情報を生成する手段であり、このマスク情報を生成するためのスキャンラインバッファ（以降、SLB と表記する）を有している。この SLB とは、各サブピクセル毎に、エッジとの交差回数（エッジの向きに基づいてインクリメント / デクリメントした合計値）の情報を有するものである。

【0015】

内外判定部 5 は、マスク生成部 4 が生成したマスク情報を元に、図形の内外判定を行ってカバレッジ情報に変換する手段である。即ち、内外判定部 5 は、サブスキャンライン単位で 1 ピクセル分のカバレッジをカウントしたそれぞれのピクセルの値を保持するための 1 ライン分のカバレッジバッファ（以降、CVB と表記する）を備え、図形の内部と判定した場合は、この CVB を用いてカバレッジ（占有率）をカウントするよう構成されている。

塗潰し処理部 6 は、内外判定部 5 のカバレッジ情報に基づき、図形の内部と判定された領域に対して塗潰し処理を実行する手段である。

フレームバッファ 7 は、ベクトル図形描画装置 100 が生成したベクトル図形のピクセルデータを保持するものである。

【0016】

また、ベクトル図形描画装置 100 の構成要素であるエッジ分割部 1、交点算出部 2、マスク生成部 4、内外判定部 5 及び塗潰し処理部 6 は、本発明の趣旨に従う図形描画プログラムをコンピュータに読み込ませてその動作を制御することにより、そのコンピュータ上でソフトウェアとハードウェアが協働した具体的な手段として実現することもできる。即ち、上記エッジ分割部 1 ~ 塗潰し処理部 6 の各構成は、それぞれの機能に対応したプログラムと、これらプログラムを実行するための CPU やメモリといったハードウェアから実現することができる。

また、このコンピュータには、通信機能を有し、上記の図形描画プログラムを実行することができる携帯電話や携帯情報端末も含まれる。尚、コンピュータ自体の構成及びその基本的な機能については、当業者が当該技術分野の技術常識に基づいて容易に認識できるものであり、本発明の本質に直接関わるものでないので詳細な記載は省略する。

【0017】

次に、実施の形態 1 の動作について説明する。

ベクトル図形描画装置 100 は、この図形描画装置 100 が搭載されたシステムの CPU から入力されたベクトル図形の情報として、頂点座標、曲線の制御点の座標、描画色等

の情報が設定され、起動命令を受けることにより動作を開始する。起動命令を受けると、エッジ分割部 1 が起動し、輪郭線が曲線である場合には曲線を直線に近似してエッジ単位に分割する。例えば、図 2 に示すような図形内部の塗潰しを行う場合、エッジ分割部 1 は輪郭線を微小なエッジに分割する。描画開始位置の頂点座標を $v(0)$ 、 i 番目の頂点座標を $v(i)$ 、頂点数を n と表すと、エッジ分割部は、

$v(0) v(1)$ 、 $v(1) v(2)$ 、 $v(2) v(3)$ 、 \dots 、 $v(i) v(i+1)$ 、 \dots 、 $v(n-1) v(0)$

という順序でエッジを生成する。

【0018】

尚、本発明のベクトル図形描画装置で描画するベクトル図形は、輪郭線が途中で交差し

10

【0019】

また、図 3 に示すように図 2 の図形の輪郭線を CPU から指定された線幅だけ太らせたような図形の描画方法の一例としては、図 2 の図形を中心として、内側と外側にそれぞれ線幅の 2 分の 1 の距離だけ離れた点の座標を算出し、図 3 の拡大図に示すように四角形を連続して描画する方法がある。この時、それぞれの四角形がパスに対応し、複数のパスをまとめたものが描画する図形となる。

図 3 では、描画開始位置の頂点座標を $v(0)$ 、 i 番目の頂点座標を $v(i)$ 、頂点数を n と表すと、エッジ分割部は、

20

$v(0) v(1)$ 、 $v(1) v(3)$ 、 $v(3) v(2)$ 、 $v(2) v(0)$ 、
 $v(2) v(3)$ 、 $v(3) v(5)$ 、 $v(5) v(4)$ 、 $v(4) v(2) \dots$

という順序でエッジを生成する。尚、ここで示した順序は一つの例であり、更に輪郭線が滑らかに繋がるように、別の図形を挿入することも可能である。

以上で示したようにエッジ分割部 1 は、エッジの始点、終点の座標を交点算出部に対して随時出力する。この時、出力するエッジが図形の間中エッジ (0) であるか最終エッジ (1) であるかという判定フラグも一緒に出力する。

【0020】

交点算出部 2 は、エッジ分割部 1 から出力されたエッジと、スキャンラインとの交点座標を算出する。エッジの両端の座標を $v_0(x_0, y_0)$ 、 $v_1(x_1, y_1)$ 、スキャンラインの Y 座標を y とする場合、スキャンラインとエッジとの交点 X 座標 $x_s(y)$ は、以下の式 (1) で求めることができる。ただし、 $(y_0 = y_1)$ の時には、交点座標の算出は行わない。

30

$$x_s(y) = x_0 + (x_1 - x_0) / (y_1 - y_0) \times (y - y_0) \dots (1)$$

【0021】

ここで、本発明では、ベクトル図形の輪郭線部分のアンチエイリアス処理をサブピクセル分割法を利用して行う。サブピクセル分割法では、各ピクセルに対して、図形内部に含まれるサンプリング点の個数をカウントすることによりピクセル単位のカバレッジを算出する。

40

【0022】

図 4 に 4×4 および 2×2 のサブピクセル分割の例を示す。

図 4 では、点線で示す図形を描画する場合に、図形内部に含まれるサブピクセルを黒丸、図形外部のサブピクセルを白丸で示している。1 ピクセル当たりのサンプル数を多く取るほどカバレッジの精度が向上する。一般に $s_x \times s_y$ のサブピクセル分割において X 方向のサブピクセル分割数 (s_x) と、Y 方向のサブピクセル分割数 (s_y) の値が互いに異なってもよい。

従って、上式 (1) においては、 y の値を Y 方向のサンプル数 (y_{sub}) を考慮した値として、次の式 (2)、(3) で表せる値とする。但し、 i_{y0} はエッジの最小 Y 座標の整数部、 i_{y1} はエッジの最大 Y 座標の整数部、 n は整数とする。式 (2) で表される

50

dy が Y 方向の刻み幅を表す。また、 $ysub$ の値は 1 以上の任意の整数値でよいが、2 のべき乗で表される値とするのが望ましい。

$$dy = 1 / (2 \times ysub) \quad \dots (2)$$

$$y = iy0 + (2n + 1) \times dy \quad \dots (3)$$

$$iy0 \leq y \leq iy1 \text{ かつ } \min(y0, y1) \leq y < \max(y0, y1) \dots (4)$$

以降、式 (3) で表すことのできる各 Y 座標のことをサブスキャンラインと呼ぶ。

【0023】

交点算出部 2 は、 y の値が式 (4) の条件を満たす範囲内で式 (1) ~ (3) に従って、サブスキャンラインの Y 座標を $2dy$ ずつ変化させながら、エッジとの交点を順次求める。全ての交点座標の算出が終了した後、エッジ分割部 1 から出力された判定フラグの値が最終エッジ (1) である場合には、マスク生成部 4 を起動する。中間エッジ (0) である場合には、エッジ分割部 1 から次のエッジの座標と判定フラグを受け取り、同様の処理を繰り返す。

【0024】

エッジデータリスト 3 は、交点算出部 2 が算出したサブスキャンラインとエッジとの交点情報を保存するためのリストである。本発明では、アンチエイリアス処理に対応するため、エッジデータリスト 3 のエッジデータを、エッジの向き、交点 X 座標の整数部、サンプル数の 3 つの情報で構成する。

【0025】

エッジの向きは、イーブンオッド規則、ノンゼロ規則に対応するために必要となる。エッジの向きの定義は、エッジの両端の Y 座標を比較して ($y0 < y1$) ならば上向き (UP)、($y0 > y1$) ならば下向き (DOWN) とする。交点 X 座標の整数部の値をエッジデータリスト 3 に保存し、小数点以下の情報は、サブピクセル分割の精度に応じて最低限必要な精度をサンプル数としてエッジデータリスト 3 に保存する。

【0026】

サンプル数とは、処理中のサブスキャンライン上において各ピクセル内部でエッジの右側にあるサンプル点の個数を表す。

図 5 に 4×4 のサブピクセル分割の例を示す。図 5 (a)、(b) において、斜線付きの丸印で示したのがピクセル内部でエッジの右側に位置するサンプル点である。このようなサンプル点の個数を各サブスキャンライン ($ysub0 \sim ysub3$) でカウントする。サンプル点の中心がエッジ上にある場合には、エッジの右側にあるとして判定する。また、エッジが図 5 (b) に示すようにピクセルの境界をまたぐような場合には、サブスキャンライン単位で交点 X 座標の整数部の値が異なることになる。

【0027】

以上で示したエッジの向き、交点 X 座標の整数部、サンプル数の 3 つのデータを組として、エッジデータリスト 3 に登録する。

図 6 は、エッジデータリスト 3 のデータ構成を示す (Y 方向のサブピクセル分割数 = 2 の場合)。

各サブスキャンラインは Y 座標 (y) と、サブピクセルカウンタ ($ysub$) の値によって決まる。各行のデータが当該サブスキャンラインとエッジとの交点座標の数に対応している。各データは、(エッジの向き (0: DOWN、1: UP)、交点 X 座標の整数部、サンプル数) を 2 つずつ組にしたものである。2 つのデータでエッジの向きは、どちらか一方が DOWN、もう一方が UP となる。また 1 サブスキャンライン上に含まれるエッジデータは必ず偶数個となる。図 6 の方式は 1 つの例であり、他のデータ構成とすることも当然可能である。

【0028】

マスク生成部 4 は、交点算出部 2 からの起動命令を受けると、エッジデータリスト 3 に保存された交点情報を随時読み出し、アンチエイリアス処理のためのマスク情報を生成する。従来技術では、各サブスキャンライン上の交点座標を小さいものから順にソートする必要があったが、本発明の方式では X 座標のソートを必要としない。本発明では、1 サブ

10

20

30

40

50

スキャンライン分の図形のマスク情報を格納できる S L B を利用し、1 サブスキャンラインの処理が終わる度にこのバッファをクリアして繰り返し利用する。

【 0 0 2 9 】

更に、ノンゼロ塗潰し規則にも対応するため、S L B の各アドレスに複数ビットを割り当てる。ビット数はエッジとスキャンラインとの交差回数に依存し、例えば、ビット数を 8 とした場合にはエッジとスキャンラインとの交差回数（エッジの向きに基づきインクリメントおよびデクリメントした合計値）が 2 5 5 の図形まで対応することができる。

従って、S L B のサイズは、最大で（フレームバッファ 7 の X 方向サイズ × X 方向サブピクセル数 × ビット数）だけ必要である。

【 0 0 3 0 】

次に、S L B のリニアアドレスと、X アドレスとの対応を図 7 に示す。X アドレスは、X 方向サンプル数（x s u b）に応じて動的に変化させる。（x s u b）が 1 の場合には、アンチエイリアス処理は無効となる。

S L B の 0 番地が X 座標の 0 に対応するとすれば、図 7 において S L B の i 番地が以下の X 座標に対応する。記号（>>）は右方向へのビットシフト演算子を表す。

例 1 . X 方向サンプル数 = 1 の場合、S L B の i 番地が X 座標 i に対応する。

例 2 . X 方向サンプル数 = 2 の場合、S L B の i 番地が X 座標（i >> 1）に対応する。

例 3 . X 方向サンプル数 = 4 の場合、S L B の i 番地が X 座標（i >> 2）に対応する。

【 0 0 3 1 】

S L B の X 座標の基準値は 0 としてもよいが、例えば、図 8 に示すベクトル図形の包含矩形の最小 X 座標（x m i n）を基準値として S L B の容量を節約することもできる。この時必要となる S L B のサイズは、（包含矩形の幅 × X 方向サブピクセル数 × ビット数）だけ必要である。

【 0 0 3 2 】

次に、S L B の状態の一例を図 9 を用いて説明する。図 9（a）は S L B における値の一例を示している。また、このような S L B の値となる図形としては例えば図 1 0 に示すような図形である。尚、図 9 の例では X 方向サブピクセル分割数（x s u b）は 2 としている。

【 0 0 3 3 】

図 9（a）において、例えば、X 座標（X アドレス）が 2 の位置において、上向きのエッジとスキャンラインとが交差し、S L B の値は「+ 1」となる。尚、図 1 0 において、+ 1, + 1, + 2, ... といった位置がサブピクセルの位置に相当している。次に、X 座標が 3 の位置において、同様に、上向き（U P）のエッジとスキャンラインとが交差するため、S L B の値はインクリメントされ「+ 2」となる。そして、次の X 座標が 4 の位置ではサブピクセル毎に上向きのエッジと交差するため、x s u b = 0 の S L B の値が「+ 3」、x s u b = 1 の値が「+ 4」となる。そして、次の X 座標が 5 の位置のサブピクセル x s u b = 0 では下向き（D O W N）のエッジと交差するためデクリメントされて「+ 3」となり、更に x s u b = 1 でも下向きのエッジと交差することから「+ 2」となる。

【 0 0 3 4 】

このように、X 座標の増加する向きにスキャンラインを辿っていき、エッジと交差する度に、上向きエッジでは S L B の値を + 1 とし、下向きエッジではその値を - 1 する。

【 0 0 3 5 】

内外判定部 5 は、S L B を随時読み出し、ピクセル単位のカバレッジ情報に変換する。内外判定方法は以下の通りとする。

（1）塗潰し規則がイーブンオッド規則の場合は、S L B の値が奇数の時に図形の内部であると判定する。

（2）塗潰し規則がノンゼロ規則の場合は、S L B の値が 0 以外であった時に図形の内部であると判定する。

10

20

30

40

50

このような規則に従って内外判定を行うと、例えば、図 9 および図 10 の図形では、図形の内部と判定される領域は、イーブンオッド規則では、+ 1 と + 3 で示した領域であり、ノンゼロ規則では、+ 1, + 2, + 3, + 4 で示した領域となる。

【0036】

本発明では、1 スキャンライン分のピクセル単位のカバレッジ情報を格納できる C V B を利用する。C V B の各ピクセルのビット数は、サブピクセル分割数をカウントできるだけのビット数を確保する。例えば、2 × 2 のサブピクセル分割の時は 0 ~ 4 を表現するための 3 ビット、4 × 4 のサブピクセル分割の時は 0 ~ 16 を表現するための 5 ビットが必要である。

【0037】

内外判定部 5 は、1 サブスキャンラインの処理が終わる度に図形の内部と判定された領域に対応する C V B の値をインクリメントする。また、1 スキャンライン上に含まれる Y 方向サブピクセル分割数分のサブスキャンラインの処理が終わる度に C V B をクリアして繰り返し利用する。尚、C V B の X アドレスの基準値は 0 としてもよいが、ベクトル図形の包含矩形の最小 X 座標を基準値とすれば、C V B の容量を節約できる。

【0038】

次に、図 9 を参照し、S L B の情報から、内外判定を行う具体例について説明する。図 9 の例では、上述したように、X 方向サブピクセル分割数 (x s u b) は 2 としており、このような x s u b の値を考慮して、S L B のサブピクセル x s u b 個分の情報を 1 ピクセルのカバレッジに変換する。例えば、S L B の値が (+ 2) および (+ 4) のサブピクセルは、イーブンオッド規則では図形の外部となるため、図 9 (b) に示すように、C V B の値は X アドレスが 3 では「0」、X アドレスが 4 では一つのサブピクセルが奇数であるため、「+ 1」となる。また、ノンゼロ規則では S L B の値がゼロ以外のサブピクセルは図形の内部となるため、図 9 (c) に示すように、C V B の値は X アドレス 2 から X アドレス 5 までが「+ 2」、X アドレス 6 では、一方のサブピクセルのみがゼロ以外であるため、その値は「+ 1」となる。

【0039】

塗潰し処理部 6 は、内外判定部 5 によって図形の内部と判定された領域内を塗潰すように順次ピクセルを生成する。ピクセルの描画色 c (R、G、B の各成分に対応) は C P U から設定された描画色を s r c、既にフレームバッファに書かれている色を d s t とおくと、内外判定部 5 によって算出されたカバレッジの値 (c v) を用いて以下の式 (5) によって計算されるのが一例である。

$$c = s r c \times c v + d s t \times (1 - c v) \quad \dots (5)$$

これにより、ベクトル図形の輪郭線にアンチエイリアス処理を施して描画することができる。

【0040】

また、透過色を扱う場合には、一般的なアルファブレンド式を用いて、以下のように計算できる。但し、描画色のアルファ成分を s r c a、アルファブレンド結果を b とおく。

$$b = s r c \times s r c a + d s t \times (1 - s r c a) \quad \dots (6)$$

$$c = b \times c v + d s t \times (1 - c v) \quad \dots (7)$$

また、ピクセル単位でカラー勾配を計算してグラデーション塗潰しを実現したり、テクスチャアドレスを計算してテクスチャ貼り付けを行ったりすることも可能である。

【0041】

次に、実施の形態 1 によるベクトル図形描画処理の動作の詳細について説明する。

図 11 及び図 12 は、実施の形態 1 のベクトル図形描画装置によるベクトル図形描画処理の流れを示すフローチャートであり、以下、これらの図に沿って説明する。

まず、エッジ分割部 1 は、図形の輪郭線が曲線であるか否かを判定する (ステップ S T 1)。図形の輪郭線が曲線である場合には、微小線分 (エッジ) に分割を行う (ステップ S T 2)。輪郭線が直線である場合には、直線をそのままエッジとして扱う。

【0042】

次に、交点算出部 2 は、エッジとサブスキャンラインが交差するか否かを判定する（ステップ S T 3）。判定結果が Y E S である場合には、サブスキャンラインとエッジとの交点座標を式（1）を用いて算出し、交点座標を算出後、前述したエッジの向き、交点 X 座標の整数部、サンプル数をエッジデータリスト 3 に保存する（ステップ S T 4）。判定結果が N O である場合には、交点座標の算出とエッジデータの登録は行わない。

次に、処理しているエッジがベクトル図形の最終エッジであるかを判定し（ステップ S T 5）、そうでない場合は上記のステップ S T 3 に戻って同様の処理を繰り返す。一方、ステップ S T 5 において、最終エッジである場合には、塗潰しを開始する。塗潰し開始後のフローチャートは図 1 2 に示すものである。

【 0 0 4 3 】

図 1 2 において、塗潰しを開始されると、マスク生成部 4 は、Y 座標カウンタ（ y ）の初期値を図形の包含矩形の最小 Y 座標（ y_{min} ）に、Y 方向サブピクセルカウンタ（ y_{sub} ）を 0 に設定する（ステップ S T 6）。次に、マスク生成部 4 は、エッジデータリスト 3 の Y 座標カウンタの値に対応するアドレスからエッジデータを 1 組読み出す（ステップ S T 7）。上述したように 1 組のエッジデータは、一方が U P エッジの情報、もう一方が D O W N エッジの情報となっている。

【 0 0 4 4 】

続いて、以下の条件に従って S L B 判定フラグ（ x_{flag} ）を生成する（ステップ S T 8）。但し、読み出した i 番目のエッジデータの整数 X 座標をそれぞれ $x(i)$ 、 $x(i+1)$ として、これらの大小比較結果から以下の（A）、（B）の条件に従って S L B を更新する範囲（ x_{left} 、 x_{right} ）と S L B 判定フラグを定める。

【 0 0 4 5 】

（A） $x(i) < x(i+1)$ の時

$x_{left} = x(i)$ 、 $x_{right} = x(i+1)$

i 番目のエッジデータの向きが U P である場合、 $x_{flag} = 1$

（B） $x(i) \geq x(i+1)$ の時

$x_{left} = x(i+1)$ 、 $x_{right} = x(i)$

$i+1$ 番目のエッジデータの向きが U P である場合、 $x_{flag} = 1$

【 0 0 4 6 】

マスク生成部 4 は、生成した S L B 判定フラグの値を判定し（ステップ S T 9）、S L B 判定フラグが 1 である場合には、 x_{left} から x_{right} の範囲の S L B をインクリメントする（ステップ S T 10）。一方、S L B 判定フラグが 0 である場合には、 x_{left} から x_{right} の範囲の S L B をデクリメントする（ステップ S T 11）。

但し、 $x = x_{left}$ の位置においては、サンプル点がエッジの右側にあるサブピクセルに対応する S L B だけインクリメントする。 $x = x_{right}$ の位置においては、サンプル点がエッジの左側にあるサブピクセルの位置に対応する S L B のみをインクリメントする。

次に、マスク生成部 4 は、最終エッジデータかを判定し（ステップ S T 12）、最終エッジデータでなかった場合はステップ S T 7 に戻って、これらの処理を繰り返し、最終エッジデータであった場合は、内外判定部 5 を起動し、ステップ S T 13 に移行する。

【 0 0 4 7 】

ステップ S T 13 において、内外判定部 5 は、処理中のサブスキャンライン上における S L B の情報を読み出し、前述した内外判定規則に従って、ピクセル単位のカバレッジに変換する。次に、処理中のサブスキャンラインが最終サブスキャンラインである、即ち、Y 方向サブピクセルカウンタが（ $y_{sub} = (Y \text{ 方向サンプル数} - 1)$ ）の条件を満たすか否かを判定する（ステップ S T 14）。最終サブスキャンラインでない場合は、Y 方向サブピクセルカウンタ（ y_{sub} ）をインクリメントした後（ステップ S T 15）、ステップ S T 7 に戻ってこれ以降の処理を繰り返す。一方、ステップ S T 14 において、最終サブスキャンラインであった場合には、スキャンライン y 上のピクセル生成を開始する（ステップ S T 16）。

10

20

30

40

50

【 0 0 4 8 】

図 1 3 ~ 図 1 6 に 4×4 のサブピクセル分割時の図形描画例を示す。

図 1 3 は、1 スキャンライン上の 3 ピクセル分をサブピクセル単位で表したものである。また、図 1 4 はスキャンライン上のエッジデータを示す図である。1 スキャンラインは 4 つのサブスキャンラインで構成される。 $y_{sub} = 0 \sim 2$ のサブスキャンラインは、エッジとの交差回数がそれぞれ 4 回であり、 $y_{sub} = 3$ のサブスキャンラインは、エッジとの交差回数が 2 回である。マスク生成部 4 及び内外判定部 5 は、図 1 4 に示すエッジデータを元に、図 1 3 に黒丸で示しているサブピクセルが図形の内部に含まれるようにするため、以下の手順に従って判定を行う。

【 0 0 4 9 】

10

図 1 5 は、 $y_{sub} = 0$ のサブスキャンラインの処理手順を例として示す説明図であり、図中の (1) ~ (3) の状態が以下の (1) ~ (3) の説明に対応している。

(1) マスク生成部 4 は、図 1 4 の $y_{sub} = 0$ に対応するアドレスからエッジデータ 0、1 を読み出した結果、前記判定条件 (B) により

$x_{left} = x_{right} = p + 1$ 、 $x_{flag} = 1$ となるので

$x = p + 1$ の範囲の S L B をインクリメントする。

【 0 0 5 0 】

(2) 図 1 4 の $y_{sub} = 0$ に対応するアドレスからエッジデータ 2、3 を読み出した結果、前記判定条件 (B) により

$x_{left} = p - 1$ 、 $x_{right} = p$ 、 $x_{flag} = 1$ となるので

$p - 1 \sim x \sim p$ の範囲の S L B をインクリメントする。

20

【 0 0 5 1 】

(3) $y_{sub} = 0$ のサブスキャンライン上では、エッジデータ 2、3 の組が最終なので S L B へのマスク生成を終了し、内外判定部 5 は、4 サブピクセル分の内外判定結果を 1 ピクセル分のカバレッジ情報として変換し、C V B をインクリメントする ($y_{sub} = 0$ であるから、C V B は 0 クリアされている)。その後、S L B をクリアし、 $y_{sub} = 1$ のサブスキャンラインを同様に処理する。

【 0 0 5 2 】

そして、 $y_{sub} = 1$ 、 $y_{sub} = 2$ のサブスキャンラインに対しても同様の処理を行った後、 $y_{sub} = 3$ の最終サブスキャンラインの処理を開始する。

30

【 0 0 5 3 】

図 1 6 は、 $y_{sub} = 3$ のサブスキャンラインの処理手順を例として示す説明図であり、図中の (4) ~ (6) の状態は以下の (4) ~ (6) の説明に対応している。

(4) $y_{sub} = 2$ のサブスキャンラインの処理が終了した時点で C V B は図 1 6 の (4) に示す状態になっている。

【 0 0 5 4 】

(5) 次いで、マスク生成部 4 は、図 1 4 の $y_{sub} = 3$ に対応するアドレスからエッジデータ 0 と 3 を読み出した結果、前記判定条件 (B) により

$x_{left} = p - 1$ 、 $x_{right} = p + 1$ 、 $x_{flag} = 1$ となるので

$p - 1 \sim x \sim p + 1$ の範囲の S L B をインクリメントする。

40

【 0 0 5 5 】

(6) $y_{sub} = 3$ のサブスキャンライン上では、エッジデータ 0、3 の組が最終なので S L B へのマスク生成を終了し、内外判定部 5 は、4 サブピクセル分の内外判定結果を 1 ピクセル分のカバレッジ情報として変換し、C V B をインクリメントする。最終サブスキャンラインであるから、C V B のアドレスを X 座標、C V B に保存されている値をカバレッジ、Y 座標カウンタの値を Y 座標として、1 スキャンライン分のピクセルデータ生成を開始する。カバレッジが 0 のピクセルは描画する必要がない。

【 0 0 5 6 】

図 1 2 に戻り、ピクセル生成が開始されると、塗潰し処理部 6 は、内外判定部 5 で図形の内部と判定された領域に含まれるピクセルを順次生成して塗潰し処理を行う (ステップ

50

S T 1 6)。その後、1 スキャンライン上で図形の内部に含まれる全てのピクセルの塗潰し処理が終了すると、Y 座標が最終値 (y m a x) であるか否かの判定を行う (ステップ S T 1 7)。Y 座標が最終値でない場合には、Y 座標カウンタ y のインクリメント、Y 方向サブピクセルカウンタ (y s u b) のリセットを行った後 (ステップ S T 1 8)、ステップ S T 7 に戻り、これ以降の処理を繰り返す。ステップ S T 1 7 において、Y 座標が最終値 (y m a x) である場合には、ベクトル図形の描画を終了する。

【 0 0 5 7 】

以上のように、この実施の形態 1 のベクトル図形描画装置によれば、ベクトル図形の少なくとも頂点座標を含むベクトル図形のデータを入力し、入力データが曲線である場合には、この曲線を微小な線分であるエッジに分割するエッジ分割部と、エッジ分割部が生成したエッジとスキャンラインとの交点座標をサブピクセル精度で求めた値をエッジデータとして算出する交点算出部と、算出されたエッジデータをサブスキャンライン単位で取得し、図形のマスク情報を生成するマスク生成部と、マスク生成部が生成したマスク情報を元に、図形の内外判定を行ってカバレッジ情報に変換する内外判定部と、内外判定部のカバレッジ情報に基づき、前記図形の内部と判定された領域に対して塗潰し処理を実行する塗潰し処理部とを備えたので、ベクトル図形描画装置として多大な計算量を要することなく高速な描画を実行することができる。

【 0 0 5 8 】

また、実施の形態 1 のベクトル図形描画装置によれば、交点算出部は、交点座標の算出時にピクセル内部におけるエッジとサンプル点との位置関係を判定し、この判定結果と交点座標の整数部およびエッジの向きの情報をエッジデータとして算出するようにしたので、アンチエイリアス処理を実現する際に、多大な計算量を要することなく高速な描画を実行することができる。

【 0 0 5 9 】

また、実施の形態 1 のベクトル図形描画装置によれば、マスク生成部は、サブピクセル毎に、エッジの向きに基づいたスキャンラインとの交差回数を保持する 1 ライン分のスキャンラインバッファを備えたので、イーブンオッド規則やノンゼロ規則といった塗潰し規則に対応することができる。

【 0 0 6 0 】

また、実施の形態 1 のベクトル図形描画装置によれば、内外判定部は、サブスキャンライン単位で 1 ピクセル分のカバレッジをカウントしたそれぞれのピクセルの値を保持するための 1 ライン分のカバレッジバッファを備え、このカバレッジバッファを用いてカバレッジ情報を保持するようにしたので、アンチエイリアス処理を施した任意のベクトル図形を塗潰し規則 (イーブンオッド規則、ノンゼロ規則) を考慮した上で、それぞれ 1 ライン分のスキャンラインバッファ及びカバレッジバッファを繰り返し利用することにより、単純な処理で高速に描画することができる。

【 0 0 6 1 】

また、実施の形態 1 のベクトル図形描画装置のプログラムによれば、コンピュータを上記いずれかのベクトル描画装置として機能させるようにしたので、多大な計算量を要することなく高速な描画を実行することができるベクトル図形描画装置をコンピュータ上に実現することができる。

また、実施の形態 1 のベクトル図形描画装置によれば、ベクトル図形描画装置のハードウェア化を行う場合でも回路規模を最小限に抑えることができる。

更に、スキャンラインとエッジとの交差回数が多い図形を扱ったり、アンチエイリアス処理のサンプル数を変化させたりする場合でも、エッジデータリストの構成およびスキャンラインバッファのアドレス割り当てを変えることによって柔軟に対応することが可能である。

【 0 0 6 2 】

実施の形態 2 .

上記実施の形態 1 では、図形に含まれるエッジ分割部 1 が生成したエッジとスキャンラ

10

20

30

40

50

インとの交点座標を全てエッジデータリスト 3 に保存してからベクトル図形の塗潰しを開始していたが、エッジとスキャンラインとの交点座標の算出を 1 スキャンライン分行う度に 1 スキャンライン分の塗潰しを行い、同様の処理をベクトル図形の包含矩形の Y 方向サイズ分だけ繰り返すようにしてもよく、これを実施の形態 2 として以下説明する。

【 0 0 6 3 】

図面上の構成は実施の形態 1 と同様であるため、図 1 を援用して説明する。

実施の形態 2 の交点算出部 2 は、エッジ分割部 1 が生成したエッジとスキャンラインとの交点 X 座標をサブピクセル精度で求めた値をエッジデータとして算出してエッジデータリスト 3 に登録し、また、マスク生成部 4 は、エッジデータリスト 3 に登録されているエッジデータをサブスキャンライン単位で取得し、図形のマスク情報を生成するよう構成されている。また、内外判定部 5 及び塗潰し処理部 6 は、1 スキャンライン分の値が得られる度に処理を行うようになっている。

10

【 0 0 6 4 】

次に、実施の形態 2 の動作について説明する。

図 17 は、実施の形態 2 によるベクトル図形描画処理の塗潰し開始までの流れを示すフローチャートである。

最初に Y 座標カウンタを最小 Y 座標、Y 方向サブピクセルカウンタ y_{sub} を 0 に設定する（ステップ S T 1 0 1）。エッジ分割部 1 は、図形の輪郭線が曲線であるか否かを判定する（ステップ S T 1 0 2）。図形の輪郭線が曲線である場合には、微小線分（エッジ）に分割を行う（ステップ S T 1 0 3）。輪郭線が直線である場合には、直線をそのままエッジとして扱う。

20

【 0 0 6 5 】

次に、交点算出部 2 は、入力されたエッジが Y 座標カウンタの値に対応するサブスキャンラインと交差するか否かを判定する（ステップ S T 1 0 4）。判定結果が Y E S である場合、交点算出部 2 は、実施の形態 1 で説明した式（1）～（3）に従って、Y 座標カウンタの値に対応するサブスキャンラインとエッジとの交点座標を算出し、実施の形態 1 と同様に、エッジの向き、交点 X 座標の整数部、サンプル数をエッジデータリスト 3 に保存する（ステップ S T 1 0 5）。判定結果が N O である場合、交点 X 座標の算出およびエッジデータリスト 3 へのエッジデータの登録は行わない。

30

【 0 0 6 6 】

次に、処理中のサブスキャンラインが最終サブスキャンラインである、即ち、Y 方向サブピクセルカウンタが（ $y_{sub} = (Y \text{ 方向サンプル数} - 1)$ ）の条件を満たすか否かを判定する（ステップ S T 1 0 6）。最終サブスキャンラインでない場合は、Y 方向サブピクセルカウンタ（ y_{sub} ）をインクリメントした後（ステップ S T 1 0 7）、ステップ S T 1 0 4 に戻り、ステップ S T 1 0 4 以降の処理を繰り返す。

【 0 0 6 7 】

ステップ S T 1 0 6 において、最終サブスキャンラインである場合には、処理中のエッジが図形の最終エッジであるか否かを判定する（ステップ S T 1 0 8）。ここでの判定結果が N O である場合は次のエッジに対してステップ S T 1 0 4 以下の処理を繰り返す。判定結果が Y E S である場合には、1 スキャンライン分の塗潰しを開始する（ステップ S T 1 0 9）。1 スキャンライン分の塗潰し手順は、実施の形態 1 における図 12 において y_{min} 、 y_{max} に対して現在処理中のスキャンライン Y 座標の値を代入したものと等しい。図 12 のステップ S T 1 7 に達すると、必ず 1 スキャンライン分の塗潰しが終了する。

40

【 0 0 6 8 】

1 スキャンライン分の塗潰しが終了すると、Y 座標が最終値（ y_{max} ）であるか否かの判定を行う（ステップ S T 1 1 0）。Y 座標が最終値でない場合には、Y 座標カウンタ（ y ）のインクリメント、Y 方向サブピクセルカウンタ（ y_{sub} ）のリセットを行った後（ステップ S T 1 1 1）、ステップ S T 1 0 4 以下の処理を繰り返す。Y 座標が最終値（ y_{max} ）である場合には、ベクトル図形の描画を終了する。

50

【 0 0 6 9 】

以上のように、実施の形態 2 のベクトル図形描画装置によれば、エッジとスキャンラインとの交点座標の算出を 1 スキャンライン分行う度に 1 スキャンライン分の塗潰しを行い、同様の処理をベクトル図形の包含矩形の Y 方向サイズ分だけ繰り返すようにしたので、エッジデータリストの使用領域を節約することができ、ベクトル図形描画装置としての回路規模を更に抑えることができる。

【図面の簡単な説明】

【 0 0 7 0 】

【図 1】この発明の実施の形態 1 によるベクトル図形描画装置を示す構成図である。

【図 2】ベクトル図形の塗潰しを行う場合のエッジ分割方式の説明図である。

10

【図 3】ベクトル図形の輪郭線の太線化を行う場合のエッジ分割方式の説明図である。

【図 4】サブピクセル分割によるアンチエイリアス処理の説明図である。

【図 5】エッジとサンプル点の位置関係の判定方法を示す説明図である。

【図 6】エッジデータリストの構成を示す説明図である。

【図 7】スキャンラインバッファの X アドレスの割り当てを示す説明図である。

【図 8】ベクトル図形の描画例と包含矩形を示す説明図である。

【図 9】スキャンラインバッファの情報を元に内外判定によりカバレッジを求める方法を示す説明図である。

【図 10】図 9 のスキャンラインバッファの値となる図形を示す説明図である。

【図 11】実施の形態 1 によるベクトル図形描画処理の塗潰し開始までの流れを示すフローチャートである。

20

【図 12】実施の形態 1 によるベクトル図形描画処理の塗潰し開始以降の流れを示すフローチャートである。

【図 13】 4×4 のサブピクセル分割で図形を塗潰す場合の一例を示す説明図である。

【図 14】図 13 の図形を描画する場合に各エッジから生成されるエッジデータを示す説明図である。

【図 15】図 13 の図形を描画する場合に、 $y_{sub} = 0$ のサブスキャンラインの処理手順を示す説明図である。

【図 16】図 13 の図形を描画する場合に、 $y_{sub} = 3$ のサブスキャンラインの処理手順を示す説明図である。

30

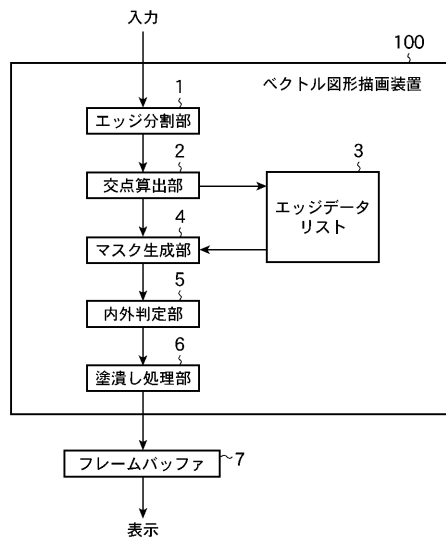
【図 17】実施の形態 2 によるベクトル図形描画処理の流れを示すフローチャートである。

【符号の説明】

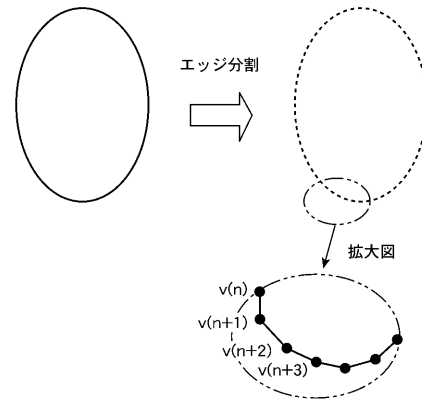
【 0 0 7 1 】

1 エッジ分割部、2 交点算出部、3 エッジデータリスト、4 マスク生成部、5 内外判定部、6 塗潰し処理部、100 ベクトル図形描画装置。

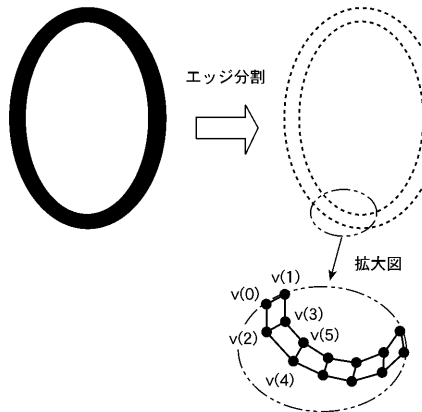
【図 1】



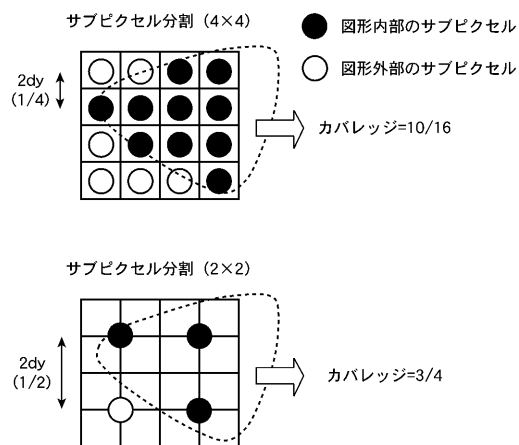
【図 2】



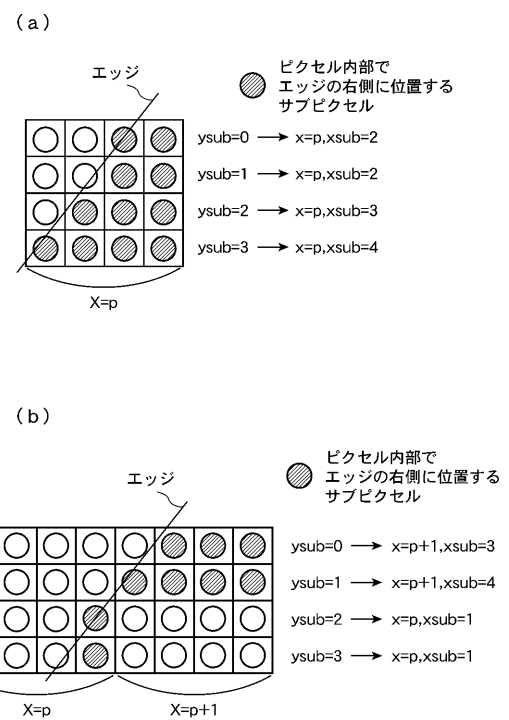
【図 3】



【図 4】



【図 5】



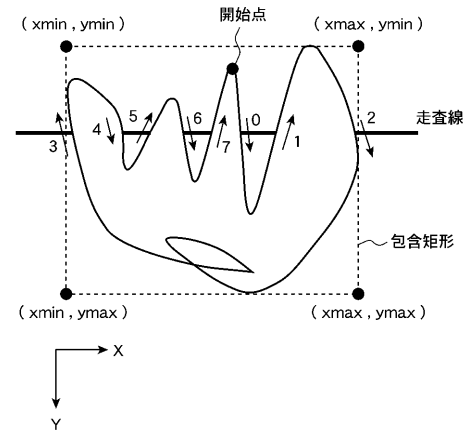
【図 6】

Y座標、サブピクセル					
y=0、ysub=0	データ0	データ1	データ2	データ3	...
y=0、ysub=1	データ0	データ1	データ2	データ3	...
y=1、ysub=0	データ0	データ1	データ2	データ3	...
y=1、ysub=1	データ0	データ1			...
y=2、ysub=0	データ0	データ1			...
y=2、ysub=1	データ0	データ1			...
⋮	⋮	⋮			
⋮	⋮	⋮			

【図 7】

SLBリニアアドレス	0	1	2	3	4	5	6	7
Xアドレス (X方向サブピクセル分割数=1)	0	1	2	3	4	5	6	7
Xアドレス (X方向サブピクセル分割数=2)	0	0	1	1	2	2	3	3
Xアドレス (X方向サブピクセル分割数=4)	0	0	0	0	1	1	1	1

【図 8】



【図 9】

(a)

0	0	0	0	+1	+1	+2	+2	+3	+4	+3	+2	+1	0	0	0	...
[0	0	1	1	2	2	3	3	4	4	5	5	6	6	7	7	...

↑ Xアドレス

内外判定結果

(b) カバレッジバッファ (イーブンオッド規制)

0	0	+2	0	+1	+1	+1	0	...
---	---	----	---	----	----	----	---	-----

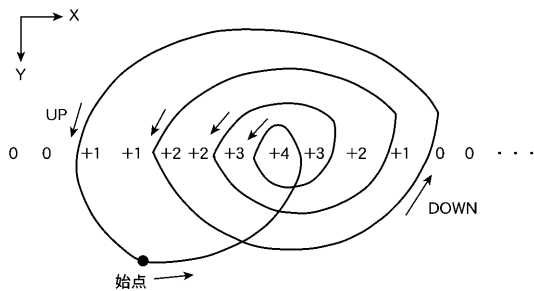
(c) カバレッジバッファ (ノンゼロ規制)

0	0	+2	+2	+2	+2	+1	0	...
---	---	----	----	----	----	----	---	-----

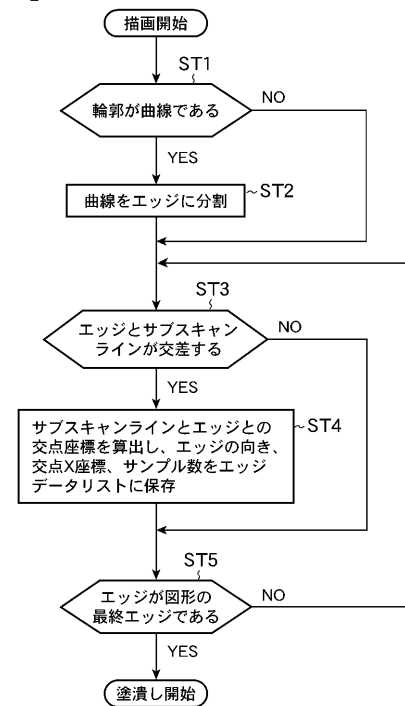
[0 1 2 3 4 5 6 7 ...]

↑ Xアドレス

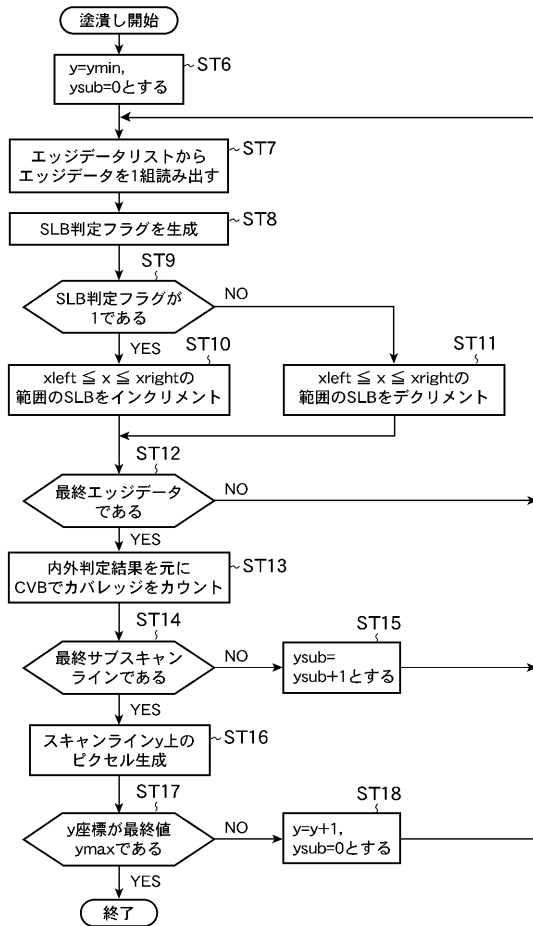
【図 10】



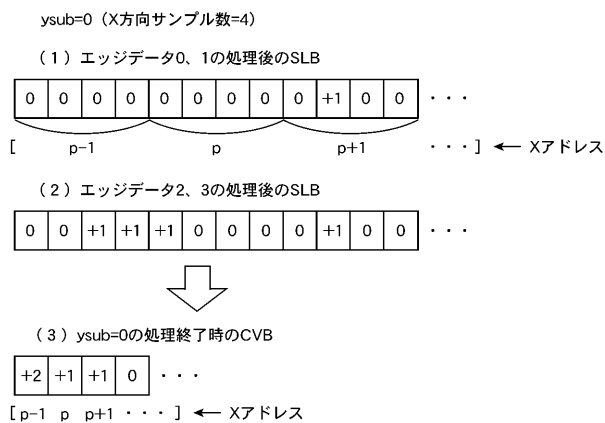
【図 11】



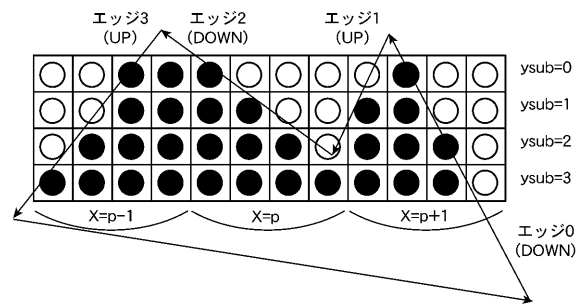
【図 12】



【図 15】



【図 13】

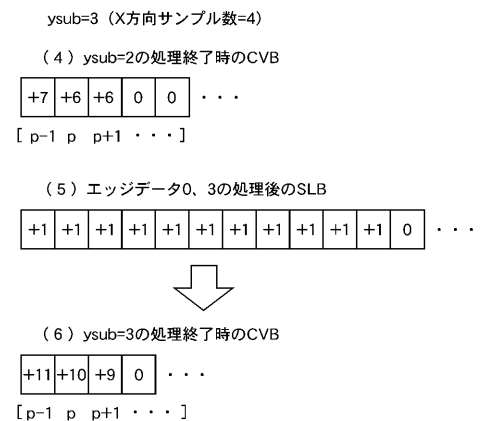


【図 14】

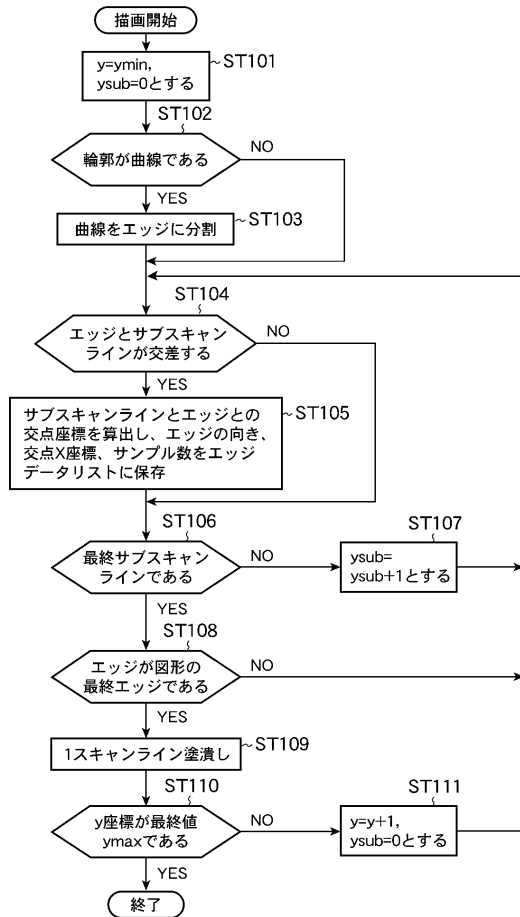
エッジデータ: (エッジの向き(0: DOWN, 1: UP), 交点X座標の整数部, サンプル数)

	エッジデータ0	エッジデータ1	エッジデータ2	エッジデータ3
ystub=0	(0, p+1, 2)	(1, p+1, 3)	(0, p, 3)	(1, p-1, 2)
ystub=1	(0, p+1, 2)	(1, p+1, 4)	(0, p, 2)	(1, p-1, 2)
ystub=2	(0, p+1, 1)	(1, p, 0)	(0, p, 1)	(1, p-1, 3)
ystub=3	(0, p+1, 1)			(1, p-1, 4)

【図 16】



【図 17】



フロントページの続き

- (72)発明者 根岸 博康
東京都千代田区丸の内二丁目7番3号 三菱電機株式会社内
- (72)発明者 兼子 貴弘
東京都千代田区丸の内二丁目7番3号 三菱電機株式会社内
- (72)発明者 大西 宏
東京都千代田区丸の内二丁目7番3号 三菱電機株式会社内
- (72)発明者 鳥居 晃
東京都千代田区丸の内二丁目7番3号 三菱電機株式会社内
- (72)発明者 石田 良平
東京都千代田区丸の内二丁目7番3号 三菱電機株式会社内
- Fターム(参考) 5B080 AA13 GA27 GA28