

(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2010-55611

(P2010-55611A)

(43) 公開日 平成22年3月11日(2010.3.11)

(51) Int.Cl.
G06T 11/00 (2006.01)F I
G06T 11/00 I O O Aテーマコード (参考)
5 B O 8 O

審査請求 未請求 請求項の数 7 O L (全 18 頁)

(21) 出願番号	特願2009-188003 (P2009-188003)	(71) 出願人	000005223
(22) 出願日	平成21年8月14日 (2009. 8. 14)		富士通株式会社
(31) 優先権主張番号	200810146771. 6		神奈川県川崎市中原区上小田中4丁目1番1号
(32) 優先日	平成20年8月29日 (2008. 8. 29)	(74) 代理人	100099759
(33) 優先権主張国	中国 (CN)		弁理士 青木 篤
		(74) 代理人	100119987
			弁理士 伊坪 公一
		(74) 代理人	100081330
			弁理士 樋口 外治
		(74) 代理人	100141254
			弁理士 榎原 正巳
		(74) 代理人	100113826
			弁理士 倉地 保幸

最終頁に続く

(54) 【発明の名称】 グラフィックラスタライゼーションにおける三角形トラバース方法及び装置

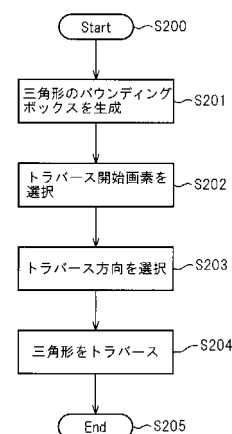
(57) 【要約】

【課題】非常に高速かつ効率的な三角形トラバースを実行できるグラフィックラスタライゼーションにおける三角形トラバース方法及び装置を提供する。

【解決手段】この方法は、三角形をカバーする画素からなる最小矩形であるバウンディングボックスを生成し、バウンディングボックスに基づいて三角形をトラバースするためのトラバース開始画素を選択し、トラバース開始画素と三角形の位置関係に基づいてトラバース方向を選択し、トラバース方向においてトラバース開始画素から三角形をトラバースすることを含む。三角形をトラバースすることは、トラバース開始画素及びトラバース方向に対する三角形のエッジ関数の傾きに基づいて次の走査開始画素を計算することを含む。

【選択図】 図 2

図2



【特許請求の範囲】**【請求項 1】**

グラフィックラスライゼーションにおける三角形トラバース方法であって、
三角形をカバーする画素からなる最小矩形であるバウンディングボックスを生成し、
前記バウンディングボックスに基づいて前記三角形をトラバースするためのトラバース
開始画素を選択し、

前記トラバース開始画素と前記三角形の位置関係に基づいてトラバース方向を選択し、
前記トラバース方向において前記トラバース開始画素から前記三角形をトラバースする
、ことを含み、

前記三角形をトラバースすることは、前記トラバース開始画素及び前記トラバース方向
に対する前記三角形のエッジ関数の傾きに沿って現在の走査開始画素から次の走査開始画
素を計算することを含む、
三角形トラバース方法。

10

【請求項 2】

前記三角形をトラバースすることは、前記傾きを求めるために前記トラバース方向にお
いて現在の行または列における前記三角形の内部にある最初の画素を検出することを含む
、請求項 1 に記載の三角形トラバース方法。

【請求項 3】

前記三角形をトラバースすることは、現在の行において前記三角形の内部にある前記最
初の画素と前に走査された行において前記三角形の内部にある前記最初の画素との距離、
または、現在の列において前記三角形の内部にある前記最初の画素と前に走査された列に
おいて前記三角形の内部にある前記最初の画素との距離を用いて前記傾きを求めること
により、前記次の走査開始画素を導出する、請求項 2 に記載の三角形トラバース方法。

20

【請求項 4】

前記三角形をトラバースすることは、現在の行または列の走査を終了するか否かを決定
することをさらに含み、現在の行または列における現在の画素が前記三角形の外側にあり
、かつ前記トラバース方向に沿って前の画素が前記三角形の内部にある場合に、現在の行
または列の走査を終了する、請求項 1 ～ 3 の何れか一項に記載の三角形トラバース方法。

【請求項 5】

前記トラバース方向を選択することは、水平方向または垂直方向のうち、前記トラバ
ース開始画素から前記三角形の内部にある画素の方向を、前記トラバース方向とする、請
求項 1 ～ 4 の何れか一項に記載の三角形トラバース方法。

30

【請求項 6】

前記三角形をトラバースすることは、前記トラバース方向において前記三角形の内部
にある画素が存在しない場合に前記三角形のトラバースを終了することを含む、請求項 1 ～
5 の何れか一項に記載の三角形トラバース方法。

【請求項 7】

グラフィックラスライゼーションにおける三角形トラバース装置であって、
三角形をカバーする画素からなる最小矩形であるバウンディングボックスを生成するバ
ウンディングボックス生成ブロックと、

40

前記バウンディングボックスに基づいて前記三角形をトラバースするためのトラバース
開始画素を選択するトラバース開始画素選択ブロックと、

前記トラバース開始画素と前記三角形の位置関係に基づいてトラバース方向を選択する
トラバース方向選択ブロックと、

前記トラバース方向において前記トラバース開始画素から前記三角形をトラバースする
トラバースブロックと、を有し、

前記トラバースブロックは、前記トラバース開始画素及び前記トラバース方向に対する
前記三角形のエッジ関数の傾きに沿って現在の走査開始画素から次の走査開始画素を計算
する手段を有する、

三角形トラバース装置。

50

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、一般に、イメージを表示する方法及び装置に関する。特に、本発明は、グラフィックラスタライゼーションにおける三角形トラバース方法及び装置に関する。

【背景技術】

【0002】

グラフィックラスタライゼーションは、ラスターグラフィックディスプレイへ伝達するために幾何学的要素（線、三角形、ポリゴン等）を画素に変換する手順である。三角形は、その有用な性質（常に凸であり、常に単平面に属し、複雑な2次元要素の作成が容易）のため、グラフィックラスタライゼーションにおいて最も重要な要素である。「三角形トラバース」は、三角形の内部にある全ての画素を検出する手順である。

10

【0003】

エッジ関数は、ハードウェアにおいて三角形をラスタライズする基礎となる。「エッジ関数」は、三角形の3本のエッジの3本の線の関数であり、三角形の3個の頂点により形成される。三角形内の画素は、この画素の座標における3個のエッジ関数値が全て正または全て負となるという特性を持つ。そのため、エッジ関数は画素が三角形の内部にあるか否かを決定するための有用なツールである。

【0004】

エッジ関数に基づく、多くの三角形トラバース方法が提案されている。例えば、バウンディングボックストラバース、バックトラックトラバース及びジグザグトラバースなどである。

20

【0005】

上記のトラバースアルゴリズムは、全て走査線に基づいたトラバース方法である。これらの基本的なアイデアは、三角形のバウンディングボックスを作成し、何らかの順序に従ってそのバウンディングボックスを走査することにある。

【発明の概要】

【0006】

一つの側面において、グラフィックラスタライゼーションにおける三角形トラバース方法が提供される。この方法は、三角形をカバーするピクセルからなる最小矩形であるバウンディングボックスを生成し、そのバウンディングボックスに基づいて三角形をトラバースするためのトラバース開始画素を選択し、トラバース開始画素と三角形間の位置関係に基づいてトラバース方向を選択し、そのトラバース方向においてトラバース開始画素から三角形をトラバースすることを含む。三角形をトラバースすることは、トラバース開始画素とトラバース方向に対する三角形のエッジ関数の傾きに基づいて、次の走査開始画素を計算することを含む。

30

【0007】

他の側面において、グラフィックラスタライゼーションにおける三角形トラバース装置が提供される。この装置は、三角形の頂点座標を受け取り、その頂点座標を用いることにより三角形のエッジ関数の係数を計算し、三角形のバウンディングボックスを生成し、かつトラバース開始画素及びトラバース方向を選択するセットアップブロックと、エッジ関数の係数、トラバース開始画素及びトラバース方向に基づいて三角形をトラバースするトラバースブロックとを有する。またトラバースブロックは、トラバース開始画素とトラバース方向に対する三角形のエッジ関数の傾きに基づいて、次の走査開始画素を計算する。

40

【0008】

ここに開示される方法及び装置は、トラバースにおいてできる限り三角形の外側の画素にアクセスしないことにより、非常に高速かつ効率的な三角形トラバースを実行できる。

【図面の簡単な説明】

【0009】

【図1】三角形を描写するシステムのブロック図である。

50

【図 2】一つの実施形態による三角形トラバース方法の概略を示すフローチャートである。

【図 3】図 1 に示したシステムのセットアップブロック 104 及びトラバースブロック 105 の詳細を示す図である。

【図 4】三角形のバウンディングボックスの一例を示す図である。

【図 5】三角形のバウンディングボックスを生成する方法を示すフローチャートである。

【図 6】四つの異なるケースのうちの一つにおいてそれぞれ選択されるトラバース開始画素を示す図である。

【図 7】トラバース方向を選択する方法を示すフローチャートである。

【図 8】二つの異なるケースのうちの一つにおいてそれぞれ選択されるトラバース方向を示す図である。

【図 9】三角形の完全なトラバースを示す一例の図である。

【図 10】次の走査行 / 列における走査開始画素の計算方法を示すフローチャートである。

【発明を実施するための形態】

【0010】

好ましい実施形態についての以下の記述において、説明目的のため、多数の具体的なディテールが本発明を深く理解するために記述されている。しかしながら、これらの具体的なディテールなしに本発明を実施してもよいことは、当業者にとって明らかであろう。

【0011】

三角形トラバース方法の効率は、主として、三角形の外側にある画素にアクセスする冗長なコストに依存する。そのため、この実施形態の目的は、極力三角形の外側の画素にアクセスしないようにすることにある。

【0012】

図 1 は、三角形を描写するシステムのブロック図である。このシステムは、頂点メモリ 101 と、変換ブロック 102 と、投影ブロック 103 と、セットアップブロック 104 と、トラバースブロック 105 と、フレームバッファ 106 と、ディスプレイ 107 とを有する。

【0013】

まず、頂点情報（物体座標）が頂点メモリ 101 から変換ブロック 102 へ読み出される。そして変換ブロック 102 は、頂点座標を変換し、投影ブロック 103 が変換された頂点座標をデバイス座標に投影する。セットアップブロック 104 及びトラバースブロック 105 は、デバイス座標を使用して、どの画素が三角形の内部にあるかを計算し、それらの画素のパラメータ（例えば、色 / テクスチャ）を計算する。そしてそのパラメータはフレームバッファ 106 に書き込まれる。ディスプレイ 107 は、フレームバッファ 106 に書き込まれた内容に基づいて表示する。

【0014】

図 2 は、一つの実施形態による三角形トラバース方法の概略を示すフローチャートである。この三角形トラバース方法は、バウンディングボックスを生成し（ステップ S201）、トラバース開始画素を選択し（ステップ S202）、トラバース方向を選択し（ステップ S203）、そして三角形をトラバースする（ステップ S204）。

【0015】

以下、三角形トラバース処理方法及び装置を詳しく記述する。

【0016】

図 3 は、図 1 に示したシステムのセットアップブロック 104 及びトラバースブロック 105 の詳細を示す図である。セットアップブロック 104 は、エッジ関数計算ブロック 301 と、バウンディングボックス生成ブロック 302 と、トラバース開始画素及びトラバース方向選択ブロック 303 とを有する。またトラバースブロック 105 は、開始画素登録ブロック 304 と、走査ブロック 305 と、次開始画素計算ブロック 306 と、描画判定ブロック 307 と、パラメータ計算ブロック 308 とを有する。

【 0 0 1 7 】

セットアップブロック 1 0 4 のエッジ関数計算ブロック 3 0 1 は、投影ブロック 1 0 3 から三角形の 3 個の頂点の座標 $V0(x0,y0)$ 、 $V1(x1,y1)$ 及び $V2(x2,y2)$ を受け取る。 $i=0,1,2$ に対して、 (xi,yi) は浮動小数点値である。また、3 個の頂点 $V0$ 、 $V1$ 及び $V2$ は、三角形において時計回りに選択される。3 個の頂点の座標に基づいて、エッジ関数計算ブロック 3 0 1 は、三角形の 3 本のエッジのエッジ関数を計算する。

【 数 1 】

$$e0(x,y)=a0*x+b0*y+c0$$

10

$$e1(x,y)=a1*x+b1*y+c1$$

$$e2(x,y)=a2*x+b2*y+c2$$

【 0 0 1 8 】

セットアップブロック 1 0 4 のバウンディングボックス生成ブロック 3 0 2 は、投影ブロック 1 0 3 から三角形の 3 個の頂点の座標を受け取り、その頂点座標を用いて三角形のバウンディングボックスを生成する。三角形のバウンディングボックスは、その三角形をカバーする画素からなる最小矩形であり、その矩形の 4 個のコーナーにおける画素の座標 $(xmin,ymin)$ 、 $(xmin,ymax)$ 、 $(xmax,ymin)$ 及び $(xmax,ymax)$ により表される。

20

【 0 0 1 9 】

図 4 は、三角形のバウンディングボックスの一例を示す図である。以下、三角形のバウンディングボックスを生成する処理について、図 4 に示した例を参照しつつ説明する。三角形 2 0 1 の 3 個の頂点は、それぞれ、 $(x0,y0)$ 、 $(x1,y1)$ 及び $(x2,y2)$ である。3 個の頂点の座標から、以下の中間値が得られる。

【 数 2 】

$$fxmin = \min(x0,x1,x2), \quad fxmax = \max(x0,x1,x2)$$

$$fymin = \min(y0,y1,y2), \quad fymax = \max(y0,y1,y2)$$

30

ここで関数 $\min()$ 及び $\max()$ は、3 個の変数の最小値及び最大値を計算するために使用される。 $fxmin$ 、 $fxmax$ 、 $fymin$ 及び $fymax$ の全ては、浮動小数点値である。

【 0 0 2 0 】

画素 2 0 3 は座標の整数値に配置されるので、上記の中間値は、以下の操作により整数値に変換される。

【 数 3 】

$$xmin = \text{ceil}(fxmin), \quad xmax = \text{floor}(fxmax)$$

$$ymin = \text{ceil}(fymin), \quad ymax = \text{floor}(fymax)$$

40

ここで $\text{ceil}(\text{float } a)$ 関数は、 a 以上の最小整数を返す関数である。例えば、 $\text{ceil}(1.5)=2$ である。また $\text{floor}(\text{float } a)$ 関数は、 a 以下の最大整数を返す関数である。例えば、 $\text{floor}(1.5)=1$ である。

【 0 0 2 1 】

50

三角形の 1 個以上の頂点は、表示面（図 4 では図示せず）の外側に存在する可能性があるため、クリップ操作が必要となる。表示面の範囲がクリップウインドウ[clipxmin,clipxmax]、[clipymin,clipymax]により定義されるとすると、クリップ操作は以下になる。

```
if(xmin < clipxmin) xmin = clipxmin;
if(ymin < clipymin) ymin = clipymin;
if(xmax > clipxmax) xmax = clipxmax;
if(ymax > clipymax) ymax = clipymax;
```

【 0 0 2 2 】

クリップ操作は三角形のバウンディングボックスの 4 個のコーナーにおける画素の座標、すなわち、(xmin,ymin)、(xmin,ymax)、(xmax,ymin) 及び (xmax,ymax) を与え、それらはバウンディングボックスを定義する。

【 0 0 2 3 】

エッジ関数計算ブロック 3 0 1 により計算された三角形のエッジ関数とバウンディングボックス生成ブロック 3 0 2 により生成された三角形のバウンディングボックスに基づいて、セットアップブロック 1 0 4 のトラバース開始画素及びトラバース方向選択ブロック 3 0 3 はトラバース開始画素及びトラバース方向を決定する。

【 0 0 2 4 】

特に、図 5 に示した処理によれば、トラバース開始画素及びトラバース方向選択ブロック 3 0 3 は、バウンディングボックス生成ブロック 3 0 2 から受け取った fxmin、fxmax、fymin 及び fymax を用いることにより、バウンディングボックス生成ブロック 3 0 2 において計算された 4 個のコーナーにおける画素(xmin,ymin)、(xmin,ymax)、(xmax,ymin) 及び (xmax,ymax) からトラバース開始画素を選択する。トラバース開始画素は、三角形のバウンディングボックスの 4 個のコーナーにおける画素(xmin,ymin)、(xmin,ymax)、(xmax,ymin) 及び (xmax,ymax)（すなわち、左上、左下、右上、右下）のうちの一つであり、三角形の 3 個の頂点の座標関係に依存する。その選択を示すコードは以下の通りである。

```
if((x0 == fxmin && y0 == fymin) || (x1 == fxmin && y1 == fymin) || (x2 == fxmin && y2 == fymin)) startpos = topleft ( ステップ S 5 0 1 及び S 5 0 2 );
else if((x0 == fxmax && y0 == fymin) || (x1 == fxmax && y1 == fymin) || (x2 == fxmax && y2 == fymin)) startpos = topright ( ステップ S 5 0 3 及び S 5 0 4 );
else if((x0 == fxmin && y0 == fymax) || (x1 == fxmin && y1 == fymax) || (x2 == fxmin && y2 == fymax)) startpos = bottomleft ( ステップ S 5 0 5 及び S 5 0 6 );
;
else if((x0 == fxmax && y0 == fymax) || (x1 == fxmax && y1 == fymax) || (x2 == fxmax && y2 == fymax)) startpos = bottomright ( ステップ S 5 0 7 及び S 5 0 8 );
```

なお、上記のコードにおいて、変数startposはトラバース開始画素の位置を表す。また、topleft、topright、bottomleft、bottomrightは、それぞれ、バウンディングボックスの左上、右上、左下、右下のコーナーを表す。

【 0 0 2 5 】

図 6 は、四つの異なるケースのうちの一つにおいてそれぞれ選択されるトラバース開始画素を示す図である。図 6 (a)、(b)、(c) 及び (d) は、それぞれ、図 5 におけるステップ S 5 0 2、S 5 0 4、S 5 0 6 及び S 5 0 8 に対応する。

【 0 0 2 6 】

図 5 は、所定の順序、すなわち、左上、右上、左下、右下の順序でトラバース開始画素を選択しているが、他の順序に従ってトラバース開始画素を選択してもよい。また、図 6 は、三角形のただ 1 個の頂点のみがバウンディングボックスの 1 個のコーナーの画素に含まれる例として三角形を示した。しかし、三角形の 2 個以上の頂点とその三角形のバウンディングボックスのコーナーの画素に含まれてもよい。

【 0 0 2 7 】

図示されたように、クリップ操作の前に、バウンディングボックスのコーナーの一つにおける画素に含まれる三角形の頂点が少なくとも一つ存在する。バウンディングボックスがクリップ操作の対象でない場合、トラバース開始画素は、バウンディングボックスの4個のコーナーにある画素のうち、三角形の頂点が含まれる画素の何れかである。バウンディングボックスがクリップ操作の対象となる場合、トラバース開始画素は、クリップ操作の前においてトラバース開始画素が存在するコーナーに対応するコーナーにある画素の一つである。

【0028】

また、トラバース開始画素及びトラバース方向選択ブロック303は、エッジ関数計算ブロック301から受け取ったエッジ関数の係数と、バウンディングボックスの4個のコーナーにある画素(xmin,ymin)、(xmin,ymax)、(xmax,ymin)及び(xmax,ymax)に基づいて、トラバース方向を選択する。二つのトラバース方向が考えられる。すなわち、水平方向(行)と垂直方向(列)である。トラバース開始画素から三角形の内部に存在する画素が水平ラインにある場合、トラバース方向を水平とし、そうでなければ、トラバース方向を垂直とする。

10

【0029】

三角形に対するトラバース開始画素の位置に基づいて、トラバース開始画素から三角形の内部に存在する画素が水平ラインにあるか否かを決定できる。三角形に対するトラバース開始画素の位置は、トラバース開始画素における三角形の3個のエッジ関数の値から得られる。画素が三角形の外側にある場合、この画素における3個のエッジ関数の値を用いて、この画素が三角形の左/右/上/下の何れにあるか否かを決定できる。

20

【0030】

以下の4個の式を用いて、画素p(x,y)と三角形の位置関係を決定する。

```
(1) InLeftTriangle(x,y)
{
    if((e0(x,y) < 0 && a0 >= 0) || (e1(x,y) < 0 && a1 >= 0) || (e2(x,y) < 0 && a2 >= 0))
        return true; // (三角形の左側にある)
    else return false; // (三角形の左側でない)
}

(2) InRightTriangle(x,y)
{
    if((e0(x,y) < 0 && a0 <= 0) || (e1(x,y) < 0 && a1 <= 0) || (e2(x,y) < 0 && a2 <= 0))
        return true; // (三角形の右側にある)
    else return false; // (三角形の右側でない)
}

(3) InTopTriangle(x,y)
{
    if((e0(x,y) < 0 && b0 <= 0) || (e1(x,y) < 0 && b1 <= 0) || (e2(x,y) < 0 && b2 <= 0))
        return true; // (三角形の上方にある)
    else return false; // (三角形の上方でない)
}

(4) InBottomTriangle(x,y)
{
    if((e0(x,y) < 0 && b0 >= 0) || (e1(x,y) < 0 && b1 >= 0) || (e2(x,y) < 0 && b2 >= 0))
        return true; // (三角形の下方にある)
    else return false; // (三角形の下方でない)
}
```

30

40

【0031】

図7は、トラバース方向を選択する方法を示すフローチャートである。互いに異なるトラバース開始画素に対して、この方法はバウンディングボックスの4個のコーナーにある

50

画素に基づいてトラバース方向を選択する。図 7 に示したステップ S 7 0 1、S 7 0 5、S 7 0 9 及び S 7 1 3 は、トラバース開始画素の四つの異なるケースを表す。四つのブランチ S 7 0 2 - S 7 0 4、S 7 0 6 - S 7 0 8、S 7 1 0 - S 7 1 2 及び S 7 1 4 - S 7 1 6 は、それぞれ、上記の四つの異なるケースに対応し、トラバース方向を得る決定をなす。

【 0 0 3 2 】

図 8 (a) 及び (b) は、図 7 に示した処理により得られた二つの異なるトラバース方向の例を示す図である。図 8 (a) は、図 7 に示したステップ S 7 0 1、S 7 0 2 及び S 7 0 4 による水平なトラバース方向を示し、一方、図 8 (b) は、図 7 に示したステップ S 7 0 1 - S 7 3 0 による垂直なトラバース方向を示す。

10

【 0 0 3 3 】

トラバース開始画素がチェックされてトラバース方向を選択した、図 7 に示した順序と、図 8 に示した実例は、全て例示である。図 7 に示した順序と異なる順序、及び図 8 に示した例と異なる例とすることも可能である。

【 0 0 3 4 】

この時点で、セットアップブロック 1 0 4 は、エッジ関数、トラバース開始画素及びトラバース方向を算出している。トラバースブロック 1 0 5 は、セットアップブロック 1 0 4 により準備された情報に基づいて三角形をトラバースする。

【 0 0 3 5 】

トラバースブロック 1 0 5 の開始画素登録ブロック 3 0 4 は、先ず、セットアップブロック 1 0 4 のトラバース開始画素及びトラバース方向選択ブロック 3 0 3 からトラバース開始画素を受け取り、そのトラバース開始画素を現在の走査開始画素として登録する。走査ブロック 3 0 5 は開始画素登録ブロック 3 0 4 から現在の走査開始画素を受け取り、セットアップブロック 1 0 4 のトラバース開始画素及びトラバース方向選択ブロック 3 0 3 からトラバース方向を受け取る。そして走査ブロック 3 0 5 は、そのトラバース方向において現在の走査開始画素から三角形の現在の行 / 列を走査する。そのトラバースにおいて、各行 / 列の走査開始画素を、極力三角形の外側の画素にアクセスしないよう、適応的に調節して、トラバース速度を向上する。

20

【 0 0 3 6 】

走査において、走査ブロック 3 0 5 は、走査された画素を描画判定ブロック 3 0 7 へ送る。描画判定ブロック 3 0 7 は、画素が三角形の内部にあるか否かを決定する。その画素の座標における三角形の 3 個のエッジ関数の値全てが正である場合、その画素は三角形の内部にあると決定できる。描画判定ブロック 3 0 7 は、走査ブロック 3 0 5 から受け取った現在の行 / 列において三角形の内部にある最初の画素を、次開始画素計算ブロック 3 0 6 へ送る。現在の行 / 列に三角形の内部にある画素が存在しない場合、トラバースを終了し、トラバースを終了させることを示す命令を走査ブロック 3 0 5 へ送る。

30

【 0 0 3 7 】

また、描画判定ブロック 3 0 7 は、走査ブロック 3 0 5 から受け取った各画素 $p(x,y)$ に対して、現在の行 / 列の走査を終了するか否かを決定する。描画判定ブロック 3 0 7 が現在の行 / 列の走査を終了しないと決定した場合、描画判定ブロック 3 0 7 は画素 $p(x,y)$ をパラメータ計算ブロック 3 0 8 へ送る。描画判定ブロック 3 0 7 が現在の行 / 列の走査を終了すると決定した場合、描画判定ブロック 3 0 7 は、現在の行 / 列の走査を終了させることを示す命令を走査ブロック 3 0 5 へ送る。

40

【 0 0 3 8 】

画素 $p(x,y)$ が以下の条件の一つを満たす場合、描画判定ブロック 3 0 7 は現在の行 / 列の走査を終了すると決定する。

```

I (startpos == topleft || startpos == bottomleft) && (direction == horizontal) && (InRightTriangle(x,y) == True);
II (startpos == topright || startpos == bottomright) && (direction == horizontal) && (InLeftTriangle(x,y) == True);

```

50

```
III (startpos == topleft || startpos == topright) && (direction == vertical)
&& (InBottomTriangle(x,y) == True);
```

```
IV (startpos == bottomleft || startpos == bottomright) && (direction == vertical)
&& (InTopTriangle(x,y) == True)
```

なお、変数startposは、トラバース開始画素の位置を表す。また、topleft、topright、bottomleft、bottomrightは、それぞれ、バウンディングボックスの左上、右上、左下、右下のコーナーを表す。また変数directionは、トラバース方向を表し、horizontal及びverticalは、それぞれ、水平及び垂直を表す。

【 0 0 3 9 】

描画判定ブロック 3 0 7 から画素を受け取る度に、次開始画素計算ブロック 3 0 6 は次の行 / 列の走査開始画素を計算する。次の行 / 列の走査開始画素は、エッジ関数の係数に基づいて適応的に決定される。すなわち、現在の行 / 列において三角形の内部となる最初の画素と、次の行 / 列において三角形の内部となる最初の画素との距離を計算することにより、次の行 / 列の走査開始画素が得られる。

【 0 0 4 0 】

図 9 は、次の走査画素の選択方法及びこの実施形態によるトラバース方法を簡潔に図示する、三角形の完全なトラバースを示す一例の図である。図 9 において、トラバース開始画素は左上の画素であり、トラバース方向は水平方向である。またトラバース開始画素は、第 1 の走査行における走査開始画素でもある。現在の行において三角形の内部となる最初の画素の座標を(xc,yc)とし、前の行において三角形の内部となる最初の画素の座標を(xp,yp)とすると、次の行における走査開始画素(xn,yn)は、以下のように計算される。

【 数 4 】

$$\text{deltax} = xc - xp - 1$$

$$\text{deltay} = 1$$

$$xn = xc + \text{deltax}$$

$$yn = yc + \text{deltay}$$

【 0 0 4 1 】

同様に、三角形及び異なるトラバース方向に対して、異なる位置における走査開始画素の組み合わせに関しても、次の行 / 列における走査開始画素を、エッジ関数の係数を用いることにより適応的に選択できる。

【 0 0 4 2 】

図 1 0 は、次の行 / 列における走査開始画素の計算方法を示すフローチャートである。図示した四つのブランチ S 1 0 0 1 - S 1 0 0 7、S 1 0 0 8 - S 1 0 1 4、S 1 0 1 5 - S 1 0 2 1 及び S 1 0 2 2 - S 1 0 2 7 を使用して、走査開始画素とトラバース方向の四つの異なる組み合わせに対して、それぞれ、次の行 / 列における走査開始画素を計算する。

【 0 0 4 3 】

図示したように、トラバース方向が水平であれば、deltaxはxcとxp間の距離である。xc= xpであれば、deltax=0となる。そうでなければ、deltax=|xc-xp|-1となる。deltayの値は、トラバース開始画素がバウンディングボックスの上端の行にあるか下端の行にあるかに因る。トラバース開始画素が上端の行にあれば、deltay=1である。そうでなければ、deltay=-1である。

【 0 0 4 4 】

トラバース方向が垂直であれば、deltayはycとyp間の距離である。yc=ypであれば、deltay=0となる。そうでなければ、deltay=|yc-yp|-1となる。deltaxの値は、トラバース開始画素がバウンディングボックスの左端の列にあるか右端の列にあるかに因る。トラバース

10

20

30

40

50

ス開始画素が左端の列にあれば、 $\text{deltax}=1$ である。そうでなければ、 $\text{deltax}=-1$ である。

【 0 0 4 5 】

図 1 0 に示した四つのブランチの順序は単なる例示にすぎない。他の順序とすることも可能である。

【 0 0 4 6 】

上記の方法によれば、次開始画素計算ブロック 3 0 6 は、次の行 / 列における走査開始画素を計算し、その走査開始画素を開始画素登録ブロック 3 0 4 へ送る。開始画素登録ブロック 3 0 4 は、次開始画素計算ブロック 3 0 6 から受け取ったその走査開始画素を現在の走査開始画素として登録する。

【 0 0 4 7 】

描画判定ブロック 3 0 7 から現在の行 / 列の終了を示す命令を受け取ると、走査ブロック 3 0 5 は、開始画素登録ブロック 3 0 4 から以前に登録された現在の走査開始画素を取り出し、次の行 / 列の走査を開始する。

【 0 0 4 8 】

この走査処理は、トラバース処理が終わるまで繰り返される。

【 0 0 4 9 】

描画判定ブロック 3 0 7 から受け取った各画素 $p(x,y)$ について、パラメータ計算ブロック 3 0 8 は、色、テクスチャといったその画素のパラメータを計算し、算出されたパラメータ及び画素をフレームバッファ 1 0 6 へ送る。

【 0 0 5 0 】

これまで、好ましい実施形態について述べてきた。本発明を特定の実施形態において説明してきたが、本発明は、ハードウェア、ソフトウェア、ファームウェアまたはそれらの組み合わせにおいて実行でき、また本発明は、システム、サブシステム及びそのパーツまたはサブパーツに適用できる。ソフトウェアで実行される場合、この実施形態の各ブロックは、実質的に、プロセッサ上で実行されることにより、所要のタスクを達成するコードセグメントとなる。また、ハードウェアで実行される場合、この実施形態のブロックのそれぞれは、そのブロックの機能を実現する演算回路またはメモリ回路を有する。

【 0 0 5 1 】

以上説明した実施形態及びその変形例に関し、更に以下の付記を開示する。

(付記 1)

グラフィックラスタライゼーションにおける三角形トラバース方法であって、
三角形をカバーする画素からなる最小矩形であるバウンディングボックスを生成し、
前記バウンディングボックスに基づいて前記三角形をトラバースするためのトラバース開始画素を選択し、

前記トラバース開始画素と前記三角形の位置関係に基づいてトラバース方向を選択し、
前記トラバース方向において前記トラバース開始画素から前記三角形をトラバースすることを含み、

前記三角形をトラバースすることは、前記トラバース開始画素及び前記トラバース方向に対する前記三角形のエッジ関数の傾きに沿って現在の走査開始画素から次の走査開始画素を計算することを含む、
三角形トラバース方法。

(付記 2)

前記バウンディングボックスがクリップ操作の対象でない場合、前記トラバース開始画素は、前記バウンディングボックスの 4 個のコーナーの画素のうち、前記三角形の頂点が含まれるコーナーの画素の一つであり、

前記バウンディングボックスがクリップ操作の対象である場合、前記トラバース開始画素は、前記クリップ操作が行われる前において、前記三角形の頂点が含まれる前記バウンディングボックスのコーナーの一つに対応する、前記バウンディングボックスの 4 個のコーナーの画素のうちの一つのコーナーの画素である、付記 1 に記載の三角形トラバース方法。

10

20

30

40

50

(付記 3)

前記トラバース方向を選択することは、水平方向または垂直方向のうち、前記トラバース開始画素から前記三角形の内部にある画素の方向を、前記トラバース方向とする、付記 1 に記載の三角形トラバース方法。

(付記 4)

前記三角形をトラバースすることは、前記傾きを求めるために前記トラバース方向において現在の行または列における前記三角形の内部にある最初の画素を検出することを含む、付記 1 に記載の三角形トラバース方法。

(付記 5)

前記三角形をトラバースすることは、現在の行において前記三角形の内部にある前記最初の画素と前に走査された行において前記三角形の内部にある前記最初の画素との距離、または、現在の列において前記三角形の内部にある前記最初の画素と前に走査された列において前記三角形の内部にある前記最初の画素との距離を用いて前記傾きを求めることにより、前記次の走査開始画素を導出する、付記 4 に記載の三角形トラバース方法。

10

(付記 6)

前記三角形をトラバースすることは、現在の行または列の走査を終了するか否かを決定することをさらに含み、現在の行または列における現在の画素が前記三角形の外側にあり、かつ前記トラバース方向に沿って前の画素が前記三角形の内部にある場合に、現在の行または列の走査を終了する、付記 1 に記載の三角形トラバース方法。

20

(付記 7)

前記三角形をトラバースすることは、前記トラバース方向において前記三角形の内部にある画素が存在しない場合に前記三角形のトラバースを終了することを含む、付記 1 に記載の三角形トラバース方法。

(付記 8)

グラフィックラスタライゼーションにおける三角形トラバース装置であって、
三角形をカバーする画素からなる最小矩形であるバウンディングボックスを生成するバウンディングボックス生成ブロックと、

前記バウンディングボックスに基づいて前記三角形をトラバースするためのトラバース開始画素を選択するトラバース開始画素選択ブロックと、

前記トラバース開始画素と前記三角形の位置関係に基づいてトラバース方向を選択するトラバース方向選択ブロックと、

30

前記トラバース方向において前記トラバース開始画素から前記三角形をトラバースするトラバースブロックと、を有し、

前記トラバースブロックは、前記トラバース開始画素及び前記トラバース方向に対する前記三角形のエッジ関数の傾きに沿って現在の走査開始画素から次の走査開始画素を計算する手段を有する、

三角形トラバース装置。

(付記 9)

前記バウンディングボックスがクリップ操作の対象でない場合、前記トラバース開始画素は、前記バウンディングボックスの 4 個のコーナーの画素のうち、前記三角形の頂点が含まれるコーナーの画素の一つであり、

40

前記バウンディングボックスがクリップ操作の対象である場合、前記トラバース開始画素は、前記クリップ操作が行われる前において、前記三角形の頂点が含まれる前記バウンディングボックスのコーナーの一つに対応する、前記バウンディングボックスの 4 個のコーナーの画素のうちの一つのコーナーの画素である、付記 8 に記載の三角形トラバース装置。

(付記 10)

前記トラバース方向選択ブロックは、水平方向または垂直方向のうち、前記トラバース開始画素から前記三角形の内部にある画素の方向を、前記トラバース方向とする、付記 8 に記載の三角形トラバース装置。

50

(付記 1 1)

前記トラバースブロックは、前記傾きを求めるために前記トラバース方向において現在の行または列における前記三角形の内部にある最初の画素を検出する手段を有する、付記 8 に記載の三角形トラバース装置。

(付記 1 2)

前記トラバースブロックは、現在の行において前記三角形の内部にある前記最初の画素と前に走査された行において前記三角形の内部にある前記最初の画素との距離、または、現在の列において前記三角形の内部にある前記最初の画素と前に走査された列において前記三角形の内部にある前記最初の画素との距離を用いて前記傾きを求めることにより、前記次の走査開始画素を導出する、付記 1 1 に記載の三角形トラバース装置。

10

(付記 1 3)

前記トラバースブロックは、現在の行または列の走査を終了するか否かを決定する手段を有し、現在の行または列における現在の画素が前記三角形の外側にあり、かつ前記トラバース方向に沿って前の画素が前記三角形の内部にある場合に、現在の行または列の走査を終了する、付記 8 に記載の三角形トラバース装置。

(付記 1 4)

前記トラバースブロックは、前記トラバース方向において前記三角形の内部にある画素が存在しない場合に前記三角形のトラバースを終了する手段をさらに有する、付記 8 に記載の三角形トラバース装置。

20

【符号の説明】

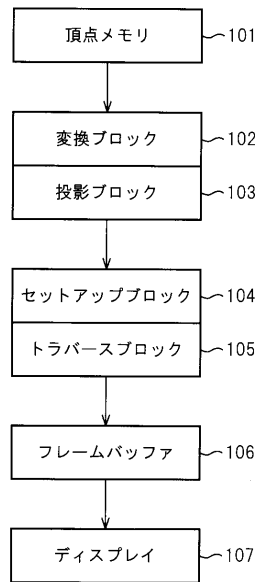
【0 0 5 2】

- 1 0 1 頂点メモリ
- 1 0 2 変換ブロック
- 1 0 3 投影ブロック
- 1 0 4 セットアップブロック
- 1 0 5 トラバースブロック
- 1 0 6 フレームバッファ
- 1 0 7 ディスプレイ
- 3 0 1 エッジ関数計算ブロック
- 3 0 2 バウンディングボックス生成ブロック
- 3 0 3 トラバース開始画素及びトラバース方向選択ブロック
- 3 0 4 開始画素登録ブロック
- 3 0 5 走査ブロック
- 3 0 6 次開始画素計算ブロック
- 3 0 7 描画判定ブロック
- 3 0 8 パラメータ計算ブロック

30

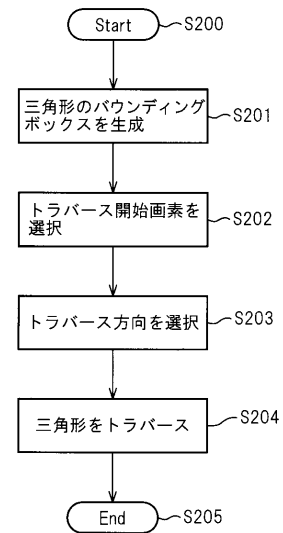
【図 1】

図1



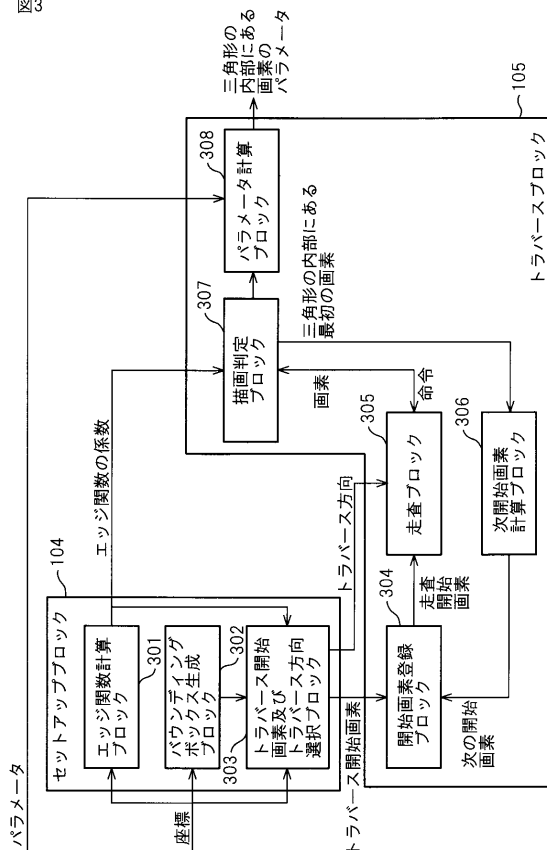
【図 2】

図2



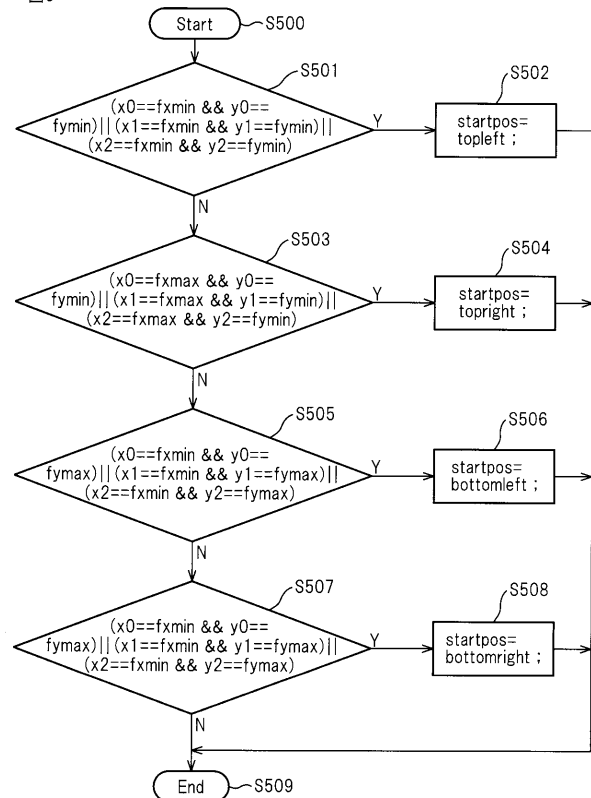
【図 3】

図3

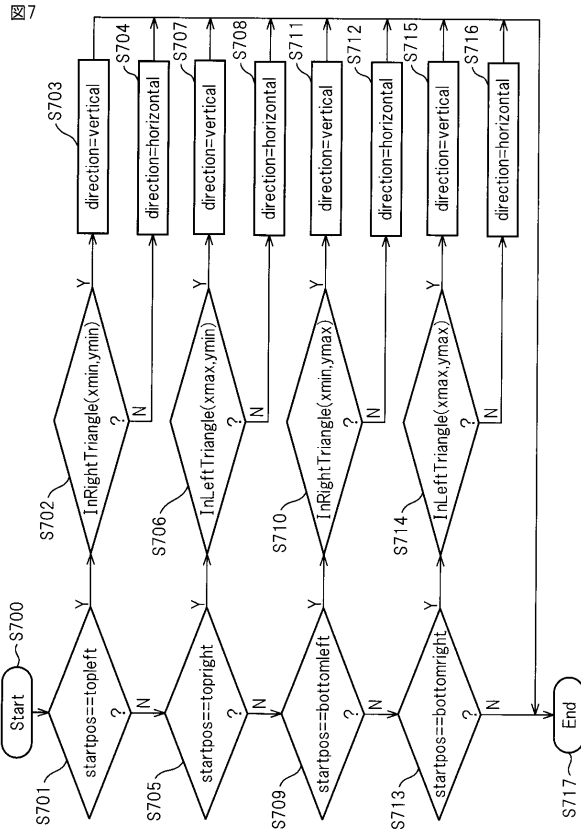


【図 5】

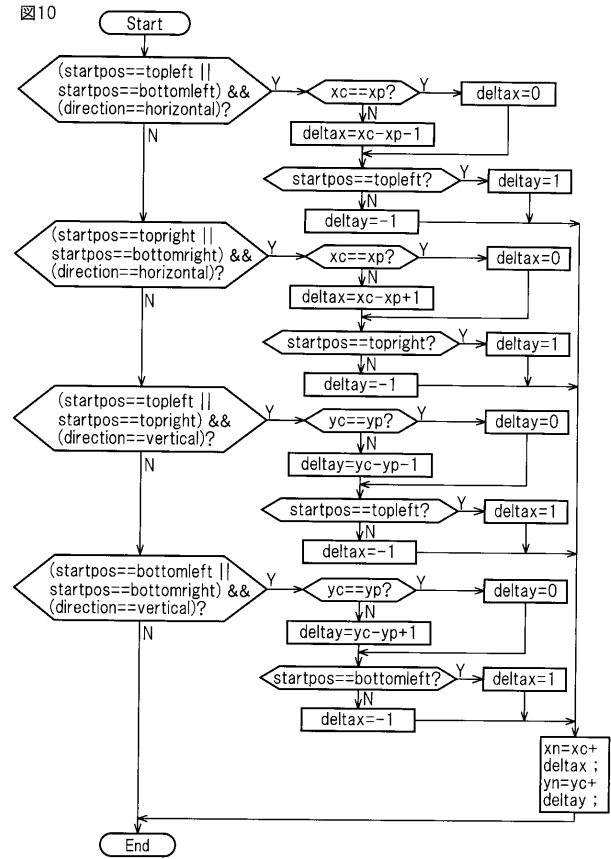
図5



【図 7】

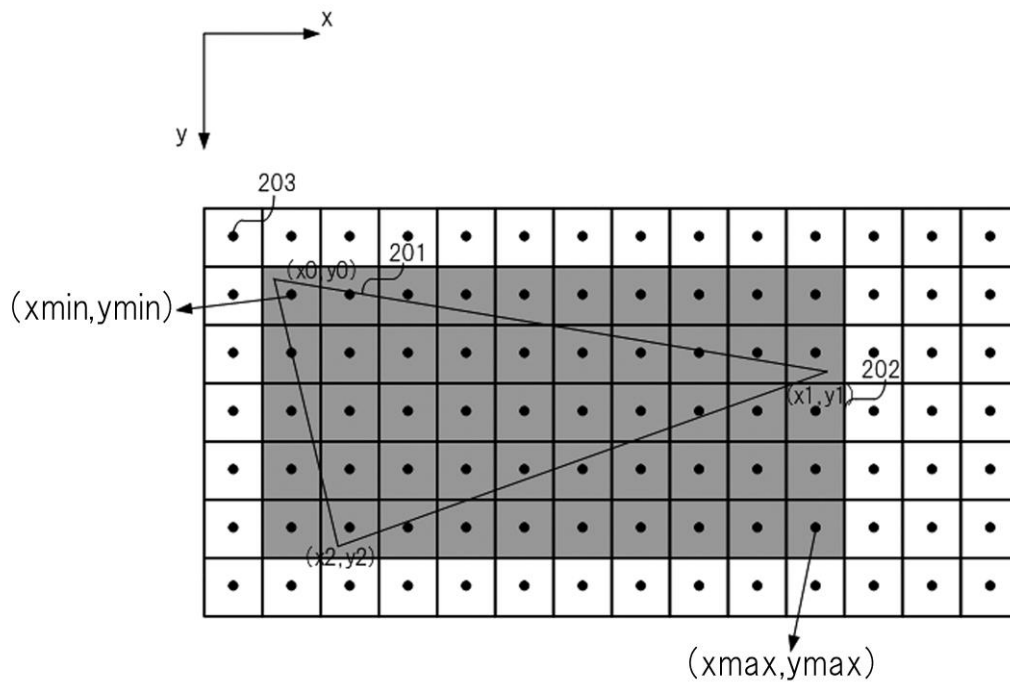


【図 10】



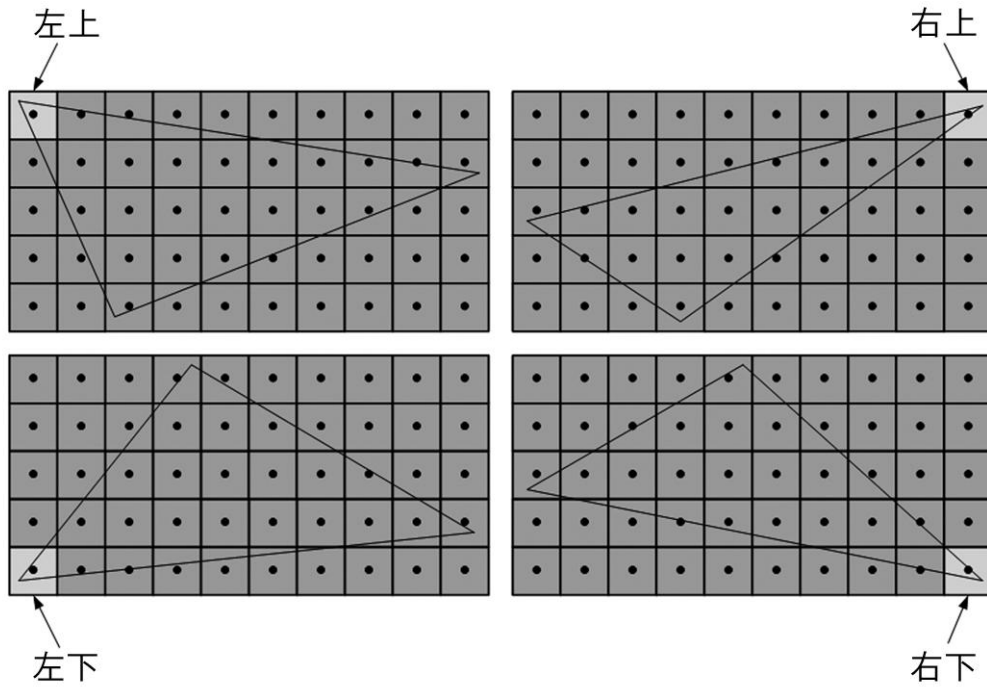
【図 4】

図 4



【 図 6 】

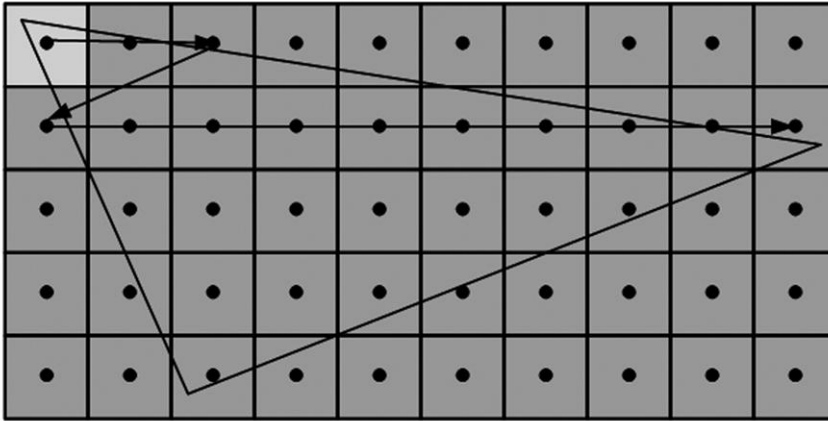
図6



【 図 8 】

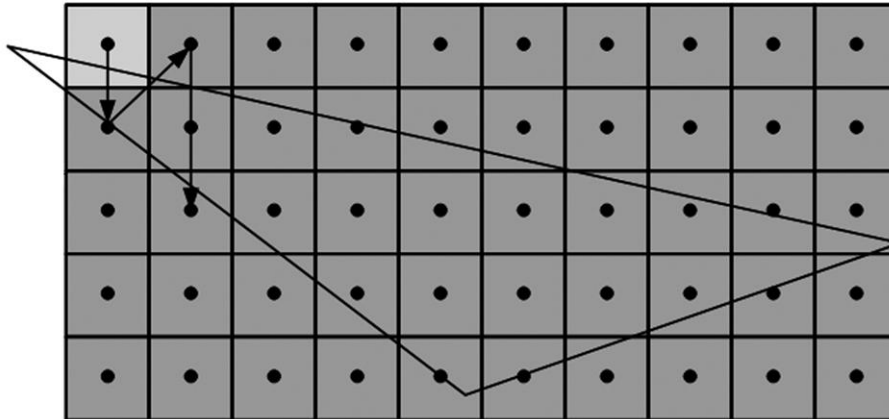
図8

InRightTriangle==No,トラバース方向=水平



(a)

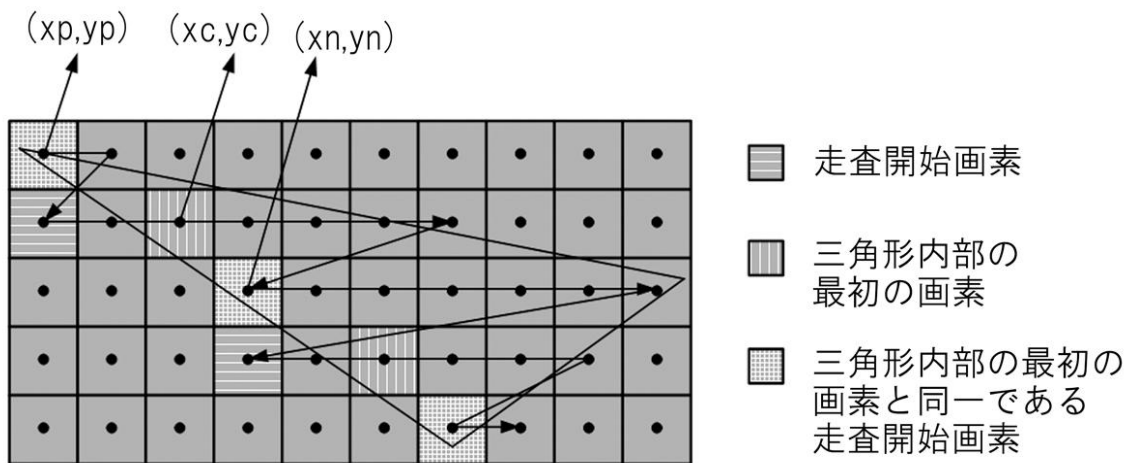
InRightTriangle==Yes,トラバース方向=垂直



(b)

【図 9】

図9



トラバース開始画素=左上 &&
 トラバース方向=水平

フロントページの続き

(74)代理人 100114177

弁理士 小林 龍

(72)発明者 バイ シャンフイ

中華人民共和国, シャンハイ, ヤン アン ロード(イー) 222, バンド センター, 31 /
フロアー

(72)発明者 洲鎌 康

神奈川県川崎市中原区上小田中4丁目1番1号

(72)発明者 タン チミン

中華人民共和国, シャンハイ, ヤン アン ロード(イー) 222, バンド センター, 31 /
フロアー

Fターム(参考) 5B080 AA14 GA25