

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号
特許第4928325号
(P4928325)

(45) 発行日 平成24年5月9日 (2012.5.9)

(24) 登録日 平成24年2月17日 (2012.2.17)

(51) Int. Cl.

F I

G O 6 T 3 / 0 0 (2 0 0 6 . 0 1)

G O 6 T 5 / 3 0 (2 0 0 6 . 0 1)

G O 6 T 7 / 6 0 (2 0 0 6 . 0 1)

G O 6 T 9 / 2 0 (2 0 0 6 . 0 1)

G O 6 T 3 / 0 0 5 0 0 A

G O 6 T 5 / 3 0 A

G O 6 T 7 / 6 0 2 5 0 A

G O 6 T 9 / 2 0

請求項の数 21 (全 31 頁)

(21) 出願番号	特願2007-85894 (P2007-85894)	(73) 特許権者	000001007
(22) 出願日	平成19年3月28日 (2007.3.28)		キヤノン株式会社
(65) 公開番号	特開2007-293829 (P2007-293829A)		東京都大田区下丸子3丁目30番2号
(43) 公開日	平成19年11月8日 (2007.11.8)	(74) 代理人	100076428
審査請求日	平成22年3月26日 (2010.3.26)		弁理士 大塚 康德
(31) 優先権主張番号	特願2006-100377 (P2006-100377)	(74) 代理人	100112508
(32) 優先日	平成18年3月31日 (2006.3.31)		弁理士 高柳 司郎
(33) 優先権主張国	日本国 (JP)	(74) 代理人	100115071
(31) 優先権主張番号	特願2006-100380 (P2006-100380)		弁理士 大塚 康弘
(32) 優先日	平成18年3月31日 (2006.3.31)	(74) 代理人	100116894
(33) 優先権主張国	日本国 (JP)		弁理士 木村 秀二
(31) 優先権主張番号	特願2006-100381 (P2006-100381)	(72) 発明者	石田 良弘
(32) 優先日	平成18年3月31日 (2006.3.31)		東京都大田区下丸子3丁目30番2号 キ
(33) 優先権主張国	日本国 (JP)		ヤノン株式会社内

最終頁に続く

(54) 【発明の名称】 画像処理方法、画像処理装置、プログラムおよび記憶媒体

(57) 【特許請求の範囲】

【請求項 1】

抽出手段が、細線化された線図形の端点と交点とに基づいて、前記細線化された線図形の端点と端点の間、または端点と交点の間、または交点と交点の間をつなぐ線素または閉曲線ごとに前記細線化された線図形を分割し、当該分割された線素および閉曲線それぞれの輪郭情報を抽出する抽出工程と、

線芯化手段が、前記抽出工程で抽出された輪郭情報の位置を半画素単位に調整することにより、線芯化されたベクトルデータを生成する線芯化工程と、

を有することを特徴とする画像処理方法。

【請求項 2】

抽出手段が、細線化された線画像を含む画像を、画素マトリクス単位でラスタ走査し、前記画素マトリクス内の注目画素の値と前記注目画素の周囲の画素値とを、予め定めたパターンのいずれに該当するか判定し、当該判定されたパターンに応じて定められているベクトル列に基づいて、前記細線化された線図形の端点と端点の間、または端点と交点の間、または交点と交点の間をつなぐ線素または閉曲線それぞれの輪郭情報を抽出する抽出工程と、

線芯化手段が、前記抽出工程で抽出された輪郭情報に基づいて、線芯化されたベクトルデータを生成する線芯化工程と、

を有することを特徴とする画像処理方法。

【請求項 3】

抽出手段が、細線化された線図形の端点と交点とに基づいて、前記細線化された線図形の端点と端点の間、または端点と交点の間、または交点と交点の間をつなぐ線素または閉曲線ごとに前記細線化された線図形を分割し、当該分割された線素および閉曲線それぞれの輪郭情報を抽出する抽出工程と、

線芯化手段が、前記抽出工程で抽出された輪郭情報に基づいて、線芯化されたベクトルデータを生成する線芯化工程と、

補助ベクトル付きデータ生成手段が、前記線図形における端点または交点に関する情報に基づいて、前記線芯化されたベクトルデータの始端点と終端点とを判別し、当該判別された終端点と始端点との間に挿入する補助ベクトルを生成し、当該生成された補助ベクトルが付加された補助ベクトル付きベクトルデータを生成する補助ベクトル付きデータ生成工程と、

を有することを特徴とする画像処理方法。

【請求項 4】

細線化手段が、線図形を含む 2 値画像に対して細線化処理を行うことにより、細線化された線図形を得る細線化工程と、

抽出手段が、前記細線化工程で得た前記細線化された線図形を、当該細線化された線図形の端点と端点の間、または端点と交点の間、または交点と交点の間をつなぐ線素または閉曲線ごとに分割し、当該分割して得られた前記細線化された線図形の線素および閉曲線それぞれに基づいて、前記細線化された線図形の線素および閉曲線それぞれを囲む輪郭情報を抽出する抽出工程と、

線芯化手段が、前記抽出工程で抽出された輪郭情報に基づいて、線芯化されたベクトルデータを生成する線芯化工程と、

を有することを特徴とする画像処理方法。

【請求項 5】

細線化された線図形の端点と交点とに基づいて、前記細線化された線図形の端点と端点の間、または端点と交点の間、または交点と交点の間をつなぐ線素または閉曲線ごとに前記細線化された線図形を分割し、当該分割された線素および閉曲線それぞれの輪郭情報を抽出する抽出手段と、

前記抽出手段で抽出された輪郭情報の位置を半画素単位に調整することにより、線芯化されたベクトルデータを生成する線芯化手段と、

を備えることを特徴とする画像処理装置。

【請求項 6】

前記抽出手段で抽出される輪郭情報には、前記線素または前記閉曲線ごとの輪郭ベクトルと、前記線図形における端点または交点に対応する前記輪郭ベクトルを示す輪郭点情報とを含むことを特徴とする請求項 5 に記載の画像処理装置。

【請求項 7】

前記輪郭点情報には、前記輪郭ベクトルを構成する輪郭点が終点となる前記輪郭ベクトルの向き、及び輪郭点が始点となる前記輪郭ベクトルの向きに関する情報が含まれることを特徴とする請求項 6 に記載の画像処理装置。

【請求項 8】

前記抽出手段は、前記細線化された線画像を含む画像を、画素マトリクス単位でラスタ走査し、前記画素マトリクス内の注目画素の値と前記注目画素の周囲の画素値とを、予め定めたパターンのいずれに該当するか判定し、当該判定されたパターンに応じて定められているベクトル列に基づいて、前記細線化された線図形の端点と端点の間、または端点と交点の間、または交点と交点の間をつなぐ線素または閉曲線それぞれの輪郭情報を抽出することを特徴とする請求項 5 乃至 7 のいずれか 1 項に記載の画像処理装置。

【請求項 9】

前記画素マトリクスのサイズは、3 画素 × 3 画素であることを特徴とする請求項 8 に記載の画像処理装置。

【請求項 10】

前記線図形における端点または交点に関する情報に基づいて、前記線芯化されたベクトルデータの始端点と終端点とを判別し、当該判別された終端点と始端点との間に挿入する補助ベクトルを生成し、当該生成された補助ベクトルが付加された補助ベクトル付きベクトルデータを生成する補助ベクトル付きデータ生成手段を、更に有することを特徴とする請求項 5 乃至 9 のいずれか 1 項に記載の画像処理装置。

【請求項 1 1】

前記補助ベクトル付きデータ生成手段は、前記線図形における端点または交点に基づいて、前記線芯化されたベクトルデータに始端点と終端点とがないと判別した場合、閉ループであることを示す情報を当該ベクトルデータに付加することを特徴とする請求項 1 0 に記載の画像処理装置。

10

【請求項 1 2】

前記補助ベクトル付きベクトルデータに対して、平滑化処理を実行する平滑化手段と、前記平滑化処理により平滑化されたベクトルデータと、前記始端点と前記終端点とに基づいて、平滑化された始端点から終端点へ向かう 1 方向のベクトルデータを生成する非周回ベクトルデータ生成手段とを更に有することを特徴とする請求項 1 0 に記載の画像処理装置。

【請求項 1 3】

2 値画像に対して細線化処理を行うことにより、前記細線化された線図形を得る細線化手段を更に備え、

前記抽出手段では、前記細線化手段で得た前記細線化された線図形の端点と交点とに基づいて、前記細線化された線図形の端点と端点の間、または端点と交点の間、または交点と交点の間をつなぐ線素または閉曲線ごとに前記細線化された線図形を分割し、当該分割された線素および閉曲線それぞれの輪郭情報を抽出することを特徴とする請求項 5 乃至 1 2 のいずれか 1 項に記載の画像処理装置。

20

【請求項 1 4】

細線化された線画像を含む画像を、画素マトリクス単位でラスタ走査し、前記画素マトリクス内の注目画素の値と前記注目画素の周囲の画素値とを、予め定めたパターンのいずれに該当するか判定し、当該判定されたパターンに応じて定められているベクトル列に基づいて、前記細線化された線図形の端点と端点の間、または端点と交点の間、または交点と交点の間をつなぐ線素または閉曲線それぞれの輪郭情報を抽出する抽出手段と、

30

前記抽出手段で抽出された輪郭情報に基づいて、線芯化されたベクトルデータを生成する線芯化手段と、

を備えることを特徴とする画像処理装置。

【請求項 1 5】

細線化された線図形の端点と交点とに基づいて、前記細線化された線図形の端点と端点の間、または端点と交点の間、または交点と交点の間をつなぐ線素または閉曲線ごとに前記細線化された線図形を分割し、当該分割された線素および閉曲線それぞれの輪郭情報を抽出する抽出手段と、

前記抽出手段で抽出された輪郭情報に基づいて、線芯化されたベクトルデータを生成する線芯化手段と、

40

前記線図形における端点または交点に関する情報に基づいて、前記線芯化されたベクトルデータの始端点と終端点とを判別し、当該判別された終端点と始端点との間に挿入する補助ベクトルを生成し、当該生成された補助ベクトルが付加された補助ベクトル付きベクトルデータを生成する補助ベクトル付きデータ生成手段と、

を備えることを特徴とする画像処理装置。

【請求項 1 6】

線図形を含む 2 値画像に対して細線化処理を行うことにより、細線化された線図形を得る細線化手段と、

前記細線化手段で得た前記細線化された線図形を、当該細線化された線図形の端点と端点の間、または端点と交点の間、または交点と交点の間をつなぐ線素または閉曲線ごとに

50

分割し、当該分割して得られた前記細線化された線図形の線素および閉曲線それぞれに基づいて、前記細線化された線図形の線素および閉曲線それぞれを囲む輪郭情報を抽出する抽出手段と、

前記抽出手段で抽出された輪郭情報に基づいて、線芯化されたベクトルデータを生成する線芯化手段と、

を備えることを特徴とする画像処理装置。

【請求項 17】

コンピュータを、

細線化された線図形の端点と交点とに基づいて、前記細線化された線図形の端点と端点の間、または端点と交点の間、または交点と交点の間をつなぐ線素または閉曲線ごとに前記細線化された線図形を分割し、当該分割された線素および閉曲線それぞれの輪郭情報を抽出する抽出手段、

前記抽出手段で抽出された輪郭情報の位置を半画素単位に調整することにより、線芯化されたベクトルデータを生成する線芯化手段、として機能させるためのプログラム。

【請求項 18】

コンピュータを、

細線化された線画像を含む画像を、画素マトリクス単位でラスタ走査し、前記画素マトリクス内の注目画素の値と前記注目画素の周囲の画素値とを、予め定めたパターンのいずれに該当するか判定し、当該判定されたパターンに応じて定められているベクトル列に基づいて、前記細線化された線図形の端点と端点の間、または端点と交点の間、または交点と交点の間をつなぐ線素または閉曲線それぞれの輪郭情報を抽出する抽出手段、

前記抽出手段で抽出された輪郭情報に基づいて、線芯化されたベクトルデータを生成する線芯化手段、

として機能させるためのプログラム。

【請求項 19】

コンピュータを、

細線化された線図形の端点と交点とに基づいて、前記細線化された線図形の端点と端点の間、または端点と交点の間、または交点と交点の間をつなぐ線素または閉曲線ごとに前記細線化された線図形を分割し、当該分割された線素および閉曲線それぞれの輪郭情報を抽出する抽出手段、

前記抽出手段で抽出された輪郭情報に基づいて、線芯化されたベクトルデータを生成する線芯化手段、

前記線図形における端点または交点に関する情報に基づいて、前記線芯化されたベクトルデータの始端点と終端点とを判別し、当該判別された終端点と始端点との間に挿入する補助ベクトルを生成し、当該生成された補助ベクトルが付加された補助ベクトル付きベクトルデータを生成する補助ベクトル付きデータ生成手段、

として機能させるためのプログラム。

【請求項 20】

コンピュータを、

線図形を含む 2 値画像に対して細線化処理を行うことにより、細線化された線図形を得る細線化手段、

前記細線化手段で得た前記細線化された線図形を、当該細線化された線図形の端点と端点の間、または端点と交点の間、または交点と交点の間をつなぐ線素または閉曲線ごとに分割し、当該分割して得られた前記細線化された線図形の線素および閉曲線それぞれに基づいて、前記細線化された線図形の線素および閉曲線それぞれを囲む輪郭情報を抽出する抽出手段、

前記抽出手段で抽出された輪郭情報に基づいて、線芯化されたベクトルデータを生成する線芯化手段、

として機能させるためのプログラム。

【請求項 21】

請求項 17 乃至 20 のいずれか 1 項に記載のプログラムを格納したコンピュータ可読の記憶媒体。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、図面や文書画像内などの線図形の画像処理技術に関するものである。

【背景技術】

【0002】

近年、ペーパーレス化への要求が高まり、既存の画像データを電子化して再利用することが頻繁に行われるようになってきた。従来、画像データの再利用は、画像データを2値化処理した後にベクトル化技術を用いてベクトルデータに変換し、このベクトルデータをCADソフト等で使用するという形で行われてきている。

10

【0003】

2値画像のベクトル化に関しては、出願人は既に特許文献1を提案している。この特許文献1では、画像データにおける注目画素と、その近傍画素の状態を保持し、この注目画素をラスト走査順に取り出し、その注目画素とその近傍画素の状態に基づいて、水平方向及び垂直方向の画素間ベクトルを検出している。そして、「これら画素間ベクトル同士の接続状態を判別して、この判別された画素間ベクトルの接続状態をもとに、画像データの輪郭を抽出するように動作する。」という手法を提案している。

【0004】

20

この特許文献1による手法は、画像の中の全ての輪郭線を1回のラスト走査順だけで抽出でき、かつ、全画像データを記憶するための画像メモリを必要としないため、メモリの容量を少なくできる効果を有している。

【0005】

更に、出願人は特許文献2において、特許文献1に開示されるベクトル抽出ルールのもジュール化により効率的な輪郭点抽出が可能であることを開示している。

【0006】

また、一方で、出願人は特許文献3において、2値画像の輪郭情報を用いて高画質な変倍画像を得る画像処理装置を紹介している。特許文献3は、2値画像からアウトラインベクトルを抽出し、抽出したアウトラインベクトル表現の状態です望の倍率(任意)で滑らかに変倍されたアウトラインベクトルを作成し、滑らかに変倍されたアウトラインベクトルから2値画像を再生成することによって、任意の倍率で変倍された高画質のデジタル2値画像を得るものである。ここで、2値画像からアウトラインベクトルを抽出する方法の一例として、特許文献1や特許文献2で開示される方法が用いられる。

30

【0007】

更に、出願人は既に特許文献4を提案している。特許文献4においては、2値画像の輪郭情報を直線のみならず、2次や3次のベジェ曲線近似をすることで、より少ないデータ量で高画質な変倍画像を表現する輪郭情報を開関近似する処理手法を開示している。

【0008】

図2(a)~(d)は、2値画像からアウトラインベクトルを抽出し、抽出したアウトラインベクトル表現を滑らかに関数近似したアウトラインベクトルを用いて所望の倍率で変倍された高画質のデジタル2値画像を得る一連の流れでの各段階でのデータの様子を示す図である。図2(a)は、入力されるラインアート画像の一例を示している。図2(b)は、図2(a)の2値画像から、特許文献1または特許文献2による手法で抽出される輪郭ベクトルデータを輪郭画像として描画して可視化したものである。図2(c)は、同抽出された輪郭ベクトルデータを特許文献3による手法で滑らかに関数近似して得られるアウトラインベクトルを描画して可視化したものである。図2(d)は、同平滑化(関数近似)されたベクトルに囲まれた領域を中塗りして得られる所望の倍率で変倍された高画質のデジタル2値画像の例である。

40

【0009】

50

先述のように、ペーパーレス化への要求が高まりのもと、ビジネス文書においても、既存の画像データを電子化して再利用することが望まれている。原図のもつ各部分の部分的な太さの差や面積等をも、所望とされる変倍率を反映した形で（即ち、原図中での太い部分は変倍後の図中でも相対的に太く、原画中での細い部分では変倍後の図中でも相対的に細く）変倍するような再利用の形態に対しては、上述の特許文献 1 または特許文献 2 と特許文献 3、若しくは、特許文献 1 または特許文献 2 と特許文献 4 により、オリジナル文書中の 2 値図形の輪郭（アウトライン）ベクトルを抽出して、2 次元形状を任意の拡大縮小率で変倍することで、その実現が可能である。

【 0 0 1 0 】

しかしながら、例えば、ビジネス文書で既存の画像データを再利用する場合や、CAD/CA
Mシステム等で再利用する場合などに、図形を構成する 1 本の線を外部輪郭線と内部輪郭
線との 2 本の輪郭線で表したデータよりも、ある図形を構成する 1 本の線を、線幅情報（
線の太さや、線の面積に関する情報）を持たない単なる 1 本の線（直線、開曲線、閉曲線
など）として扱いたい場合がある。

【 0 0 1 1 】

また、もともと太さを意識せずに、細線のみで描かれている線画の再利用の要望も強い。
このような細線を再利用する場合、部分的な曲率や長さ等を変更したり、線の集まりの
中から一部の線のみを削除したり、別な線を付与したり等の編集操作を行ってから利用す
ることが多い。

【特許文献 1】特許第 3 0 2 6 5 9 2 号公報

【特許文献 2】特開平 5 - 1 0 8 8 2 3 号公報

【特許文献 3】特許第 3 0 4 9 6 7 2 号公報

【特許文献 4】特開 2 0 0 5 - 3 4 6 1 3 7 号公報

【発明の開示】

【発明が解決しようとする課題】

【 0 0 1 2 】

上記の特許文献 1 や特許文献 2 による方法では、ラスト走査順に記憶された 2 値画像から
画像中の全ての輪郭線を抽出できる。しかしながら、太さを意識せずに、細線のみで描
かれている線画や、ある図形を構成する線を、それぞれ、各線の端点や交点間をつなぐ独
立した線や閉曲線の集まりとして抽出することができなかった。一方、上述した従来技術
を用いて得られる図 2 (c) のようなアウトラインベクトルは、図形の線を外側の輪郭線と
内側の輪郭線とで表現している。このようなアウトラインベクトルから、端点や交点をつ
なぐ線で構成される閉図形の集まりを抽出する場合、線図形から抽出される輪郭線のうち
、外側輪郭線だけを用いると交点で線を分離することができず、1 つの塊になってしまう
。また一方、内側の輪郭線のみを用いると、図 2 (e) に示すように、部分図形間をつなぐ
線分や閉図形をはみ出す線分のような部分に対応する輪郭ベクトルが欠落してしまう場合
があり、不十分な処理となる。

【 0 0 1 3 】

本発明は、上述の従来例の問題点に鑑みなされたものであり、細線化された 2 値画像を
構成する線図形から輪郭情報を抽出し、抽出した輪郭情報に基づいて線図形を線芯化する
画像処理技術を提供する。つまり、1 本の線を、内側輪郭線と外側輪郭線の 2 本で表すの
ではなく、線の中心を通る 1 本の線で表すような画像処理技術を提供する。

【 0 0 1 4 】

当該画像処理を行う際は、図形の端点や交点を検出し、それらの端点や交点に基づいて
分離される線素あるいは閉曲線ごとに、線情報（ベクトルデータ）を生成する。

【 0 0 1 5 】

更に、これらの画像処理を行う際、画像を走査する回数が少なく済むようにする。また
、線素に分離した後、それぞれの線素に対してそのまま平滑化等の画像処理を行うと、線
素の端点（つまり、図形の端点もしくは交点）の位置がずれてしまう場合があるので、ず
れないようにする技術も提供する。

【課題を解決するための手段】

【0016】

上記課題を解決するために、本発明の画像処理装置は、

細線化された線図形の端点と交点とに基づいて、前記細線化された線図形の端点と端点の間、または端点と交点の間、または交点と交点の間をつなぐ線素または閉曲線ごとに前記細線化された線図形を分割し、当該分割された線素および閉曲線それぞれの輪郭情報を抽出する抽出手段と、

前記抽出手段で抽出された輪郭情報の位置を半画素単位に調整することにより、線芯化されたベクトルデータを生成する線芯化手段と、

を備えることを特徴とする。

10

【発明の効果】

【0020】

本発明によれば、画像内の線を、中心を通る1本の線で表すことができる。

【0021】

また、ベクトルデータとして得られるので、再利用が容易になる。

【0022】

さらに、線幅0の線として利用することが可能になる。

【発明を実施するための最良の形態】

【0023】

(第1実施形態)

20

図1は本発明の実施形態に係る画像処理装置の機能ブロック図を示す図である。同図において、細線化済2値画像獲得部200は、2値画像入力部210で得た2値画像を、2値画像細線化部220で細線化する。2値画像入力部210は、スキャナ等の画像入力部211と、スキャン画像の2値化処理を行う2値化部212とで構成される。2値画像細線化部220では、縦横の4方向のいずれかで連結した連結画素(4-連結)に基づいて、2値画像の細線化処理を行うことによって、細線化された2値画像を出力する。即ち、スキャナ等の画像入力部211で光電走査により読み取られた画像データを公知の2値化部212で閾値処理により2値化する。そして、2値画像細線化部220では、例えば、公知の細線化手法であるHilditch法(酒井幸市著"デジタル画像処理の基礎と応用"第2版, ISBN4-7898-3707-6, CQ出版社, 2004年2月1日発行, P.51-P.54, 等)を用いて、当該得られ

30

た2値画像の細線化処理を行う。

【0024】

図3(a)は、2値画像細線化部220に入力される2値画像の一例を示す図である。図3(b)は、図3(a)を入力2値画像とした際の2値画像細線化部220から出力される細線化済画像を示す図である。

【0025】

次に、細線化された2値画像を、端点間毎輪郭(周回)ベクトル抽出部300で処理する。1回のラスタ走査により、処理対象の細線化済2値画像中の線図形を、それらを構成している端点や交点間をつなぐ独立した線や閉曲線毎のそれぞれに対応したベクトル列の集まり(以降、「端点間毎線素粗輪郭ベクトル」と称する)として抽出し、処理結果を出力する。つまり、細線化された2値画像内に含まれる1画素幅の線図形を、図形の端点及び交点に基づいて、線素(直線あるいは閉曲線)と閉曲線とに分割し、当該分割された線素または閉曲線ごとに、それらを囲む輪郭ベクトル列を抽出する処理が行われる。

40

【0026】

尚、出力される処理結果には、ベクトル列のそれぞれを構成する各ベクトルが、処理対象の線図形の端点(または交点)位置に対応する部分から抽出されたものか否か等を表す情報(以降、「付与情報」と称することがある)が含まれる。

【0027】

端点や交点間をつなぐ線素や閉曲線毎に得られたベクトル列(端点間毎線素粗輪郭ベクトル)の集まり(輪郭情報)は、下記のような処理を行うことにより、図形を構成する線

50

について、線幅情報を持たないベクトルデータ（もしくは線幅 0 のベクトルデータ）を生成する。このようにして生成された線幅情報の無いベクトルデータは、図形を構成する線を 1 本の線として再利用したい場合や、他のアプリケーションで線幅を変更したい場合などに適している。なお、以下では、付与情報と端点間毎線素粗輪郭ベクトルとを合わせて「端点間毎輪郭（周回）ベクトル」と称する。

【 0 0 2 8 】

端点間毎線芯（周回）ベクトル生成部 3 1 0 は、端点間毎輪郭（周回）ベクトル（端点間毎線素粗輪郭ベクトルと付与情報）が入力されると、付与情報を利用して各線素の端点位置を特定し、一画素幅（一画素単位）の線素を囲む輪郭（周回）ベクトルの各ベクトルの位置をそれぞれ半画素分ずつ予め定めた規則に基づいて決まる方向へ微調整することによって、幅 0 に線芯化された線素の周回ベクトルを表す端点間毎線芯（周回）化済ベクトルを生成する。

10

【 0 0 2 9 】

図 3 (c) は、図 3 (b) を入力 2 値画像とした際の端点間毎輪郭（周回）ベクトル抽出部 3 0 0 から出力される端点間毎輪郭（周回）ベクトルを輪郭画像として描画して可視化した結果を示す図である。なお、図 3 (c) の上方に記載されている図は、端点間毎輪郭（周回）ベクトルを描画した図の一部を拡大した図であり、図形の交点部分では、交わっている線素の端点が重なっている。図 3 (d) は、端点間毎線芯（周回）ベクトル生成部 3 1 0 において端点間毎輪郭（周回）ベクトル（図 3 (c)）が入力された場合に生成される端点間毎線芯（周回）化済ベクトルを、線画像として描画して可視化した結果を示す図である。

20

【 0 0 3 0 】

端点間毎補助ベクトル入り（周回）ベクトル生成部 3 2 0 は、端点間毎線芯（周回）ベクトル生成部 3 1 0 から入力される端点間毎線芯（周回）化済ベクトルに対して、端点に相当する部分が端点として保存される（ベジェ曲線のアンカーポイントとなる）ようにするために、補助ベクトルを挿入付加して、端点間毎補助ベクトル入り（周回）ベクトルを生成する。これは、端点間毎補助ベクトル入り（周回）ベクトル平滑部 3 3 0 により平滑化処理（関数近似処理）される際に、端点部分に相当するベクトルが、他のベクトルと統合されて端点位置が不明にならないようにするための処理である。この処理の詳細は後述する。更に、補助ベクトルを挿入された各端点間毎補助ベクトル入り（周回）ベクトルのそれぞれに対し、どのベクトルが端点部にあたるかを明示する始端点・終端点情報を生成する。なお、端点のどちらが始点になるか終点になるかは、予め定めておいた規則に基づいて決定される。

30

【 0 0 3 1 】

図 3 (e) は、端点間毎補助ベクトル入り（周回）ベクトル生成部 3 2 0 で生成された端点間毎補助ベクトル入り（周回）ベクトルを線画像として描画して可視化した結果を示す図である。

【 0 0 3 2 】

端点間毎補助ベクトル入り（周回）ベクトル平滑部 3 3 0 は、端点間毎補助ベクトル入り（周回）ベクトル生成部 3 2 0 で生成された端点間毎補助ベクトル入り（周回）ベクトルに対して、各（周回）ベクトル毎に平滑化（関数近似）処理を行い、平滑化（関数近似）された平滑化（関数近似）済端点間毎補助ベクトル入り（周回）ベクトルを生成する。

40

【 0 0 3 3 】

端点間毎平滑化済ベクトル同定（非周回化）部 3 4 0 は、平滑化（関数近似）済端点間毎補助ベクトル入り（周回）ベクトルと、端点間毎補助ベクトル入り（周回）ベクトル生成部 3 2 0 より得られる始端点・終端点情報とが入力されると、各平滑化済の（周回）ベクトル毎にそれぞれ始端点と終端点となるベクトル部分を同定して、始端点と終端点の間を結ぶ非周回型のベクトル列（始端点から終端点へ向かう 1 方向のベクトル列）を生成する。そして、平滑化済ベクトル出力部 3 5 0 を介して、当該生成された端点間毎平滑化済非周回型ベクトルデータをファイル出力もしくは通信 I / F 等を介して外部装置に出力する。

50

【 0 0 3 4 】

図 3 (f) は、図 3 (e) の端点間毎補助ベクトル入り（周回）ベクトルに対して、平滑化と非周回化とを行うことにより生成される端点間毎平滑化済（非周回）ベクトルを線画像として描画して可視化した結果を示す図である。

【 0 0 3 5 】

図 4 は、上述の機能ブロックによる処理を実現する画像処理装置の構成を例示する図である。同図において、518 は、画像処理装置のバスを表している。1 は図 1 の 2 値画像入力部 210 の画像入力部 211 を構成するスキャナである。2 は画像メモリであり、3 は画像入出力 I / O である。画像メモリ 2 は画像入出力 I / O 3 を介して、スキャナ 1 で読み取られた 2 値化前の多値画像を保持したり、2 値化された後の 2 値画像データを保持するものであり、ともに図 1 の 220 に示される 2 値画像細線化部を構成している。5 は端点間毎輪郭（周回）ベクトル抽出部であり、図 1 の端点間毎輪郭（周回）ベクトル抽出部 300 を構成している。50 は入力画像ラスタ走査部であり、画像入出力 I / O 3 を介して信号線 500 より細線化済 2 値画像データを順次入力する。513 と 514 は 3 × 3 の 9 画素で構成される走査窓の各画素の状態を入力する入力ポートである。4 は、ネットワーク等を経由して、システムの外部と通信する通信 I / F である。519 は CPU である。520 は RAM で、ワーキングメモリである。6 は ROM で、後に説明する手順に沿って CPU 519 で実行されるプログラムや予め定められるパラメータやデータ等を格納している。これらの CPU 519 やメモリ 520、ROM 6 により、図 1 の端点間毎線芯（周回）ベクトル生成部 310、端点間毎補助ベクトル入り（周回）ベクトル生成部 320、端点間毎補助ベクトル入り（周回）ベクトル平滑部 330、端点間毎平滑化済ベクトル同定（非周回化）部 340 が実現される。521 はハードディスク 522 との I / O（入出力回路）である。

【 0 0 3 6 】

以下の説明では、主として、図 5、図 6、図 10 ~ 図 25 等を用いて端点間毎輪郭（周回）ベクトル抽出部 300 での処理内容を説明する。

【 0 0 3 7 】

本実施形態では、図 6 に示すように、2 値画像における注目画素（101）と、その近傍の 8 個の画素の状態を見て処理を進めるもので、注目画素をラスタ走査し、1 画素毎にずらしながら画像全体の処理を逐次実行する。つまり、3 × 3 の画素マトリクス単位で画像をラスタスキャンしながら検査していく。つまり、注目画素の画素値と周囲の画素の画素値とに基づいて検査していくことになる。

【 0 0 3 8 】

図 6 において、× は注目画素 101 を表しており、「0」及び「2」で示された位置は、主走査方向に対し注目画素 101 と同じ位置にあり、且つ副走査方向にそれぞれ 1 ラスタ前の画素（0）及び 1 ラスタ先の画素（2）を示す。「1」及び「3」で示された位置は、注目画素 101 から主走査方向にそれぞれ 1 画素前の画素（3）及び 1 画素先の画素（1）を示している。更に、「A」及び「B」は、主走査方向に 1 画素先の位置で且つそれぞれ副走査方向に 1 ラスタ前及び 1 ラスタ先の位置にある画素を示す。また、「C」及び「D」は注目画素 101 から主走査方向に 1 画素前の位置で且つそれぞれ副走査方向に 1 ラスタ前及び 1 ラスタ先の位置にある画素を示す。

図 5 は、図 4 における端点間毎輪郭（周回）ベクトル抽出部 5 の構成例をより詳細に示す図である。同図において、図 4 と同様に 50 は入力画像ラスタ走査部であり、画像入出力 I / O（3）を介して信号線 500 より細線化済 2 値画像データを順次入力する。

【 0 0 3 9 】

501 は信号線 500 をより入力される画像データをやりとりするための入力制御（インターフェース）回路である。この信号線 500 よりラスタ走査形式で順次 2 値画像データが入力されてくる。502 はラッチで、501 より入力された画像データを、図示しない画素同期クロックに同期して 1 画素ずつ順次更新しながら保持する。次の画素同期クロックにて、ラッチ 502 は次の画素データを入力制御回路 501 より入力する。この時、既に保持していた画素

データは、その画素クロックに同期してラッチ503にラッチされて、保持される。同様にラッチ503に保持されていた画素データは、次の画素同期クロックにて、ラッチ504に保持される。

【 0 0 4 0 】

505及び506はそれぞれ 1 ラスタ分の画素データを保持するFIFO (ファーストイン・ファーストアウト・メモリ) である。FIFO505は、ラッチ504の出力を順次、画素同期クロックに同期して取り込み、1 ラスタ前のデータをラッチ507へ出力する。同様に、FIFO506も、ラッチ509の出力を取り込み、ラッチ510に 1 ラスタ前の画素データを出力する。ラッチ507、508、509及びラッチ510、511、512は共にラッチ502、503、504と全く同様に動作する。

10

【 0 0 4 1 】

このようにして、ラッチ502、503、504、507、508、509、510、511及び512に記憶された 9 個の画素は、図 6 に示した 9 (3 × 3) 画素よりなる領域の画素データを記憶していることになる。即ち、これらラッチのデータは、それぞれ図 6 の "B", "2", "C", "1", "×", "3", "A", "0", "D" に対応している。

【 0 0 4 2 】

5 1 3 と 5 1 4 は 3 × 3 の 9 画素で構成される走査窓の各画素の状態を入力する入力ポートである。513、514は共にCPU519の入力ポートで、入力ポート513は、ラッチ510、502、504、512のデータを、即ち、それぞれ図 6 における "A", "B", "C", "D" の位置のデータをCPU519に入力する。同様に、入力ポート514はラッチ511、507、503、509、508のデータ、即ち、"0", "1", "2", "3", "×" の位置のデータとして入力する。

20

【 0 0 4 3 】

515は主走査方向の画素位置を示す主走査カウンタであり、図示しない副走査同期信号によりリセットされ、画素同期信号によりカウントアップする。516は副走査方向の画素位置を示す副走査カウンタで、図示しないページ同期信号によりリセットされ、副走査同期信号によりカウントアップされる。517は入出力制御用の入出力ポートであり、入力制御回路501に対し画素データ入力の実行及び保留を指示する信号、及び入力制御回路501よりCPU519への画素データ更新を知らせる信号等を保持する。521はハードディスク522の入出力制御装置である。入出力制御ポート517、主走査カウンタ515、副走査カウンタ516、入力ポート513、514、メモリ520、ディスク1/0521はバス518を介してCPU519に接続されている。

30

【 0 0 4 4 】

こうして、CPU519は入出力制御ポート517を介して、画素データの更新を行い、主走査カウンタ515及び副走査カウンタ516を介して、注目画素の画素位置 (i , j) を知ることができる。また、入力ポート513及び514を介して、注目画素及びその近傍の 8 方向の画素の状態を知ることができる。

【 0 0 4 5 】

注目画素の処理が終了すると、CPU519は入出力制御ポート517を介して 9 個のラッチに記憶される画素データの更新を指示し、同時に画素データの更新完了の信号をリセットする。入力制御回路501は、この更新の指示を受けると、画素データの更新指示の信号をクリアするとともに、後段のラッチにラッチされる画素データを更新し、この更新が終了すると入出力制御ポート517に、更新完了の信号を出力する。

40

【 0 0 4 6 】

CPU519は、更新指示の出力後、入出力制御ポート517より更新完了の信号が入力されるのを監視している。この更新完了の信号が入力されると、新たに 9 個のラッチに記憶された画素データに関する処理を実行し、以下同様の処理を繰り返す。また、入力制御回路501は、画像領域の最終画素を注目画素として処理し終えた際に、入出力制御ポート517に終了信号を出力する。

【 0 0 4 7 】

次に注目画素及びその近傍の 8 画素の状態に応じた、それぞれの場合の処理を説明する





50

。注目画素が白画素である場合は、その処理を終了してラスト走査を1画素分進め、注目画素位置を更新する。注目画素が黒画素の場合は、近傍の画素の状態によって以下の処理を行なう。



【0048】

図10～図25は、近傍画素の状態に応じて抽出される輪郭ベクトルを示す図である。図10～図25において、内部が斜めのチェッカーパターンで埋められた大き目の印は黒画素を示し、白画素は大き目の点線の印で表記してある。「d」は"do no care"、即ち、「d」の表示のある位置の画素は白画素でも黒画素でも構わないことを示している。

【0049】

また、図10～図25の各画素マトリクスにおいて、黒丸「」印、白丸「」印は、横方向ベクトルの始点、縦方向ベクトルの終点を示している。また、黒三角「」印、白三角「」印は縦方向ベクトルの始点、横方向ベクトルの終点を示している。これらのベクトルの始点・終点間の実線矢印は、始点及び終点共に定まった矢印の向く方向の輪郭ベクトルを表している。一方、点線矢印は、始点もしくは終点のいずれか一方のみが定まった、矢印の向く方向の輪郭ベクトルを表している。これら輪郭ベクトルの始点及び終点を総称して、以降、「輪郭点」と呼ぶことにする。

【0050】

また、黒丸「」印と黒三角「」印は端点部から抽出されるベクトルの始点・終点で、それぞれ端点部から抽出される横方向ベクトルの始点及び縦方向ベクトルの終点、端点部から抽出される縦方向ベクトルの始点及び横方向ベクトルの終点であることを示している。

【0051】

図10～図25のそれぞれの輪郭点には、対応付けられる付与情報として、図9の一覧表に示した輪郭点情報の何れかが付記されているものとする。この輪郭点情報が、各輪郭点情報に付記されて抽出される付与情報となる。

【0052】

図9のNo.01～16に示される16種の輪郭点情報は、「xxx_yyy_zzz」または「xxx_yyy」なる形で表記されており、いずれも、「xxx」の部分は、対象の輪郭点が始点となるベクトルの向きを表し、「yyy」の部分は、同輪郭点が始点となるベクトルの向きを表している。上向きをUP、下向きをDOWN、右向きをRIGHT、左向きをLEFTと表現する。例えば、No.01～03はいずれも、上向きベクトルの終点かつ右向きベクトルの始点になっている輪郭点であることを意味している。

【0053】

尚、「_zzz」の部分は、この付与情報をもつ輪郭点、端点部から抽出されたものであることを意味していて、図9の端点情報の欄に記載してある。ここで、「_TL」は、線素の上端部の左側(Top Left)から抽出された端点であることを示し(図7(a)参照)、「_LT」は、線素の左端部の上側(Left Top)から抽出された端点であることを示している(図7(b)参照)。「_BR」は、線素の下端部の右側(Bottom Right)から抽出された端点であることを示し(図7(d)参照)、「_RB」は、線素の右端部の下側(Right Bottom)から抽出された端点であることを示している(図7(c)参照)。また、「_LB」は、線素の左端部の下側(Left Bottom)から抽出された端点であることを示し(図7(b)参照)、「_BL」は、線素の下端部の左側(Bottom Left)から抽出された端点であることを示している(図7(d)参照)。「_RT」は、線素の右端部の上側(Right Top)から抽出された端点であることを示し(図7(c)参照)、「_TR」は、線素の上端部の右側(Top Right)から抽出された端点であることを示している(図7(a)参照)。線素の端点ではない部分(以降、「非端点部」と称する)から抽出される輪郭点には、この「_zzz」部のない輪郭点情報が付与され、これらの例を図8(a)～(d)に示す。但し、注目画素が黒画素の場合で、近傍の画素の状態によって、図10～図25に各状態における輪郭点とその付与情報を抽出する際の実際の処理においては、上述の図9の輪郭点情報は図9の「値」の欄に対応する値をもって、抽出されるように構成するものとする。

10

20

30

40

50

【 0 0 5 4 】

図 8 において、黒画素は内部が斜めのチェッカーパターンの丸印で示され、白画素は点線の白丸印で表記されている。これら輪郭ベクトルの始点・終点位置は、主走査方向及び副走査方向共に、画素と画素の中間の位置にあるものとする。また、主走査方向及び副走査方向共に、画素のある位置は正整数で示され、画素位置を 2 次元の座標で表現する。但し、ここでは小数の表現を避けるために便宜上、以降の画素位置を偶数のみで表現することにし、始点、終点の位置を奇数の整数で表現することにする。

【 0 0 5 5 】

即ち、主走査方向 m 画素 \times 副走査方向 n 画素の画像は、 $2m \times 2n$ の正の偶数（整数）の座標表現で表すものとする。このように、主走査方向を x 座標、副走査方向を y 座標とした 2 次元座標であらわすとき、これ以降 2 値画像は、それぞれが m 画素よりなる n ラスタで構成される $m \times n$ 画素（ m, n は正の整数）でなるものとし、第 j 番目のラスタにおける第 i 番目の画素位置を $(2i, 2j)$ （ i, j は正の整数で、 $i \leq m, j \leq n$ ）で表現するものとする。また、主走査方向（ x 座標）は左から右に向かう方向が正の方向であり、副走査方向（ y 座標）は上から下に向かう向きが正の方向とする。

【 0 0 5 6 】

以下、端点間毎輪郭（周回）ベクトル抽出部 3 0 0 の一連の動作を図 2 6 ~ 図 2 8 のフローチャートに従って説明する。

【 0 0 5 7 】

本実施形態の処理は、4 方向の連結画素に基づいて細線化された 2 値画像において、端点と交点を検出しつつ、端点や交点をつなぐ互いに独立した線素や閉曲線の集まりとして、それぞれの線成分に対応する独立したベクトル群を抽出し、加えて、各輪郭点に輪郭点情報をも付与して抽出できる。したがって、その機能は特許文献 1 と大きく異なる。

【 0 0 5 8 】

また、図 1 0 ~ 図 2 5 に示したように使用する画素マトリクス（3 画素 \times 3 画素）の画素パターン群も特許文献 1 と異なる。しかしながら、3 画素 \times 3 画素の 9 画素で構成される走査窓を用いて、画像をラスタ走査し、逐次 3 画素 \times 3 画素の画素パターンとして注目画素とその近傍の 8 画素の状態を判断しながら、輪郭点を抽出していく点で、各 3 画素 \times 3 画素の画素パターンに応じた輪郭点と輪郭点情報の抽出する処理以外の処理は、基本的には特許文献 1 の開示する抽出動作と同様に構成することができる。

【 0 0 5 9 】

図 2 6 は、本実施形態の画像処理装置の CPU519 による輪郭抽出処理の全体の流れを示すフローチャートである。まずステップ S 1 において、2 値画像データからベクトル列を各輪郭点に関する輪郭点情報とともに抽出する。そして、各ベクトルの始点の座標及びこのベクトルに流入してくる（このベクトルの始点の座標が終点となっている）ベクトル、流出してゆく（このベクトルの終点の座標が始点となっている）ベクトルを出力する。

【 0 0 6 0 】

ステップ S 2 において、流入及び流出ベクトルより、図 2 9 に示すような画像中の総輪郭線数（線素の総数）、輪郭線毎の輪郭点の総点数、輪郭線中の各点の x 座標、 y 座標と各点の輪郭点情報の組み合わせを記憶したテーブルを作成する。

【 0 0 6 1 】

次に、ステップ S 3 に進み、ディスク I/O521 を介して、このテーブル情報ファイル形式でハードディスク 522 に記憶して、一連の動作を終了する。

【 0 0 6 2 】

図 2 7 は図 2 6 のステップ S 1 のベクトル列抽出処理を示すフローチャートである。まずステップ S 1 1 において、入力ポート 514 のビット 4（注目画素 "X"）を見ることにより、注目画素が白画素か黒画素かを判定する。白画素の場合はステップ S13 へ進み、黒画素の場合はステップ S12 へ進む。ステップ S12 では、注目画素の周囲 8 画素の状態を見て、その状態に応じた適当な処理ルーチンをコールし、輪郭ベクトルを抽出する。

【 0 0 6 3 】

ステップS13では、前述の如く入出力制御ポート517を介して画素位置の更新を指示する。そして、入出力制御ポート517より更新完了の信号を入力するとステップS14に進み、入出力制御ポート517を介して、最終画素の処理が終了したか否かを判断する。終了していなければステップS11へ戻り、次の注目画素も同様に処理を行う。一方、ステップS14で終了していると判断すれば、元のルーチンにリターンする。

【0064】

図28は図27のステップS12に示された周囲画素の状態により実行される処理を示すフローチャートである。

【0065】

ステップS21において、CPU519のレジスタを「0」にクリアする。次に、ステップS22に進み、図6の「0」で示された位置の画素の状態（以下、「 $f(0)$ 」で表現する）が黒画素（以降、「1」で表す）であれば、ステップS23へ進み、白画素（以降、「0」で表す）であればステップS24へ進む。

【0066】

ステップS23ではレジスタの内容に1を加える。

【0067】

次に、ステップS24に進み、図6の「1」の位置の画素の状態が黒画素であれば（ $f(1) = 1$ ）、ステップS25へ進み、レジスタの内容に2を加える。 $f(1) = 0$ であればステップS26に進み、同様に図6の「2」の位置の画素が黒画素かどうかを判断する。 $f(2) = 1$ （黒画素であれば）であればステップS27へ進み、レジスタの内容に4を加える。

次にステップS28に進み、図6の「3」の位置の画素に注目し、 $f(3) = 1$ ならステップS29へ進み、レジスタの内容に8を加えるが、そうでなければステップS30へ進む。ステップS30では、レジスタの保持する値の処理番号のルーチンをコールする。

【0068】

これにより、レジスタの内容は、図6に示した「0」、「1」、「2」、「3」の画素位置の各画素の状態に応じて、0～15の値を取り得る。これらは、その値に応じてそれぞれ、図10の（ケース（case）0）、図11の（ケース1）、図12（ケース2）、図13（ケース3）、図14（ケース4）、図15（ケース5）、図16（ケース6）、図17（ケース7）、図18（ケース8）、図19（ケース9）、図20（ケース10）、図21（ケース11）、図22（ケース12）、図23（ケース13）、図24（ケース14）、図25（ケース15）に示されている。

【0069】

図28のステップS30において、図10～図25に示される様に定義された輪郭点とその輪郭点情報とをそれぞれのケースに応じて抽出する。端点間毎輪郭（周回）ベクトル抽出部300は、1回のラスタ走査のみにより、4方向連結画素に基づいて細線化された2値画像（線図形）から、それらを構成している端点や交点間をつなぐ独立した線や閉曲線毎のそれぞれに対応したベクトル列の集まり（「端点間毎線素粗輪郭ベクトル」と、その輪郭点情報とを抽出し、その処理結果を出力する。これらは、例えば、図29に示したような形式として出力される。同図における第一輪郭や第二輪郭、第a輪郭といったまとまりとして各線素から抽出されたベクトル列を表し、それらの輪郭点ごとに輪郭点情報が付与された形式となっている。

【0070】

図31に線図形の入力例と、これから抽出される端点間毎輪郭（周回）ベクトル、即ち、端点間毎線素粗輪郭ベクトル（細線化された線素を周回するベクトル列）と、輪郭点情報を示す。

【0071】

次に、図30に示すフローチャートにより、端点間毎線芯（周回）ベクトル生成部310の動作を説明する。ここでは、輪郭情報を半画素単位に調整して線芯化することになる。同図において、処理を開始すると、ステップS100において、端点間毎輪郭（周回）ベクトル抽出部300より得られた、端点間毎輪郭（周回）ベクトルを入力する。なお、

10

20

30

40

50

端点間毎輪郭輪郭（周回）ベクトルは、線図形の端点（交点）位置に対応する部分から抽出されたものか否か等を表す情報（付与情報）と、端点や交点間をつなぐ線素（独立した線や閉曲線）毎のそれぞれに対応したベクトル列の集まり（端点間毎線素粗輪郭ベクトル）とで構成されている。

【 0 0 7 2 】

次に、ステップ S 1 1 0 では、ステップ S 1 0 0 で入力した図 2 9 に示したような形式の端点間毎輪郭（周回）ベクトルから、中に含まれる 1 つのベクトル列（図 2 9 の第 k 輪郭（1 k a））を当面の処理対象としてステップ S 1 1 0 が実行されるたびに順次昇順に定める。

【 0 0 7 3 】

ステップ S 1 2 0 では、S 1 1 0 で定めたベクトル列（総点数 p とする）の中の 1 つのベクトルの輪郭点（第 q 点（1 q p））を当面の処理対象としてステップ S 1 2 0 が実行されるたびに順次昇順に定める。

【 0 0 7 4 】

ステップ S 1 3 0 では、ステップ S 1 2 0 で定めた輪郭点の輪郭点情報を参照して、この輪郭点を終点とするベクトルか、若しくは、この輪郭点を始点とするベクトルのいずれかが左向き（L E F T）か否かを判定し、左向きの場合は、ステップ S 1 4 0 に進み、そうではない場合にはステップ S 1 5 0 に進む。

【 0 0 7 5 】

ステップ S 1 4 0 では輪郭点（座標値を $(2i, 2j)$ とする）の y 座標値を 1 減じて（副走査方向に半画素分原点側にずらして）、ステップ S 1 6 0 に進む。

【 0 0 7 6 】

ステップ S 1 5 0 では、輪郭点の y 座標値を 1 増やして（副走査方向に半画素分原点とは反対側にずらして）、ステップ S 1 6 0 に進む。

【 0 0 7 7 】

ステップ S 1 6 0 では、輪郭点の輪郭点情報を参照して、この輪郭点を終点とするベクトルか、もしくは、この輪郭点を始点とするベクトルのいずれかが上向きか否かを判定し、上向き（U P）の場合は、ステップ S 1 7 0 に進み、そうではない場合にはステップ S 1 8 0 に進む。

【 0 0 7 8 】

ステップ S 1 7 0 では輪郭点の x 座標値を 1 増やして（主走査方向に半画素分原点とは反対側にずらして）、ステップ S 1 9 0 に進む。

【 0 0 7 9 】

ステップ S 1 8 0 では、輪郭点の x 座標値を 1 減じて（主走査方向に半画素分原点側にずらして）、ステップ S 1 9 0 に進む。

【 0 0 8 0 】

ステップ S 1 9 0 では、ステップ S 1 1 0 で定めたベクトル列内の全ての輪郭点の吟味が終わったか否かを判定し、そうであればステップ S 2 0 0 へ進み、そうでなければ、同ベクトル列内の次の輪郭点にステップ S 1 3 0 からステップ S 1 9 0 までの処理を施すべくステップ S 1 2 0 に戻る。

【 0 0 8 1 】

ステップ S 2 0 0 では、ステップ S 1 0 0 で入力した端点間毎線素粗輪郭ベクトル中の全てのベクトル列の処理が終了したか否かを判定し、そうであればステップ S 2 1 0 へ進み、そうでなければ、先のベクトル列に対してステップ S 1 2 0 からステップ S 2 0 0 までの処理を施すべくステップ S 1 1 0 へ戻る。

【 0 0 8 2 】

ステップ S 2 1 0 では、ステップ S 1 0 0 で入力した端点間毎線素粗輪郭ベクトルに対する処理結果を出力する。

【 0 0 8 3 】

尚、ステップ S 1 3 0 やステップ S 1 6 0 での判定は、先の図 9 における「値」欄の数

10

20

30

40

50

値を輪郭点情報としているので、これらの数値を2進数で表現した際のLSBをbit0、MSBをbit7とすると、ステップ130ではbit2(図9では「右向(0)左向(1)」欄)を調べて、0なら右向き、1なら左向きと判定することができる。ステップ160ではbit3(図9では「上向(0)下向(1)」欄)を調べて、0なら上向き、1なら下向きと判定することができる。

端点間毎輪郭(周回)ベクトル抽出部300より得られた、端点間毎輪郭(周回)ベクトル(即ち、線図形の端点(交点)位置に対応する部分から抽出されたものか否か等を表す情報(付与情報)と、端点や交点間をつなぐ線素(独立した線や閉曲線)毎のそれぞれに対応したベクトル列の集まり(端点間毎線素粗輪郭ベクトル)とで構成される)を入力し、付与情報を利用して、一画素幅の幅を持った線素の輪郭(周回)ベクトルの各ベクトルの位置をそれぞれ半画素分ずつ相応する方向へ予め定める規則により微調整することによって、幅0に線芯化された端点間毎線芯(周回)化済ベクトルを生成する。なお、端点間毎線芯(周回)化済ベクトルもまた、例えば、図29に示したような形式として表現でき、本実施形態では、この形式で出力されるものとする。

【0084】

図31に、端点間毎線素粗輪郭ベクトル(1画素幅の線素の周囲のベクトル列)と、輪郭点情報から幅0に線芯化された端点間毎線芯(周回)化済ベクトル(線素上に描かれたベクトル列)の例を示す。

【0085】

次に、端点間毎補助ベクトル入り(周回)ベクトル生成部320の動作を説明する。端点間毎補助ベクトル入り(周回)ベクトル生成部320は、端点間毎補助ベクトル入り(周回)ベクトル平滑部330により平滑化(関数近似)処理される際に、端点部分に相当するベクトルが、他のベクトルと統合されて端点位置が不明にならぬように、端点部分に相当する部分が端点として保存される(ベジェ曲線のアンカーポイントとなる)ように補助ベクトルを挿入する。加えて、各平滑化済の(周回)ベクトル毎の始端点・終端点情報も生成する。この始端点・終端点情報は、次々工程である端点間毎平滑化済ベクトル同定(非周回化)部340において、平滑化(関数近似)済端点間毎補助ベクトル入り(周回)ベクトルから、各平滑化済の(周回)ベクトル毎にそれぞれの始端点と終端点となるベクトルを同定(判別)して、始端点と終端点の間の非周回型のベクトル列を生成するために用いられる。

【0086】

本実施形態では、端点間毎補助ベクトル入り(周回)ベクトル平滑部330は、補助ベクトル入りのベクトルデータを平滑化するので、端点を保持した状態で、線芯化されたベクトルの平滑化を行うことができる。

【0087】

本実施形態に係る端点間毎補助ベクトル入り(周回)ベクトル平滑部330によれば、ベクトル平滑化(関数近似)の流れは、以下のとおりである。

【0088】

まず、粗輪郭データと称して、2値のラスタ画像のデータより抽出された、水平ベクトル、垂直ベクトルが交互に並ぶ構成となる輪郭データを入力とし、粗輪郭上の線分より接線線分を抽出する。

【0089】

抽出された接線線分よりアンカーポイントを抽出し、抽出されたアンカーポイント間の線分により構成されるグループを2次もしくは3次ベジェ曲線、及び直線をあてはめ、あるいは、ベジェ曲線近似を行い、3次もしくは2次のベジェ曲線により置き換えていく。

【0090】

ここで、アンカーポイントとして定めた点をもとに、アンカーポイント間にある輪郭点を1つのグループとして関数近似していく方法は、関数近似法としての基本的な手法である。また、アンカーポイントは、その間が、2次もしくは3次ベジェ曲線、及び直線のいずれに関数近似されたとしても、アンカーポイント自体の位置(座標値)は、変化しない

10

20

30

40

50

という特徴をもつ。

【 0 0 9 1 】

本実施形態に係る端点間毎補助ベクトル入り（周回）ベクトル生成部 3 2 0 では、端点間毎線芯（周回）化済ベクトルの端点のそれぞれを、一方を始端点と他方を終端点とし、かつ、それらがともにアンカーポイントとなるように始端点と終端点間に補助ベクトル（補助輪郭点）を挿入していく処理をする。

【 0 0 9 2 】

接線線分とするベクトルの抽出の条件としては、（ 1 ）注目するベクトルの前後のベクトルの向きが互いに逆向きであるもの、（ 2 ）抽出済みの主接線線分に隣接し、ベクトルの長さ L_1 が L_1 （パラメータ）を満たすもの、（ 3 ）ベクトルの長さ L_2 が L_2 （パラメータ）を満たすもの等が条件となる。ここで、条件に使用されるパラメータ 1 ~ 2 は、解像度に依存する一定値でもよく、また、アウトラインサイズ等のオブジェクトサイズにより、適応的に変更しても良い。

【 0 0 9 3 】

アンカーポイントの抽出に関して、抽出された接線線分上に新たな点を抽出し、それをアンカーポイントとする。アンカーポイントは接線線分の端 2 つに対しそれぞれ抽出される。よって、一つの接線線分に対し 2 つのアンカーポイントが抽出されるが、2 つのアンカーポイントが一致した場合には一つのアンカーポイントのみ抽出されることになる。2 つのアンカーポイントが抽出される場合は、アンカーポイントに挟まれた部位は自動的にオブジェクト上の直線となる。接線線分上の一つの端点に対するアンカーポイント抽出方法としては、注目する接線線分であるベクトルに隣接するベクトルが接線線分であれば、この隣接する側の端点をアンカーポイントとすることができる。

【 0 0 9 4 】

そこで、端点間毎補助ベクトル入り（周回）ベクトル生成部 3 2 0 は、点間毎線芯（周回）化済ベクトルの端点のそれぞれを、一方を始端点とし、他方を終端点とし、かつ、それらがともにアンカーポイントとなるように、補助点を挿入する。

【 0 0 9 5 】

図 3 2 (a) を用いて、始端点としたい端点 P_0 の直前に挿入する補助輪郭点 P_{-1} の挿入方法を説明する。補助輪郭点 P_{-1} を始点とする補助ベクトル V_{-1} が、始端点としたい端点 P_0 の直後の輪郭点 P_1 を始点とするベクトル V_1 と逆向きになるように、補助輪郭点 P_{-1} を定める。このようにすれば、接線線分の抽出条件の（ 1 ）から、ベクトル V_0 が接線線分となる。

【 0 0 9 6 】

また、補助輪郭点 P_{-1} を始点とする補助ベクトル V_{-1} が上述の接線線分の抽出条件の（ 3 ）を満たすように十分な長さを持つようにする。このようにすれば、補助ベクトル V_{-1} もまた接線線分となる。補助ベクトル V_{-1} とベクトル V_0 とが共に接線線分となることにより、上述のアンカーポイント抽出方法における「注目する接線線分であるベクトルに隣接するベクトルが接線線分であれば、この隣接する側の端点をアンカーポイントとする」という条件から、始端点としたい端点 P_0 をアンカーポイントとすることができる。

次に、図 3 2 (b) を用いて、終端点としたい端点 P_n の直後に挿入する補助輪郭点 P_{n+1} の挿入方法を説明する。補助輪郭点 P_{n+1} を終点とする補助ベクトル V_n が、終端点としたい端点 P_n の直前の輪郭点 P_{n-1} を終点とするベクトル V_{n-1} と逆向きになるように、補助輪郭点 P_{n+1} を定める。このようにすれば、上述の抽出条件（ 1 ）から、ベクトル V_{n-1} が接線線分となる。また、補助輪郭点 P_{n+1} を終点とする補助ベクトル V_n が上述の接線線分の抽出条件（ 3 ）を満たすように十分な長さを持つようにする。このようにすれば、補助ベクトル V_n もまた接線線分となる。

【 0 0 9 7 】

補助ベクトル V_{n-1} とベクトル V_n とが共に接線線分となることにより、上述のアンカーポイント抽出方法における「注目する接線線分であるベクトルに隣接するベクトルが接線線分であれば、この隣接する側の端点をアンカーポイントとする」という条件から、終端

点としたい端点 P_n をアンカーポイントとすることができる。

【 0 0 9 8 】

このように、始端点と終端点がアンカーポイントになるように、それぞれ、始端点の直前への挿入点と、終端点の直後への挿入点を定めたのち、これらの挿入点が水平ベクトルと垂直ベクトルの連続で連結される。始端点から終端点までの本来の輪郭点列と、終端点の直後から始端点の直前までの挿入点を経て再び始端点につながる一連の輪郭点列を生成する。更に、この一連の輪郭点列の中で、どれが、始端点と終端点であるかを同定するための情報とを加えた始端点・終端点情報を生成する。本実施形態では、輪郭点列中の始端点の座標値と終端点の座標値そのものを改めて別途生成することで、始端点・終端点情報とする。

10

【 0 0 9 9 】

次に、図 3 3 に示すフローチャートにより、端点間毎補助ベクトル入り（周回）ベクトル生成部 3 2 0 の動作を説明する。

【 0 1 0 0 】

ステップ S 3 0 0 において、端点間毎線芯（周回）ベクトル生成部 3 1 0 より得られた、幅 0 に線芯化された端点間毎線芯（周回）化済ベクトルを入力する。

【 0 1 0 1 】

次に、ステップ S 3 1 0 では、ステップ S 3 0 0 で入力した図 2 9 に示す形式の端点間毎線芯（周回）化済ベクトルから、その中に含まれる 1 つのベクトル列（図 2 9 の第 k 輪郭（1 k a））を当面の処理対象とする。

20

【 0 1 0 2 】

次に、ステップ S 3 2 0 では、S 3 1 0 で定めたベクトル列（総点数 p とする）の中の 1 つのベクトルの輪郭点（第 q 点（1 q p））を処理対象として定め、ステップ S 3 3 0 で始端点か否か判断する。ステップ S 3 3 0 で始端点が見つけれられるまでの間、ステップ S 3 2 0 が実行されるたびに第 1 点から第 p 点までを昇順に順次 1 点ずつ処理対象として定める。

【 0 1 0 3 】

次に、ステップ S 3 3 0 では、ステップ S 3 2 0 で定めた輪郭点の輪郭点情報を参照して、この輪郭点が始端点とすべきベクトルか否かを判定する。この輪郭点が始端点とすべきベクトルの場合は、ステップ S 3 6 0 に進み、そうではない場合にはステップ S 3 4 0 に進む。ここで、輪郭点が始端点とすべきベクトルか否かは、その輪郭点の輪郭点情報を参照して、先に説明した図 9 で表記しているところの「UP_RIGHT_LT(03H)」、「DOWN_LEFT_RB(0FH)」、「LEFT_UP_BL(17H)」、「RIGHT_DOWN_TR(1BH)」のいずれかであるか否かにより判定する。これらは、図 9 の「値」の数値の bit1 と bit0 が共に「1」であるか否かを調べることで実現することもできる。

30

【 0 1 0 4 】

ステップ S 3 4 0 では、ステップ S 3 1 0 で入力した端点間毎線芯（周回）化済ベクトルの中に含まれる 1 つのベクトル列中の全てのベクトルの処理が終了したか否かを判定する。終了していなければ、該当するベクトル列中の次のベクトルに対してステップ S 3 3 0 の判定を施すべくステップ S 3 2 0 へ戻る。

40

【 0 1 0 5 】

一方、ステップ S 3 1 0 で入力したベクトル列中に、始端点候補が存在しなかった場合は 1 ベクトル列の処理が終了したと判定して、ステップ S 3 5 0 へ進む。

【 0 1 0 6 】

ステップ S 3 5 0 において、ベクトル列は端点のない閉ループである旨を示す閉ループマーカをベクトル列に対して付与した上で、それに加えて、ベクトル列中の一連の各ベクトルをそのまま出力する。

【 0 1 0 7 】

図 3 7 (a) は、ステップ S 3 5 0 で出力するデータ形式を例示する図である。対象としたベクトル列を第 s 番目の輪郭とすると、そのベクトル列中には v 個のベクトルが含ま

50

れている。

【0108】

尚、閉ループマーカとして、端点座標としては本来ありえない(-1,-1)の座標値を2点分の始端点・終端点情報として付与する。

【0109】

ステップS350の処理を終えると、ステップS420へ処理を進める。

【0110】

一方、ステップS360において、輪郭点を始端点として登録し、かつ、その他の輪郭点とともに出力してステップS370へ処理を進める。

【0111】

ステップS370では、ステップS310で定めたベクトル列(総点数pとする)の中から、前記登録した始端点の次の輪郭点から順に、1つのベクトルの輪郭点(第q点(1 ≤ q ≤ p))を入力する。そして、ステップS380で終端点が見つけれられるまでの間、ステップS370で最初に定められた輪郭点以降の点から第p点に向けて昇順に順次1点ずつ当面の処理対象として定め、ステップS380に処理を進める。

【0112】

次に、ステップS380では、先のステップS370で定めた輪郭点の輪郭点情報を参照して、この輪郭点が終端点とすべきベクトルか否かを判定する。この輪郭点が終端点とすべきベクトルの場合は、ステップS400に処理を進め、そうではない場合にはステップS390に処理を進める。

【0113】

ここで、輪郭点が終端点とすべきベクトルか否かは、その輪郭点の輪郭点情報を参照して、先に説明した図9で表記している「UP_RIGHT_TL(01H)」、「DOWN_LEFT_BR(0DH)」、「LEFT_UP_LB(15H)」、「RIGHT_DOWN_RT(19H)」のいずれかであるか否かをもちてその判定を行なう。これらは、図9の「値」の数値のbit1とbit0がそれぞれ「0」と「1」であるか否かを調べることで実現することもできる。

【0114】

ステップS390では、当該処理対象の輪郭点を、始端点～終端点間を構成するベクトルの1つとして出力してステップS370へ戻る。

【0115】

ステップS400では、当該処理対象の輪郭点を終端点として登録し、かつ、ステップS390で出力された輪郭点を始端点～終端点間を構成するベクトルとして出力してステップS410へ進む。

【0116】

ステップS410では、ベクトル列の始端点と終端点の両方が定まり、かつ、始端点～終端点間の一連の各ベクトルも出力された状態となっており、ここでは、終端点～始端点の間に挿入される一連の補助ベクトルを求める。ステップS410の内容は、図34に示すフローチャートと図35及び図36を用いて後に詳細に説明する。ステップS410の処理を終えるとステップS420へ処理を進める。

【0117】

ステップS420では、ステップS300で入力した端点間毎線芯(周回)化済ベクトル中の全てのベクトル列の処理が終了したか否かを判定する。全てが終了していればステップS430へ進み、終了していなければ、次のベクトル列に対してステップS310からステップS410までの処理を施すべくステップS310へ戻る。

【0118】

ステップS430では、ステップS300で入力した端点間毎線芯(周回)化済ベクトルに対する処理結果を出力する。

【0119】

図37(b)は、対象としたベクトル列に、始端点・終端点の両方が定まり、且つ、始端点～終端点間の一連の各ベクトルと、終端点～始端点の間に挿入される一連の補助ベクトル

10

20

30

40

50

ルが求まった場合に出力されるベクトル列に対する出力例を示す図である。同図において、対象としたベクトル列は第 u 番目とし、そのベクトル列中には始端点と終端点自身も含めて始端点～終端点間の一連の各ベクトルが t 個含まれている。更に、終端点～始端点の間に挿入される一連の補助ベクトルが r 個含まれている。ここで、第 1 点は、始端点そのものであり、第 t 点は、終端点そのものとなっている。

【 0 1 2 0 】

次に、図 3 4 に示すフローチャートにより、図 3 3 に示すフローチャートのステップ S 4 1 0 の処理の内容を更に説明する。ここでは、ベクトル列の始端点と終端点の両方が定まり、かつ、始端点～終端点間の一連の各ベクトルも出力された状態となっている。そして、終端点～始端点の間に挿入される一連の補助ベクトルを求めるのが本フローチャートの処理内容である。本処理は、図 3 3 と同様に、端点間毎補助ベクトル入り（周回）ベクトル生成部 3 2 0 により実行される。

10

【 0 1 2 1 】

ステップ S 4 1 1 では、先に図 3 2 (a) を用いて説明したように、始端点としたい端点 P_0 の直前に挿入する補助輪郭点 P_{-1} の座標値を求める。即ち、補助輪郭点 P_{-1} を始点とする補助ベクトル V_{-1} が、始端点としたい端点 P_0 の直後の輪郭点 P_1 を始点とするベクトル V_1 と逆向きになるように、かつ、補助輪郭点 P_{-1} を始点とする補助ベクトル V_{-1} が上述の接線線分の抽出条件 (3) を満たすように十分な長さ（例えば、10 画素長以上）を持つように補助輪郭点 P_{-1} を定める。

20

【 0 1 2 2 】

次に、ステップ S 4 1 2 において、先に図 3 2 (b) を用いて説明したように、終端点としたい端点 P_n の直後に挿入する補助輪郭点 P_{n+1} の座標値を求める。即ち、補助輪郭点 P_{n+1} を終点とする補助ベクトル V_n が、終端点としたい端点 P_n の直前の輪郭点 P_{n-1} を終点とするベクトル V_{n-2} と逆向きになるように、かつ、補助輪郭点 P_{n+1} を終点とする補助ベクトル V_n が上述の接線線分の抽出条件 (3) を満たすように十分な長さ（例えば、10 画素長以上）を持つように補助輪郭点 P_{n+1} を定める。

【 0 1 2 3 】

ステップ S 4 1 3 では、ステップ S 4 1 1 で求めた始端点としたい端点 P_0 の直前に挿入する補助輪郭点 P_{-1} を始点とする補助ベクトル V_{-1} と、ステップ S 4 1 2 で求めた終端点としたい端点 P_n の直後に挿入する補助輪郭点 P_{n+1} を終点とする補助ベクトル V_n が、共に水平ベクトルとなっているか、あるいは、共に垂直ベクトルとなっているかを判定する。

30

【 0 1 2 4 】

共に水平ベクトル、あるいは、共に垂直ベクトルの場合は、処理をステップ S 4 1 4 に進める。一方、ステップ S 4 1 3 の判定で、共に水平ベクトル、あるいは、共に垂直ベクトルでない場合は、処理をステップ S 4 1 6 に進める。

【 0 1 2 5 】

V_{-1} と V_n が、共に水平ベクトルとなっているか、あるいは、共に垂直ベクトルとなっている場合の例を図 3 5 に示す。

【 0 1 2 6 】

ステップ S 4 1 4 では終端点としたい端点 P_n の次の次にあたる補助輪郭点 P_{n+2} を、 P_{n+1} からみて P_{-1} から遠くなる方向に、上述の接線線分の抽出条件 (3) を満たすように、 P_{n+2} を終点とするベクトル V_{n+1} が十分な長さ（例えば、10 画素長以上）を持つように定める。

40

【 0 1 2 7 】

そして、ステップ S 4 1 5 において、水平ベクトルと垂直ベクトルが交互につながるような形式で、 P_{n+2} と P_{-1} を連結する様に、終端点としたい端点 P_n の 3 点後の補助輪郭点（この点は、始端点としたい端点 P_0 の 2 点前の補助輪郭点でもある）である点 P_{n+3} を定める。このようにして補助輪郭点 P_{n+1} 、 P_{n+2} 、 P_{n+3} 、 P_{-1} が定まる。そして、補助点挿入の一連のステップを終え、図 3 3 のフローチャートの S 4 2 0 にもどる。

50

【 0 1 2 8 】

一方、ステップ S 4 1 6 は、 V_{-1} と V_n の、どちらか一方が水平ベクトルとなっていて、他方は垂直ベクトルとなっている場合の処理であり、この場合の例を図 3 6 に示す。

【 0 1 2 9 】

ステップ S 4 1 6 では終端点としたい端点 P_n の次の次にあたる補助輪郭点 P_{n+2} を、 P_{n+1} からみて P_{-1} から遠くなる方向に、接線線分の抽出条件 (3) を満たすように、 P_{n+2} を終点とするベクトル V_{n+1} が十分な長さ (例えば、10 画素長以上) を持つように定める。

【 0 1 3 0 】

ステップ S 4 1 7 では、点 P_{n+3} を、 P_{n+2} からみて P_{-1} から遠くなる方向に、上述の接線線分の抽出条件 (3) を満たすように、 P_{n+3} を終点とするベクトル V_{n+2} が十分な長さ (例えば、10 画素長以上) を持つように定める。

【 0 1 3 1 】

ステップ 4 1 8 では、水平ベクトルと垂直ベクトルが交互につながるような形式で、 P_{n+3} と P_{-1} を連結する様に、終端点としたい端点 P_n の 4 点後の補助輪郭点 (この点は始端点としたい端点 P_0 の 2 点前の補助輪郭点でもある) である点 P_{n+4} を定める。このようにして補助輪郭点 P_{n+1} , P_{n+2} , P_{n+3} , P_{n+4} , P_{-1} が定まる。そして、補助点挿入の一連のステップを終え、図 3 3 のフローチャートの S 4 2 0 にもどる。

【 0 1 3 2 】

端点間毎補助ベクトル入り (周回) ベクトル生成部 3 2 0 は、端点間毎線芯 (周回) ベクトル生成部 3 1 0 より得られる上記の端点間毎線芯 (周回) 化済ベクトルを入力する。そして、次工程の処理をする端点間毎補助ベクトル入り (周回) ベクトル平滑部 3 3 0 により平滑化 (関数近似) 処理される際に、端点部分に相当するベクトルが、他のベクトルと統合されて端点位置が不明にならないように、端点部分に相当する部分が端点として保存される (ベジエ曲線のアンカーポイントとなる) ように補助ベクトルを挿入することにより、端点間毎補助ベクトル入り (周回) ベクトルを生成する。

【 0 1 3 3 】

加えて、補助ベクトルが挿入された各端点間毎補助ベクトル入り (周回) ベクトルのそれぞれに対し、どのベクトルが端点部にあたるかを明示する始端点・終端点情報をも生成する。生成された処理結果は、例えば、図 2 9 に示したような形式の各輪郭データ部分のそれぞれを図 3 7 (a) や図 3 7 (b) の各ベクトル列データに置き換えた形として表現でき、本実施形態では、この形式で出力が可能である。

【 0 1 3 4 】

尚、端点間毎補助ベクトル入り (周回) ベクトル平滑部 3 3 0 は、各 (周回) ベクトル毎に平滑化 (関数近似) された平滑化 (関数近似) 済端点間毎補助ベクトル入り (周回) ベクトルを生成する。そしてその生成結果は、次工程の処理をする端点間毎平滑化済ベクトル同定 (非周回化) 部 3 4 0 に出力される。この際、端点間毎補助ベクトル入り (周回) ベクトル生成部 3 2 0 で生成された始端点・終端点情報も次工程の処理をする端点間毎平滑化済ベクトル同定 (非周回化) 部 3 4 0 に出力される。

【 0 1 3 5 】

端点間毎平滑化済ベクトル同定 (非周回化) 部 3 4 0 は、端点間毎補助ベクトル入り (周回) ベクトル平滑部 3 3 0 より得られる、平滑化 (関数近似) 済端点間毎補助ベクトル入り (周回) ベクトルと、端点間毎補助ベクトル入り (周回) ベクトル生成部 3 2 0 より得られる始端点・終端点情報とを入力し、各平滑化済の (周回) ベクトル列毎に、そのベクトル列内の各ベクトルの座標値を、それぞれ始端点と終端点となる輪郭点の座標値と比較することにより同定して、始端点と終端点の間のみの部分の平滑化された非周回型のベクトル列を生成して端点間毎平滑化済 (非周回) ベクトルとする。

【 0 1 3 6 】

そして、次工程である平滑化済ベクトル出力部 3 5 0 を介して、ファイル出力もしくは通信 I / F 等を介して外部装置に出力する。

【 0 1 3 7 】

尚、生成された処理結果は、図 2 9 に示したような形式に準じた形として表現することができる、但し、各輪郭点はそれまでの輪郭点情報の部分をアンカーポイントであるのか否かの属性情報としての付帯情報をもつ形式が好ましい。

【 0 1 3 8 】

(第 1 実施形態の変形例)

図 2 9、図 3 7 (a)、図 3 7 (b) に示したデータ形式は、一例にすぎず、本発明の趣旨はこれに限定されるものではない。例えば、端点間毎平滑化済（非周回）ベクトルは、S V G 形式のベクトル表現をとって出力することも可能である。また、各ベクトル列中の輪郭点数のみをまとめて、輪郭点数テーブル部と各ベクトル列中の各ベクトルのデータ部とを分離して表現することも可能である。例えば、図 2 9 の輪郭線中の総点数の部分のみ第一番目の輪郭から第 a 番目の輪郭までまとめた部分を作り、その後、第一番目の輪郭から第 a 番目の輪郭までの輪郭点データをまとめた部分とするように出力形式を構成してもよい。あるいは、始端点・終端点情報としては、図 3 7 (a) に示した閉ループマーカに限るものではなく、例えば、始端点・終端点が存在するベクトル列を区別することが可能なものであればよい。

【 0 1 3 9 】

(他の実施形態)

なお、本発明の目的は、前述した実施形態の機能を実現するソフトウェアのプログラムコード（コンピュータプログラム）を記録した記憶媒体を、システムあるいは装置に供給することによっても、達成されることは言うまでもない。また、システムあるいは装置のコンピュータ（または C P U や M P U ）が記憶媒体に格納されたプログラムコードを読み出し実行することによっても、達成されることは言うまでもない。このコンピュータプログラムは、上述したフローチャートに記載したステップを情報処理装置に実行させることになる。言い換えると、このコンピュータプログラムは、コンピュータを図 1 に示した各処理部（処理手段）として機能させるためのプログラムである。

【 0 1 4 0 】

この場合、コンピュータ可読記憶媒体から読出されたプログラムコード自体が前述した実施形態の機能を実現することになり、そのプログラムコードを記憶した記憶媒体は本発明を構成することになる。

【 0 1 4 1 】

プログラムコードを供給するための記憶媒体としては、例えば、フレキシブルディスク、ハードディスク、光ディスク、光磁気ディスク、C D - R O M、C D - R、不揮発性のメモ리카ード、R O M などを用いることができる。

【 0 1 4 2 】

また、コンピュータが読出したプログラムコードを実行することにより、前述した実施形態の機能が実現される。また、プログラムコードの指示に基づき、コンピュータ上で稼働している O S（オペレーティングシステム）などが実際の処理の一部または全部を行い、その処理によって前述した実施形態が実現される場合も含まれることは言うまでもない。

【図面の簡単な説明】

【 0 1 4 3 】

【図 1】本発明の実施形態に係る画像処理装置の機能ブロック図を示す図である。

【図 2】2 値画像のアウトラインベクトルによる処理例を示す図である。

【図 3】本発明によるラインアート画像のベクトル処理の流れによる一連の処理経過の例を示す図である。

【図 4】本発明を実施する画像処理装置を実現する機器構成例を示す図である。

【図 5】端点間毎輪郭（周回）ベクトル抽出部の構成例を示す図である。

【図 6】本実施例における注目画素とその近傍画素を示す図である。

【図 7】端点部より検出される輪郭点例とそれらに付与される輪郭点情報を示す図である

10

20

30

40

50

。

【図 8】端点部ではない部分より検出される輪郭点例とそれらに付与される輪郭点情報を示す図である。

【図 9】輪郭点情報の一覧表を示す図である。

【図 10】端点間毎輪郭（周回）ベクトル抽出における注目画素と近傍画素の状態に応じた輪郭点と輪郭点情報の抽出パターン（ケース（case）0）を示す図である。

【図 11】端点間毎輪郭（周回）ベクトル抽出における注目画素と近傍画素の状態に応じた輪郭点と輪郭点情報の抽出パターン（ケース（case）1）を示す図である。

【図 12】端点間毎輪郭（周回）ベクトル抽出における注目画素と近傍画素の状態に応じた輪郭点と輪郭点情報の抽出パターン（ケース（case）2）を示す図である。

10

【図 13】端点間毎輪郭（周回）ベクトル抽出における注目画素と近傍画素の状態に応じた輪郭点と輪郭点情報の抽出パターン（ケース（case）3）を示す図である。

【図 14】端点間毎輪郭（周回）ベクトル抽出における注目画素と近傍画素の状態に応じた輪郭点と輪郭点情報の抽出パターン（ケース（case）4）を示す図である。

【図 15】端点間毎輪郭（周回）ベクトル抽出における注目画素と近傍画素の状態に応じた輪郭点と輪郭点情報の抽出パターン（ケース（case）5）を示す図である。

【図 16】端点間毎輪郭（周回）ベクトル抽出における注目画素と近傍画素の状態に応じた輪郭点と輪郭点情報の抽出パターン（ケース（case）6）を示す図である。

【図 17】端点間毎輪郭（周回）ベクトル抽出における注目画素と近傍画素の状態に応じた輪郭点と輪郭点情報の抽出パターン（ケース（case）7）を示す図である。

20

【図 18】端点間毎輪郭（周回）ベクトル抽出における注目画素と近傍画素の状態に応じた輪郭点と輪郭点情報の抽出パターン（ケース（case）8）を示す図である。

【図 19】端点間毎輪郭（周回）ベクトル抽出における注目画素と近傍画素の状態に応じた輪郭点と輪郭点情報の抽出パターン（ケース（case）9）を示す図である。

【図 20】端点間毎輪郭（周回）ベクトル抽出における注目画素と近傍画素の状態に応じた輪郭点と輪郭点情報の抽出パターン（ケース（case）10）を示す図である。

【図 21】端点間毎輪郭（周回）ベクトル抽出における注目画素と近傍画素の状態に応じた輪郭点と輪郭点情報の抽出パターン（ケース（case）11）を示す図である。

【図 22】端点間毎輪郭（周回）ベクトル抽出における注目画素と近傍画素の状態に応じた輪郭点と輪郭点情報の抽出パターン（ケース（case）12）を示す図である。

30

【図 23】端点間毎輪郭（周回）ベクトル抽出における注目画素と近傍画素の状態に応じた輪郭点と輪郭点情報の抽出パターン（ケース（case）13）を示す図である。

【図 24】端点間毎輪郭（周回）ベクトル抽出における注目画素と近傍画素の状態に応じた輪郭点と輪郭点情報の抽出パターン（ケース（case）14）を示す図である。

【図 25】端点間毎輪郭（周回）ベクトル抽出における注目画素と近傍画素の状態に応じた輪郭点と輪郭点情報の抽出パターン（ケース（case）15）を示す図である。

【図 26】端点間毎輪郭（周回）ベクトル抽出処理の全体的な流れを示すフローチャートである。

【図 27】ベクトル列抽出処理を示すフローチャートである。

【図 28】ベクトル列抽出処理を示すフローチャートである。

40

【図 29】端点間毎周回ベクトル列と輪郭点情報のデータ形式の例を示す図。

【図 30】端点間毎線芯（周回）ベクトル生成部の動作を示すフローチャートである。

【図 31】線素、端点間毎周回ベクトル列と輪郭点情報、端点間毎線芯（周回）ベクトルを例示する図である。

【図 32】始端点の直前と終端点の直後に挿入する補助輪郭点を説明する図である。

【図 33】端点間毎補助ベクトル入り（周回）ベクトル生成部の動作を示すフローチャートである。

【図 34】終端点～始端点間へ挿入する補助ベクトル生成手順を示すフローチャートである。

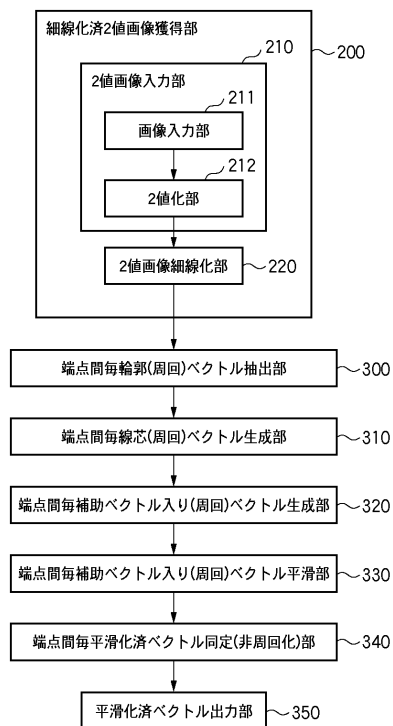
【図 35】終端点～始端点間へ挿入する補助ベクトルを説明する図である。

50

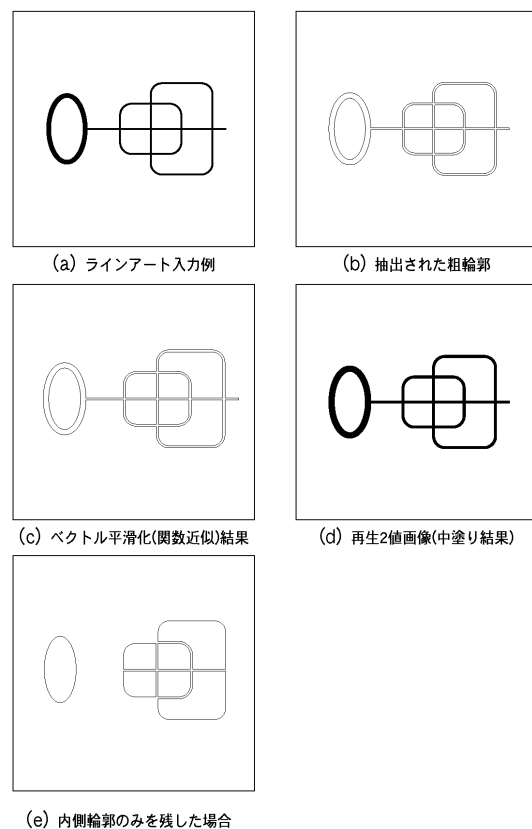
【図 3 6】終端点～始端点間へ挿入する補助ベクトルを説明する図である。

【図 3 7】端点間毎補助ベクトル入り（周回）ベクトル、及び、端点間毎平滑化済ベクトルのデータ形式の例を示す図である。

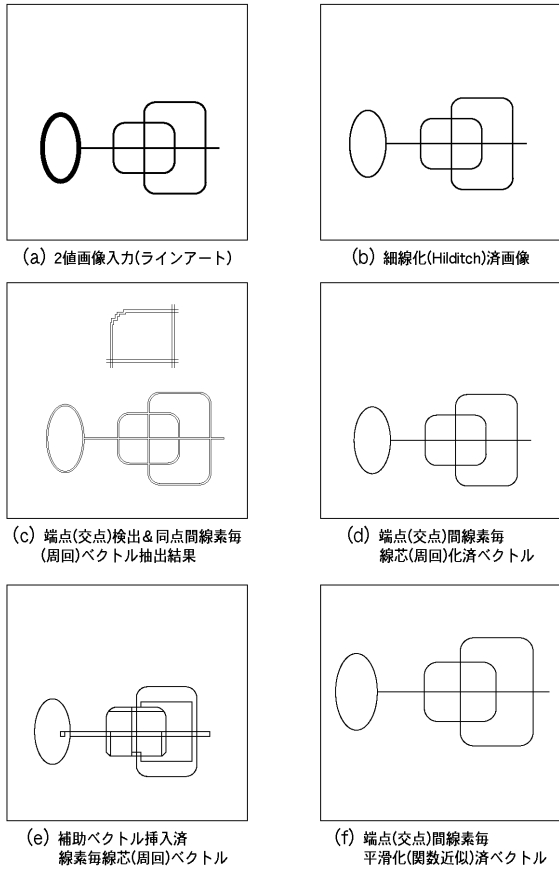
【図 1】



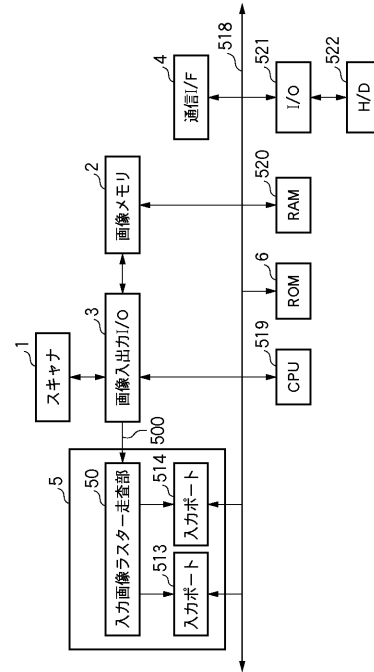
【図 2】



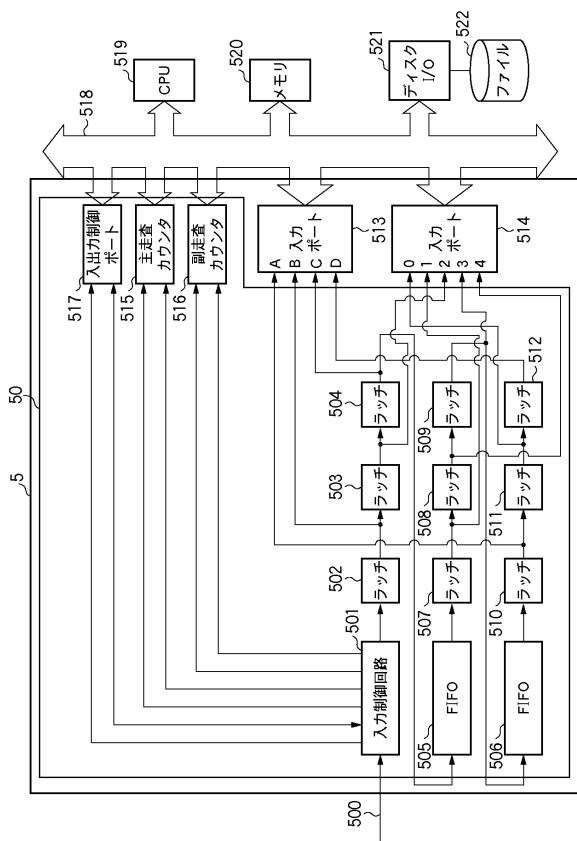
【図 3】



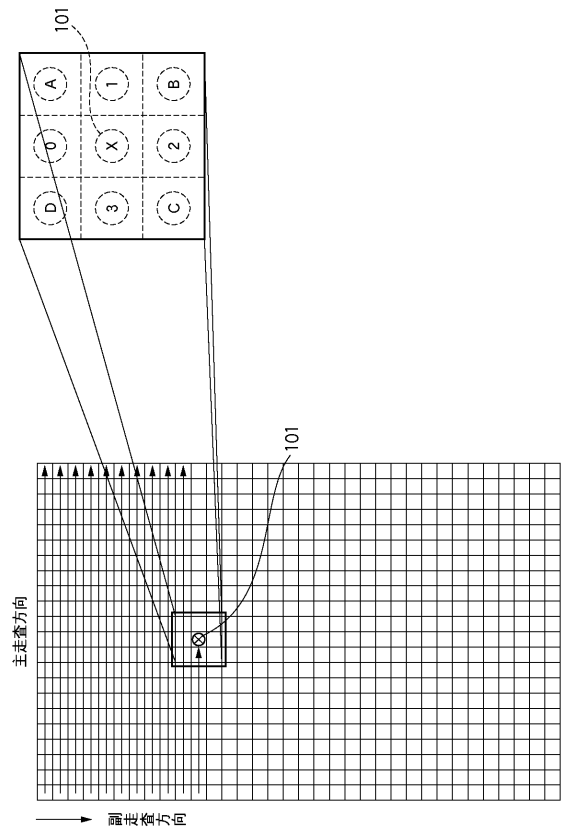
【図 4】



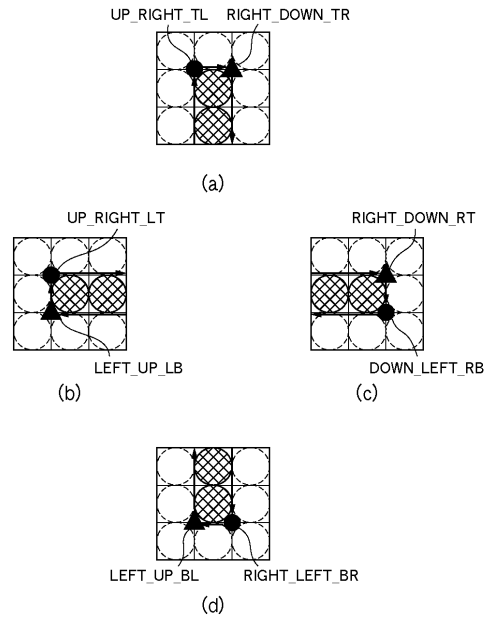
【図 5】



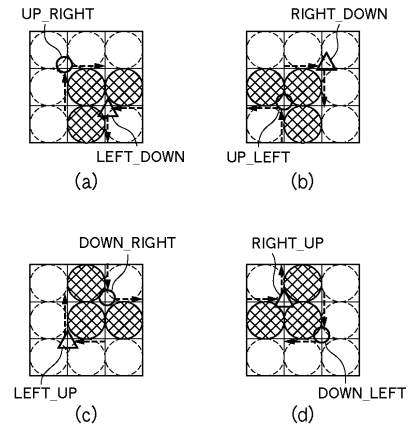
【図 6】



【図 7】



【図 8】

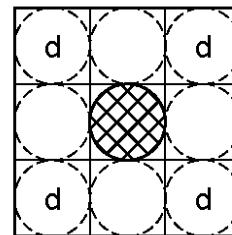


【図 9】

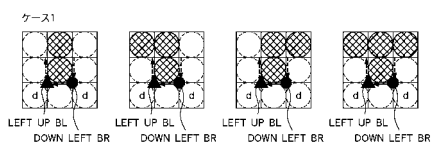
No.	端点情報	端点情報	値	水平(0) 垂直(1)	上向(0) 下向(1)	右向(0) 左向(1)	非選択(0) 選択(1)	端点(1) 非端点(0)
01.	UP_RIGHT_TL (Top Left)		01H	0	0	0	0	1
02.	UP_RIGHT		02H	0	0	0	1	0
03.	UP_RIGHT_LT (Left Top)		03H	0	0	0	1	1
04.	UP_LEFT		06H	0	0	1	1	0
05.	DOWN_RIGHT		0aH	0	1	0	1	0
06.	DOWN_LEFT_BR (Bottom Right)		0dH	0	1	1	0	1
07.	DOWN_LEFT		0eH	0	1	1	1	0
08.	DOWN_LEFT_RB (Right Bottom)		0fH	0	1	1	1	1
09.	RIGHT_UP		12H	1	0	0	1	0
10.	LEFT_UP_LB (Left Bottom)		15H	1	0	1	0	1
11.	LEFT_UP		16H	1	0	1	1	0
12.	LEFT_UP_BL (Bottom Left)		17H	1	0	1	1	1
13.	RIGHT_DOWN_RT (Right Top)		19H	1	1	0	0	1
14.	RIGHT_DOWN		1aH	1	1	0	1	0
15.	RIGHT_DOWN_TR (Top Right)		1bH	1	1	0	1	1
16.	RIGHT_DOWN		1eH	1	1	1	1	0

【図 10】

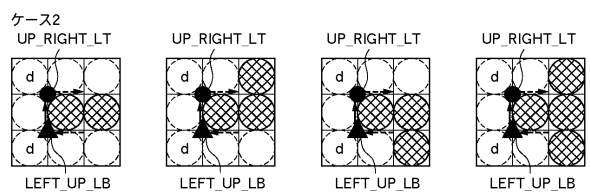
ケース0



【図 11】

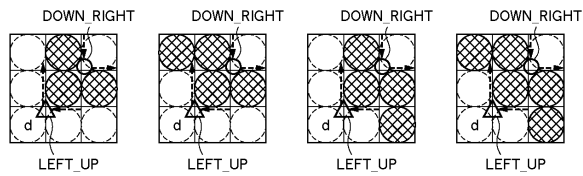


【図 12】



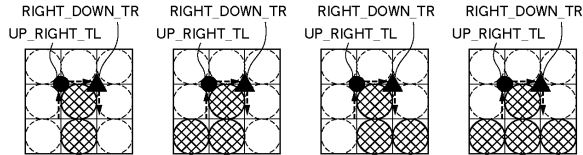
【図 13】

ケース3



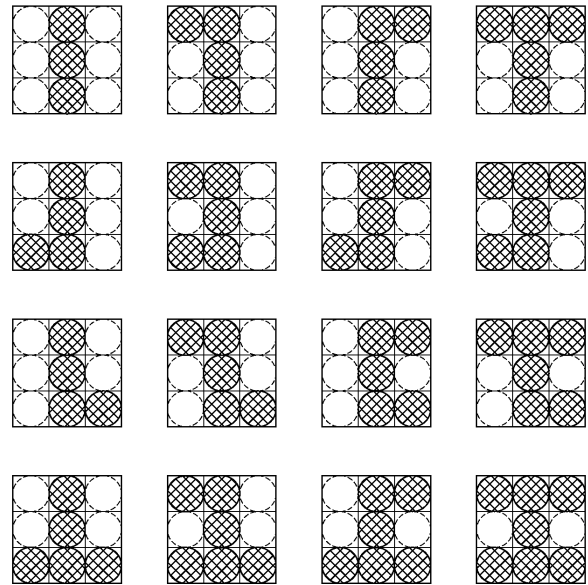
【図 14】

ケース4



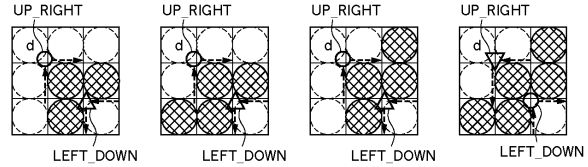
【図 15】

ケース5



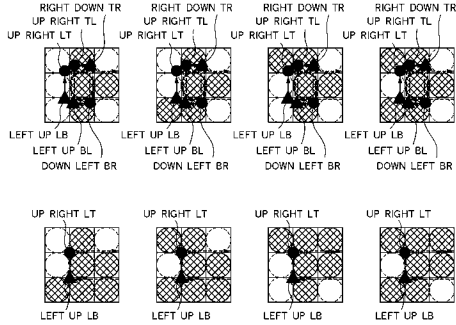
【図 16】

ケース6



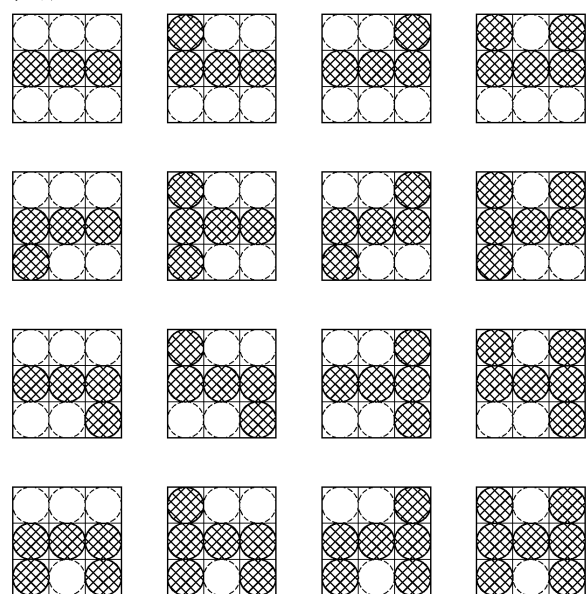
【図 17】

ケース7



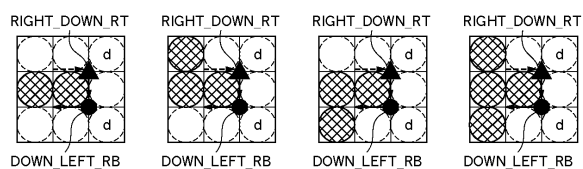
【図 20】

ケース10



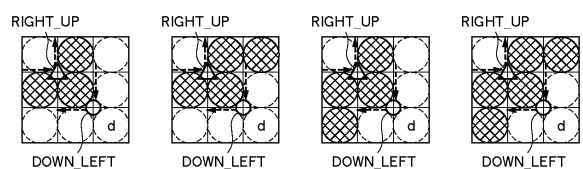
【図 18】

ケース8

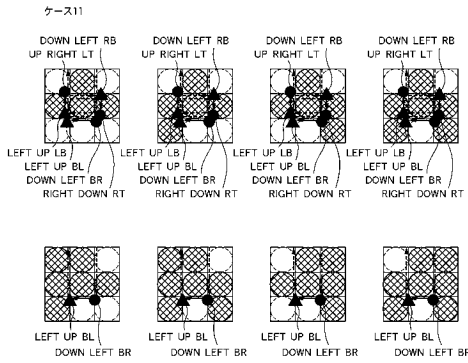


【図 19】

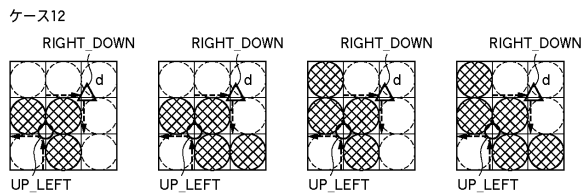
ケース9



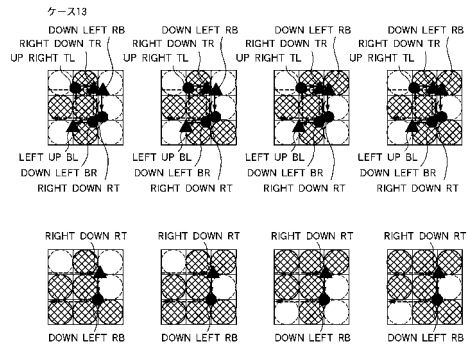
【図 2 1】



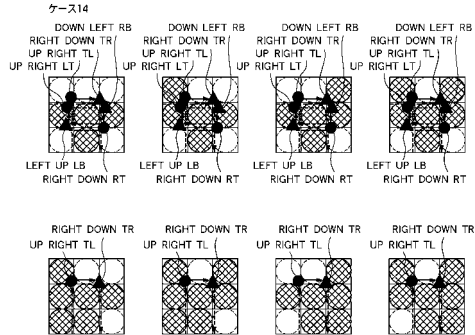
【図 2 2】



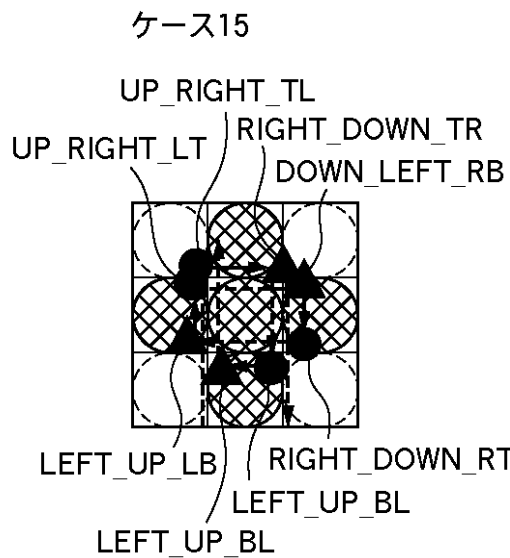
【図 2 3】



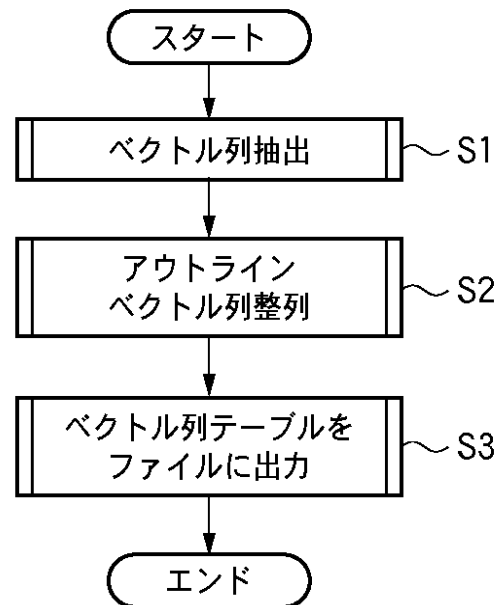
【図 2 4】



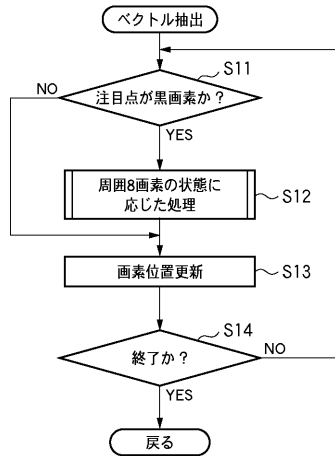
【図 2 5】



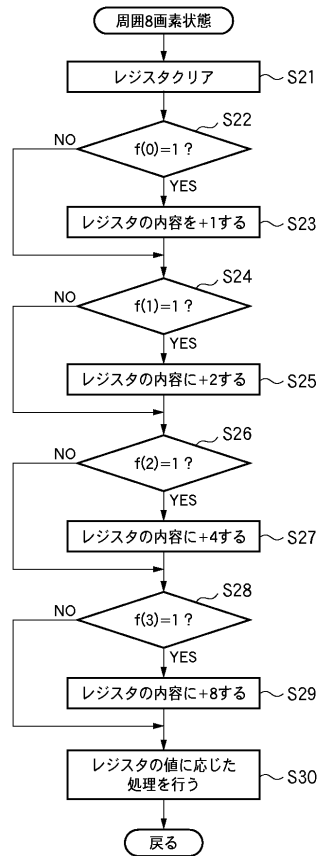
【図 2 6】



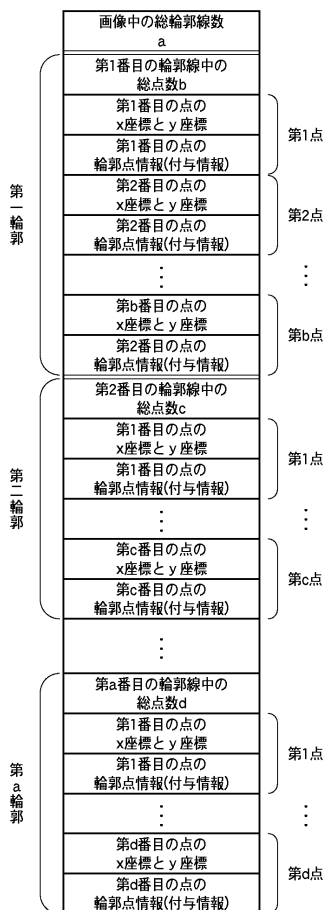
【図 27】



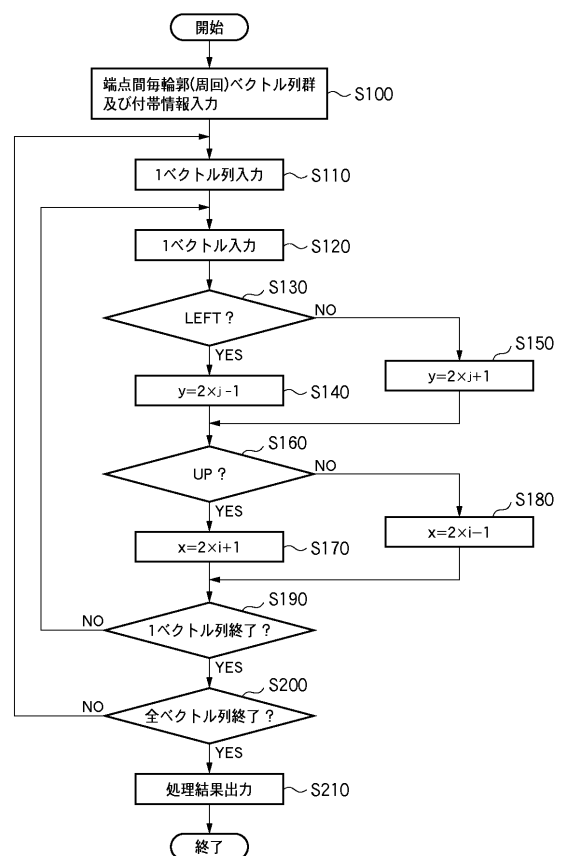
【図 28】



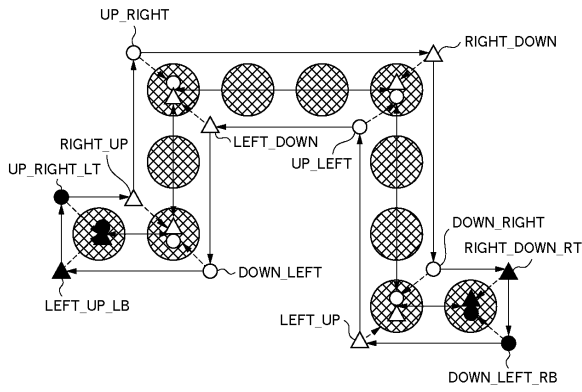
【図 29】



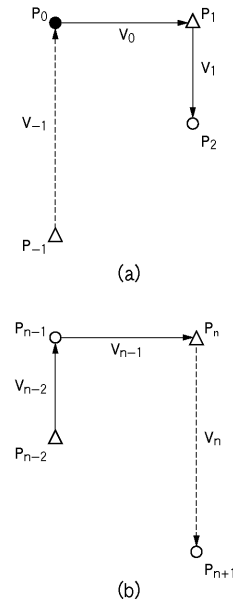
【図 30】



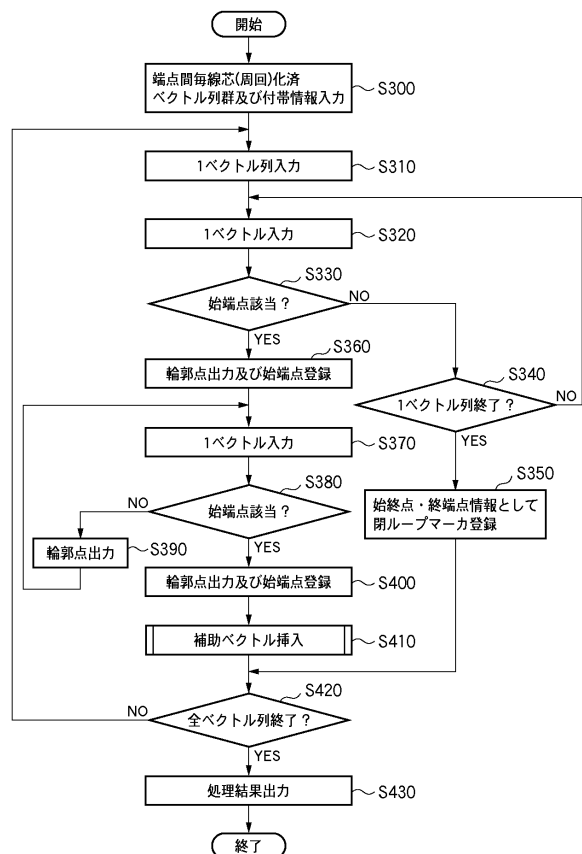
【図 3 1】



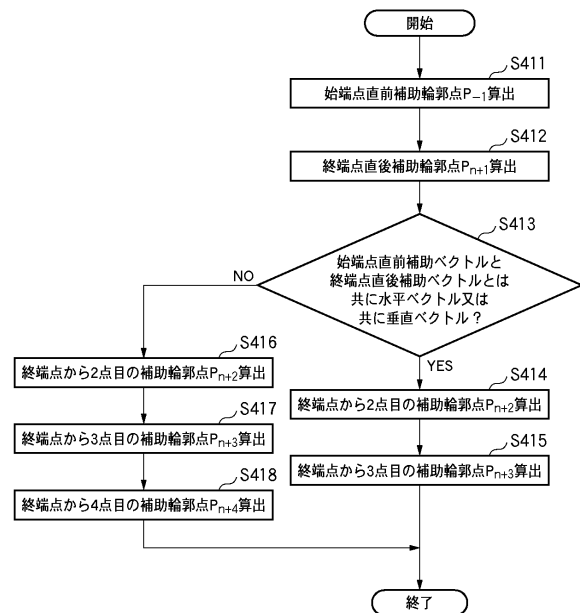
【図 3 2】



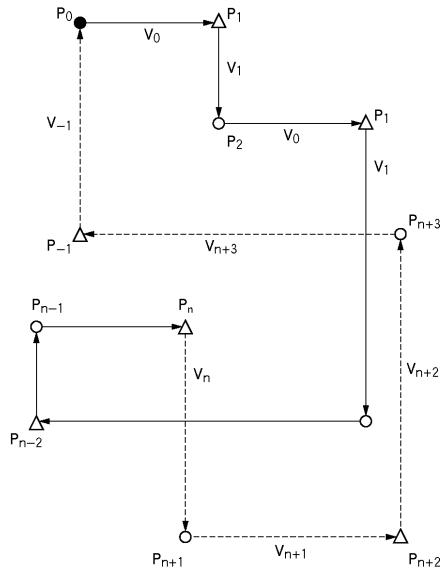
【図 3 3】



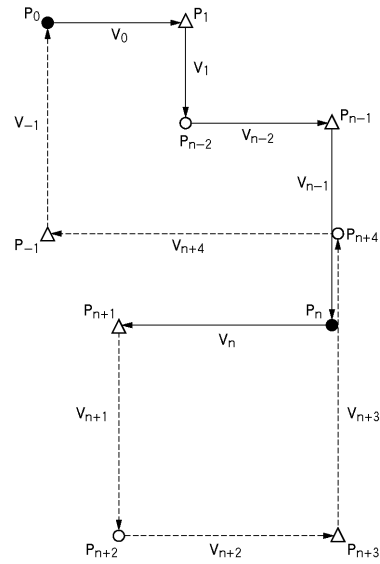
【図 3 4】



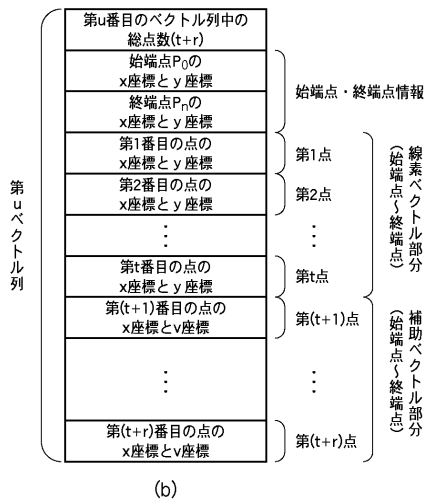
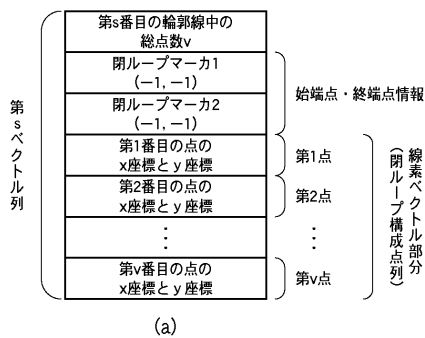
【図 3 5】



【図 3 6】



【図 3 7】



フロントページの続き

審査官 佐田 宏史

- (56)参考文献 特開 2 0 0 1 - 2 2 2 7 1 8 (J P , A)
特開平 0 5 - 1 7 4 1 4 0 (J P , A)
特開平 1 1 - 1 3 4 4 9 2 (J P , A)
特開平 1 1 - 2 7 2 8 6 9 (J P , A)
特開平 9 - 1 0 2 0 4 4 (J P , A)
特開平 4 - 1 5 7 5 7 8 (J P , A)
特開昭 6 1 - 0 0 5 3 7 5 (J P , A)
特開昭 6 1 - 0 5 1 2 7 1 (J P , A)
特開平 4 - 0 8 8 4 8 0 (J P , A)
欧州特許出願公開第 1 5 3 8 5 4 7 (E P , A 1)

- (58)調査した分野(Int.Cl. , D B 名)
G 0 6 T 3 / 0 0 , 5 / 3 0 , 7 / 6 0 , 9 / 2 0 , 1 1 / 2 0