

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第3576611号

(P3576611)

(45) 発行日 平成16年10月13日(2004.10.13)

(24) 登録日 平成16年7月16日(2004.7.16)

(51) Int.Cl.⁷

F I

G O 9 G 5/24

G O 9 G 5/24 6 2 O L

G O 6 T 11/20

G O 9 G 5/24 6 3 O B

G O 9 G 5/24 6 5 O M

G O 6 T 11/20 1 1 O U

請求項の数 8 (全 27 頁)

(21) 出願番号 特願平6-300879
 (22) 出願日 平成6年12月5日(1994.12.5)
 (65) 公開番号 特開平8-160931
 (43) 公開日 平成8年6月21日(1996.6.21)
 審査請求日 平成13年11月19日(2001.11.19)

(73) 特許権者 000001007
 キヤノン株式会社
 東京都大田区下丸子3丁目30番2号
 (74) 代理人 100077481
 弁理士 谷 義一
 (74) 代理人 100088915
 弁理士 阿部 和夫
 (72) 発明者 吉田 政幸
 東京都大田区下丸子3丁目30番2号 キ
 ヤノン株式会社内

審査官 後藤 亮治

最終頁に続く

(54) 【発明の名称】 文字発生装置ならびに方法

(57) 【特許請求の範囲】

【請求項1】

文字の輪郭の太め処理を行い、太め処理された輪郭に基づき文字を発生させる文字発生装置であって、

太め処理された対象輪郭点とその両隣の輪郭点のなす角度が鋭い角度であるか否かを判別する判別手段と、

前記判別手段により、太め処理された対象輪郭点とその両隣の輪郭点のなす角度が鋭い角度であると判別された場合、前記太め処理された対象輪郭点を補正する補正手段とを有し

、
 前記補正手段は、前記太め処理の太め量が負の場合、前記太め処理された対象輪郭点を、
 前記対象輪郭点と当該対象輪郭点の処理前の輪郭点とを結ぶ直線上で、処理前の輪郭点側の新たな輪郭点に移動させることを特徴とする文字発生装置。

10

【請求項2】

前記補正手段は、前記太め処理の太め量が正の場合、太め処理された対象輪郭点および当該対象輪郭点の処理前の輪郭点とを結ぶ直線に直角に交わる直線と前記対象輪郭点およびその両隣の輪郭点を結ぶ二つの直線との交点を新たな輪郭点として求め、当該新たな輪郭点に前記対象輪郭点を置き換えることにより対象輪郭点を補正することを特徴とする請求項1に記載の文字発生装置。

【請求項3】

前記補正手段は、角を丸める必要がある場合、前記求められた新たな輪郭点に基づき曲線

20

構成点を求めることを特徴とする請求項 2 に記載の文字発生装置。

【請求項 4】

前記太め処理により太められた輪郭点の座標値を元の文字の文字枠の幅と太め量とに基づき調整する調整手段をさらに有することを特徴とする請求項 1 に記載の文字発生装置。

【請求項 5】

文字の輪郭の太め処理を行い、太め処理された輪郭に基づき文字を発生させる文字発生方法であって、

太め処理された対象輪郭点とその両隣の輪郭点のなす角度が鋭い角度であるか否かを判別する判別ステップと、

前記判別ステップにより、太め処理された対象輪郭点とその両隣の輪郭点のなす角度が鋭い角度であると判別された場合、前記太め処理された対象輪郭点を補正する補正ステップとを有し、

前記補正ステップは、前記太め処理の太め量が負の場合、前記太め処理された対象輪郭点を、前記対象輪郭点と当該対象輪郭点の処理前の輪郭点とを結ぶ直線上で、処理前の輪郭点側の新たな輪郭点に移動させることを特徴とする文字発生方法。

【請求項 6】

前記補正ステップは、前記太め処理の太め量が正の場合、太め処理された対象輪郭点および当該対象輪郭点の処理前の輪郭点とを結ぶ直線に直角に交わる直線と前記対象輪郭点およびその両隣の輪郭点を結ぶ二つの直線との交点を新たな輪郭点として求め、当該新たな輪郭点に前記対象輪郭点を置き換えることにより対象輪郭点を補正することを特徴とする請求項 5 に記載の文字発生方法。

【請求項 7】

前記補正ステップは、角を丸める必要がある場合、前記求められた新たな輪郭点に基づき曲線構成点を求めることを特徴とする請求項 6 に記載の文字発生方法。

【請求項 8】

前記太め処理により太められた輪郭点の座標値を元の文字の文字枠の幅と太め量とに基づき調整する調整ステップをさらに有することを特徴とする請求項 5 に記載の文字発生方法。

【発明の詳細な説明】

【0001】

【産業上の利用分野】

本発明は、輪郭の座標データでコード化された字の太さを変換する文字発生装置および方法に関する。

【0002】

【従来の技術】

従来、輪郭の座標データで格納されたデータをビットマップに展開して文字を出力する場合、ROM やハードディスクに格納された輪郭の座標データを読み出し、その座標データに拡大縮小率をかけて所望のサイズに変換した後にドット形式に変換して文字データを得ていた。

【0003】

この場合、同一の書体でもウエイト（太さ、肉厚量）が異なればそれぞれに対して 1 つずつ座標データを持つ必要がある。しかし和文書体の場合 1 書体当たり 8000 文字程度あり、その 1 書体に必要な輪郭データの容量は 1 M バイトから 3 M バイトほど必要である。そしてそれぞれのウエイトに対して輪郭の座標データを持たせることになればその容量は膨大な容量になってしまうという問題点があった。

【0004】

このため、1 つの書体のデータから同一書体で異なるウエイトのデータを発生させるときに、座標の対象座標と隣接する 2 点の 3 点から太め量あるいは細め量を基準にして計算して求めていた（本出願人が先に出願した特願平 5 - 309555 号及び特願平 5 - 309556 号参照）。

10

20

30

40

50

【 0 0 0 5 】

【 発明が解決しようとする課題 】

しかしながら上記の場合、3点のなす角度が例えば30度以下のような鋭く折れている直線を変換しようとする、対象点が、文字領域をはみ出してしまったり、著しく文字の形を損ねてしまうという問題点があった。

【 0 0 0 6 】

本発明は、上記の問題点を解消するためになされたもので、少なくとも1つの書体のデータから同一書体での異なるウエイト(太さ)のデータを良好な品質で発生させ、少ないメモリ容量で、様々なウエイトの書体のデータを発生させることを可能とすることを目的としたものである。

10

【 0 0 0 7 】

【 課題を解決するための手段 】

本発明は、このような目的を達成するために、請求項1に記載の発明は、文字の輪郭の太め処理を行い、太め処理された輪郭に基づき文字を発生させる文字発生装置であって、太め処理された対象輪郭点とその両隣の輪郭点のなす角度が鋭い角度であるか否かを判別する判別手段と、前記判別手段により、太め処理された対象輪郭点とその両隣の輪郭点のなす角度が鋭い角度であると判別された場合、前記太め処理された対象輪郭点を補正する補正手段とを有し、前記補正手段は、前記太め処理の太め量が負の場合、前記太め処理された対象輪郭点を、前記対象輪郭点と当該対象輪郭点の処理前の輪郭点とを結ぶ直線上で、
処理前の輪郭点側の新たな輪郭点に移動させることを特徴とするものである。

20

【 0 0 0 8 】

また、請求項2に記載の発明は、請求項1に記載の文字発生装置であって、太め処理された対象輪郭点および当該対象輪郭点の処理前の輪郭点とを結ぶ直線に直角に交わる直線と前記対象輪郭点およびその両隣の輪郭点を結ぶ二つの直線との交点を新たな輪郭点として求め、当該新たな輪郭点に前記対象輪郭点を置き換えることにより対象輪郭点を補正することを特徴とするものである。

【 0 0 0 9 】

また、請求項3に記載の発明は、請求項2に記載の発明であって、前記補正手段は、角を丸める必要がある場合、前記求められた新たな輪郭点に基づき曲線構成点を求めることを特徴とするものである。

30

【 0 0 1 0 】

また、請求項4に記載の発明は、請求項1に記載の文字発生装置であって、前記太め処理により太められた輪郭点の座標値を元の文字の文字枠の幅と太め量とに基づき調整する調整手段をさらに有することを特徴とするものである。

【 0 0 1 1 】

また、請求項5に記載の発明は、文字の輪郭の太め処理を行い、太め処理された輪郭に基づき文字を発生させる文字発生方法であって、太め処理された対象輪郭点とその両隣の輪郭点のなす角度が鋭い角度であるか否かを判別する判別ステップと、前記判別ステップにより、太め処理された対象輪郭点とその両隣の輪郭点のなす角度が鋭い角度であると判別された場合、前記太め処理された対象輪郭点を補正する補正ステップとを有し、前記補正ステップは、前記太め処理の太め量が負の場合、前記太め処理された対象輪郭点を、前記対象輪郭点と当該対象輪郭点の処理前の輪郭点とを結ぶ直線上で、
処理前の輪郭点側の新たな輪郭点に移動させることを特徴とする。

40

【 0 0 1 2 】

また、請求項6に記載の発明は、請求項5に記載の文字発生方法であって、前記太め処理の太め量が正の場合、太め処理された対象輪郭点および当該対象輪郭点の処理前の輪郭点とを結ぶ直線に直角に交わる直線と前記対象輪郭点およびその両隣の輪郭点を結ぶ二つの直線との交点を新たな輪郭点として求め、当該新たな輪郭点に前記対象輪郭点を置き換えることにより対象輪郭点を補正することを特徴とする。

【 0 0 1 3 】

50

また、請求項 7 に記載の発明は、請求項 6 に記載の文字発生方法であって、前記補正ステップは、角を丸める必要がある場合、前記求められた新たな輪郭点に基づき曲線構成点を求めることを特徴とする。

【 0 0 1 4 】

また、請求項 8 に記載の発明は、請求項 5 に記載の文字発生方法であって、前記太め処理により太められた輪郭点の座標値を元の文字の文字枠の幅と太め量とに基づき調整する調整ステップをさらに有することを特徴とする。

【 0 0 2 0 】

【作用】

本発明は、入力された太字量または細字量に基づいて、輪郭データの座標値を変換し、調整すべき座標を抽出し、座標値を調整し、調整された輪郭データに基づいて太いまたは細い文字パターンを発生させるので、少ない文字データから肉厚量の異なる高品質な太字または細字文字を発生するものである。

10

【 0 0 2 1 】

本発明においては、入力された太字量または細字量に基づいて、変換するいずれかの文字データ候補を決定し、該決定された文字データに対応する輪郭データの座標値を変換し、該座標変換された輪郭データに基づいて、太いまたは細い文字パターンを発生して、文字データ資源を生かして最良の太字または細字の文字データを発生させるものである。

【 0 0 2 2 】

本発明においては、決定は、入力された太字量または細字量に近い肉厚量の文字データを変換する文字データ候補として、より厳密な太字または細字の文字データを発生させるものである。

20

【 0 0 2 3 】

また、本発明においては、変換は、対象座標に隣接する 2 点の座標値を参照して 1 つの変換座標を決定することにより行うので、バランスのとれた太いまたは細字の文字パターンを発生するものである。

【 0 0 2 4 】

本発明においては、変換は、抽出した各輪郭データを入力された太字量または細字量を x 方向および y 方向に対して独立して可変するにより行うので、書体毎の特徴を生かした太字または細字の文字パターンを発生するものである。

30

【 0 0 2 5 】

本発明においては、判定は、対象座標と隣接する 2 点の座標値を参照して 3 点のなす角度および入力された値が太字量あるいは細字量であるかおよび対象書体によって決定することにより行うので、良好な品質の太字または細字の文字パターンを発生するものである。

【 0 0 2 6 】

本発明においては、座標調整は、太字量が指定されたときに対象点に新たな点を挿入することにより行うので、良好な品質の太字の文字パターンを発生するものである。

【 0 0 2 7 】

本発明においては、座標調整は、太字量が指定されたときに対象点に新たな点を挿入した後、角を丸めることによって行うので、書体毎の特徴を生かした良好な太字の文字パターンを発生するものである。

40

【 0 0 2 8 】

本発明においては、座標調整は、細字量が指定されたときに対象点を移動させることによって行うので、良好な品質の細字の文字パターンを発生するものである。

【 0 0 2 9 】

本発明においては、データの発生は、変換された輪郭データに基づいてビットマップフォント、輪郭データ、グレイスケールフォントのいずれかを出力して、種々の出力手段に適切なデータ形式で太字または細字の文字データを供給するものである。

【 0 0 3 0 】

前記輪郭データは、交差するストロークによる複数の輪郭データで構成されていてもよい

50

し、交差しない複数の文字の輪郭データで構成されていてもよい。

【0031】

【実施例】

次に、本発明の実施例について図を参照して説明する。

【0032】

図1は本発明に使用するシステムの基本的な構成を示すブロック図である。本発明に使用するシステムは、日本語ワードプロセッサであってもよいし、ワークステーションあるいはコンピュータシステムであってもよい。

【0033】

図1において、1はCPU、すなわち中央処理装置であり、この装置全体の制御および演算処理等を行うものである。2はROMすなわち読み出し専用メモリであり、システム起動プログラムおよび文字パターン・データ等の記憶領域である。3はRAMすなわちランダムアクセスメモリであり、使用制限のないデータ記憶領域であり、様々な処理毎に各々のプログラムおよびデータが記憶され、実行される領域である。4はKBCすなわちキーボード制御部であり、5のKBすなわちキーボードよりキー入力データを受け取りCPU1へ伝達する。6はCRTCすなわちディスプレイ制御部であり、7はCRTすなわちディスプレイ装置であり、6のCRTCよりデータを受け取り表示する。9はFDすなわちフロッピーディスク装置あるいはHDすなわちハードディスク装置等の外部記憶装置であり、プログラムおよびデータを記憶させておき、実行時必要に応じて参照またはRAMへロードする。8はDKCすなわちディスク制御部であり、データ伝送等の制御を行うものであり、10はPRTCすなわちプリンタ制御部であり、11はPRTすなわちプリンタ装置である。12はシステムバスであり、上述の構成要素間のデータの通路となるべきものである。

【0034】

また、図2も本発明にかかる本システムの基本的な構成を示すブロック図である。本システムはホストコンピュータ（図示せず）より印刷データ（イメージデータ、文字コード、コマンド等）を受け取り印刷を行うレーザビームプリンタであってもよいし、インクジェットプリンタあるいは熱転写等の出力機であってもよい。図2において21はCPU、すなわち中央処理装置であり、この装置全体の制御および演算処理等を行うものである。22はROMすなわち読み出し専用メモリであり、システム起動プログラムおよび文字パターン・データ等の記憶領域である。23はRAMすなわちランダムアクセスメモリであり、使用制限のないデータ記憶領域であり、様々な処理毎に各々のプログラムおよびデータが記憶され、実行される領域である。10はPRTCすなわち出力制御部であり、11はPRTすなわち印刷装置であり、10のPRTCよりデータを受け取り印刷する。

【0035】

〔第1実施例〕

次に、本発明の一実施例の詳細を図3のフローチャートを用いて説明する。図3は、字を構成するストロークごとにそのストロークを構成する輪郭の座標データで文字データが表現されている場合における、全体の流れを示すフローチャートである。このストロークで構成された輪郭は、それぞれ交差する輪郭である。ここで説明する例は、ウエイト（肉厚）の異なる同一の書体がシステムに1つ存在するとき、そのデータを用いて異なるウエイトのデータを発生させるときの例である。

【0036】

ステップ3-1において図1のシステムにより実行されるアプリケーションプログラム、あるいは、図2のシステムにデータを供給するホストコンピュータ（図示せず）より入力パラメータを受け取る。ここで入力パラメータとしては出力すべき文字の文字コード、書体、ウエイト、出力サイズ、出力形式等がある。文字コードとしてはJISコード、シフトJISコード、EUCコード、UNIコード等のあらかじめ対象となるシステムがどの文字コード体系によって定められているかによって決まる。また書体としては明朝体、ゴシック体、丸ゴシック体等のシステムがあらかじめ内蔵しているデータあるいはオプショ

10

20

30

40

50

ンとして加えられたデータの中から選ばれる。ウエイトは前記書体の線の太さ情報であり、ここでは極細、細、中、太、極太等の情報が与えられる。出力サイズは実際にフォントデータを出力する際にどれ位の大きさで出力するかの情報である。出力形式は所望とするフォントの出力データ形式であり、輪郭の座標データ出力、ビットマップ出力等の要求が出される。

【0037】

次にステップ3 - 2において対象文字の座標データを読み込む。このデータはROM、RAM、ハードディスク、フロッピーディスク等にあらかじめ格納されており、ステップ3 - 1で取り込んだ入力情報の書体情報や文字コード情報から検索して必要分読み込む。このとき取り込む入力情報は、図4に示すように、文字の輪郭の特徴点を抽出した座標情報であり、それぞれの点に対して直線データ/曲線データ判別フラグ、輪郭開始点/終了点フラグ等の属性情報を持つ。そしてここで扱う曲線データの補間式は2次あるいは2次Bスプライン曲線であったり、2次あるいは3次ベジェ曲線であったりするが、どの補間式を使用しているかはあらかじめ決定されている。また、文字枠を示す座標の最小値は0、最大値は800等で表現されている。また、各ストロークの枠の基準点への文字の原点からのオフセット情報を持つ。この座標データは、図5に示すようなフォーマットでROM等に格納されている。

【0038】

そして、ステップ3 - 3においては取り込んだ座標データを入力パラメータのウエイト情報に応じて太め/細め処理を行う。この時の処理は図10のフローチャートを用いて後で詳述するが、この太め/細め処理を行った結果、各々の輪郭線が太め/細め座標に変換される。図6には、図4に示す字形を太めの座標に変換した文字が示されている。このとき、太め/細め処理前と太め/細め処理後の座標点は1対1に対応しており、それぞれの点の持つ属性フラグは変化しない。そしてステップ3 - 4において、ステップ3 - 3で得られた太め/細め座標データを入力パラメータの出力サイズに応じて拡大縮小処理を行う。この時の計算方法は、要求出力サイズを(A_x , A_y)、ステップ3 - 3で得られたそれぞれの座標値を(x , y)、拡大縮小処理後のそれぞれの座標値を(X , Y)、格納されている文字枠のサイズを(M_x , M_y)とすると、

【0039】

【数1】

$$(X, Y) = (x \times A_x / M_x, y \times A_y / M_y)$$

となる。上記計算を1文字が持つすべての座標列において計算する。また、このときステップ3 - 3で得られた各座標点における属性フラグは変化しない。

【0040】

ステップ3 - 5においては、入力パラメータの出力形式で判定し、出力形式が輪郭の座標データ出力であれば、ステップ3 - 6に進み、ステップ3 - 4で得られた拡大縮小後の座標点および座標点属性のデータ列を要求側へ返す。出力される輪郭の座標データのフォーマットは図5と同様である。

【0041】

ステップ3 - 5において、ビットマップ出力が要求されていれば、ステップ3 - 7へ進む。ステップ3 - 7からステップ3 - 13においては、実際に座標データからビットマップのデータを作成する処理である。ステップ3 - 7においては、対象となる座標データが直線であるかあるいは曲線であるかを判定する。対象となる座標データが直線である場合は、その座標点を直線のスタート点とし、次の座標点を直線のエンド点としてステップ3 - 8へ進む。対象となるデータが曲線データである場合はその座標点から曲線終了フラグが付されている座標データまでを曲線データとしてステップ3 - 9へ進む。

【0042】

ステップ3 - 8においては、曲線を発生させる処理を行う。このときの直線発生方法はDDA(Digital Differential Algorithm)によって発生させる。そしてDDAによって発生させた座標データは図7に示すように打点テーブルに

10

20

30

40

50

格納する。図 7 に示す打点テーブルは出力領域の各 y 座標に対して、x 座標のスタート座標 / ストップ座標を格納していく。DDA によって同一の y 座標に対して複数の x 座標が存在する場合は、ストロークの輪郭に対して最も外側になるように x 座標を設定する。なお、図 7 の打点テーブルは、曲線データによる塗りつぶしデータも入力されている。

【 0 0 4 3 】

ステップ 3 - 9 においては、曲線データを短い直線（ショートベクトル）の集合に変換する処理を行う。図 8 に 3 次ベジェ曲線をショートベクトルの集合に変換する様子を示している。点 A, B, C, D はステップ 3 - 3 より得られた座標変換後の曲線データ（3 次ベジェ曲線構成点）であり、これらの点からまず各点を結んだ直線の中点である点 a, 点 b, 点 c を求める。点 a は点 A と点 B の中点、点 b は点 B と点 C の中点、点 c は点 C と点 D の中点である。そして次に点 x, 点 y, 点 z を求める。また、点 a ~ c を結んだ中点を求める。点 x は点 a と点 b の中点、点 z は点 b と点 c の中点、点 y は点 x と点 z の中点である。そうすると点列 A a x y が新しい 1 つの 3 次ベジェ構成点、そして点列 y z c D がもう 1 つの 3 次ベジェ構成点となる。そしてそれぞれのベジェ構成点を同様な操作で細分化してき、ある判定基準を満たしたらそのとき細分化を中止する。そしてそれまでにできた 3 次ベジェの構成点列がショートベクトルの集合となる。

【 0 0 4 4 】

そしてステップ 3 - 10 において、ステップ 3 - 9 で求めたショートベクトルの集合に基づいて、打点テーブルに格納する。このテーブルに対する格納方法は、ステップ 3 - 8 で示した方法とまったく同様であり、すべてのショートベクトルに対して処理を終了するまで繰り返す。そしてステップ 3 - 11 において 1 つの輪郭の座標データがすべて終了したかどうかを判定し、処理が終了していればステップ 3 - 13 へ進み、処理が終了していなければステップ 3 - 12 に進む。ステップ 3 - 12 においては次のデータを処理するために現在の座標データへのポインタを更新する。直線の場合であれば次の座標データへポインタを更新し、曲線データであれば曲線の終了座標点までポインタを更新する。そしてステップ 3 - 7 に戻って新たに直線 / 曲線判定をして打点を行っていく。ステップ 3 - 13 においては 1 文字分すべての座標データに対して処理を終了したかどうかを判定し、すべての輪郭に対して処理を終了していればステップ 3 - 15 に進み、処理を終了していなければステップ 3 - 14 に進む。そしてステップ 3 - 14 においては、1 輪郭が終了した後のので次の輪郭の先頭にポインタを進めステップ 3 - 7 に戻る。そして、最終的に、図 7 に示す打点テーブルが完成する。

【 0 0 4 5 】

ステップ 3 - 15 においてはすべての座標データに対して打点処理が終了しているので、図 9 に示すように、ステップ 3 - 8 およびステップ 3 - 10 で打点テーブルに格納された各 y 座標に対する x 座標に対してノンゼロワイディング方式で塗りつぶしを行う。この方式は各スキャンラインの左側からスキャンしていき、スタート点であればフラグの値をインクリメントし、エンド点であればデクリメントする。そしてフラグの値が 0 でなければその間は 1 として塗りつぶし処理を行う。そしてステップ 3 - 16 において要求側の指定する領域にステップ 3 - 15 で求められた 1 文字分のデータを返して処理を終了する。なお、要求側が P R T C 10 であれば P R T C 10 にデータを返し、P R T 11 により印刷を行う。また、要求側が C R T C 6 であれば C R T C 6 にデータを返し、C R T 7 により表示を行う。

【 0 0 4 6 】

次に、図 10 のフローチャートを用いて、図 3 のステップ 3 - 3 における太め / 細め処理の実施例を詳述する。本発明における太め / 細めの処理は、ストロークに対応して太める / 細めることにより、パラメータを変化させ、それぞれの輪郭点座標を変更する。図 10 のステップ 10 - 1 ではウエイトに応じて輪郭の太さのパラメータを決定する。太さを決定するパラメータは輪郭に大してそれぞれ x 方向、y 方向の独立に値を持ち、横線の太め量および縦線の太め量をそれぞれ独立に管理する。例えば、明朝体を太めようとした場合、明朝体の横線はさほど太くせずに済むのに縦線は大きく太める必要があるため、x 方向

10

20

30

40

50

および y 方向に対してそれぞれ異なる値を設定する必要があるからである。これは例えば、図 11 の中明朝と太明朝とを比較すると明白である。一方、図 12 に示すように丸ゴシック体の場合は、横方向および縦方向はほぼ同じ量だけ太らせる必要がある。従って、それぞれの書体に対しても太めるべき値を変化させる必要がある。そこで、図 13 に示すように、それぞれの書体とウエイトにおける横線と縦線に対して線の中心から輪郭までの標準値のデータをあらかじめテーブルでもたせ、入力された対象座標データの書体およびウエイトと出力したい座標データのウエイトから輪郭の太め量をそれぞれ x 方向および y 方向に対して決定する。このときこの太め量が正の値となる場合は太め処理が行われ、太め量が負の値となる場合には細め処理が行われることになる。

【0047】

10

ステップ 10 - 2 においては、各ストロークのオフセット情報を太さパラメータに応じて変化させる。この値は、オフセットの x 座標および y 座標から太さパラメータの x 値および y 値を減じることによって求められる。

【0048】

ステップ 10 - 3 からステップ 10 - 14 までの間で 1 輪郭構成点のすべての座標点列に対して、太め / 細め処理を行う。ステップ 10 - 3 では、まず処理を施すための対象点を取り込む。そしてステップ 10 - 4 で対象点の両隣の点を取り込む。ステップ 10 - 5 において、対象点とその両隣の点のなす角度を求める。この時の計算式は以下になる。対象点を点 a とし、その両隣の点をそれぞれ点 b, 点 c とすると、ベクトル a b とベクトル a c からその間の角度を θ とすると、

20

【0049】

【数 2】

$$\cos \theta = \frac{\overrightarrow{ab} \cdot \overrightarrow{ac}}{|\overrightarrow{ab}| \times |\overrightarrow{ac}|}$$

【0050】

となり、 θ を求めることができる。そしてステップ 10 - 6 に進み、 θ の角度によって移動すべき点の動きを変化させる。この角度が鋭くない角（たとえば θ の値が 30 度以上）のときはステップ 10 - 7 に進み、また角度が鋭い角（たとえば θ の値が 30 未満）のときはステップ 10 - 8 に進む。

30

【0051】

ステップ 10 - 7 では θ が鋭くない角のときの太め処理を行う。この処理を図 14 に示す図を用いて説明する。対象点を B、対象点の 1 つ前の点を A、対象点の次の点を C とすると、ベクトル A B、およびベクトル B C をなす角を 2 分する方向に対して、輪郭の外側に向けて移動する。その移動量は、ステップ 10 - 1 で求めた x 太め量、y 太め量によってその斜辺となるように決定する。このとき求められた座標値は 1 ストロークの枠に対する座標値であるので、ステップ 10 - 2 で求めたオフセット座標を加えて文字の原点からの座標値とする。

40

【0052】

ステップ 10 - 8 ないしステップ 10 - 14 における θ が鋭い角ではある場合（ θ の値が 30 度未満）の太めの処理の説明を行う。ただし、太め量が正の値であるか負の値であるかによって処理を変える必要がある。そこでステップ 10 - 8 において、太め量が正の値であるか負の値であるかを判定する。太め量が正のときはステップ 10 - 9 へ進み、太め量が負の値である場合にはステップ 10 - 13 へ進む。

【0053】

ステップ 10 - 9 からステップ 10 - 12 で太め量が正の値のときの処理を行う。まずステップ 10 - 9 において、単純に鋭くない角のときと同様に、太め処理のための移動点 B（図 15 参照）を求める。図 15 において、点 a ~ c は元の輪郭、点 A ~ C は単純に太め

50

処理をした輪郭の点である。しかし、鋭い角であるためこの移動点 B は元の点から大きく移動してしまい、ときには文字領域からはみだしてしまう場合がある。これを防ぐために、中間に新たな 2 点を設けて元の点から極端にはなれないよう処理を施す。

【 0 0 5 4 】

ステップ 10 - 10 において、その新たな 2 点を計算によって求める処理を行う。この処理を図 15 を用いて説明する。図 15 において、元の輪郭の 3 点（点 a , 点 b , 点 c ）と単純に太め処理をした対応する輪郭の 3 点（点 A , 点 B , 点 C ）を用いて、ベクトル bB と同方向に距離が太め量の 2 倍の点 D を求める。ベクトル bD に垂直なベクトルでストロークをカットし、単純に太め処理を行った線分 AB および線分 BC と交わる点をそれぞれ点 E および点 F とすると、この 2 つの点 E および点 F を新たな太め処理した点として登録する。このようにして求められた座標点、点 A , 点 E , 点 F , 点 C が太め処理を施した新たな輪郭点として登録される。そしてステップ 10 - 11 において対象とする書体が丸ゴシック体のように角を丸める必要がある書体であるのかどうかを判定する。角を丸める必要がなければ、点 A , 点 E , 点 F , 点 C をそのまま太め処理後の構成点として登録しステップ 10 - 15 に進み、角を丸める必要がある書体であれば、ステップ 10 - 12 へ進み、角を丸める処理を行う。ステップ 10 - 12 においては、図 16 に示すように、線分 EF の中間に新たな点 G をとりそして線分 EG あるは線分 GF と同じ長さの点をそれぞれ線分 EA 上に点 H を、線分 FC 上に点 I をとる。そして線分 HE、線分 EG、線分 GF および線分 FI の中点それぞれ点 J , 点 K , 点 L および点 M とすると、点列 H , J , K , G および点列 G , L , M , I をそれぞれベジェの構成点として表現することが可能となり、直線点列 A , H、ベジェ曲線構成点列 H , J , K , G、ベジェ曲線構成点列 G , L , M , I、直線点列 I , C が角を丸めた新たな太め処理を施した輪郭構成点として登録される。以上ステップ 10 - 9 からステップ 10 - 12 において鋭い角度のとき、かつ太め量が正の場合の処理について説明した。

【 0 0 5 5 】

次にステップ 10 - 13 からステップ 10 - 14 における、太め量が負の場合の処理について図 17 を用いて説明する。図 17 の各点の符号の意味は図 15 と同様である。ステップ 10 - 13 において、単純に鋭くない角のときと同様に太め処理のための移動点 B を求める。しかし、鋭い角であるためこの移動点は元の点から大きく移動してしまい、時にはストロークの線分をなくしてしまう場合がある。よってこれを防ぐために、点列 a , b , c から求められた単純な太め用移動点列 A , B , C の点 B を点 b に近づけるように配置し、ストロークの特徴を残す必要がある。そこでステップ 10 - 14 その近づけるための計算を行う。この計算方法は、ベクトル bB と同方向に距離が太め量の 2 倍の点 D を求め、この点を新たな太め処理の構成点とする。つまり点列 A , D , C が鋭い角度のとき、かつ太め量が負の場合のときの新たな輪郭構成点として登録される。以上のようにしてステップ 10 - 8 からステップ 10 - 14 において、太め処理の対象となる点が鋭い角をなしているときに太め量が正の場合、負の場合のそれぞれについて説明した。

【 0 0 5 6 】

ステップ 10 - 3 からステップ 10 - 14 の処理を全輪郭点に対して処理することになる。10 - 15 において 1 輪郭分のすべての座標点において処理を行っていればステップ 10 - 17 に進み、まだ処理すべき輪郭点が存在する場合には、ステップ 10 - 16 に進み、次の輪郭点へポインタを進めて再度太め処理を行う。

【 0 0 5 7 】

ステップ 10 - 17 においては、1 文字分のすべての輪郭において処理を行ったかどうかを判定し、すべて処理を行っていればステップ 10 - 18 へ進み、まだ処理すべき輪郭が存在するときは、次の輪郭にポインタを進めて再度太め処理を行う。そしてすべての輪郭座標点において太め処理が施されたときに、ステップ 10 - 18 に進む。

【 0 0 5 8 】

太め / 細め処理を行ったことにより、図 18 に示すように、文字の全体の枠が太め処理の場合大きくなり、細め処理の場合は小さくなってしまう。よって元の文字の全体枠に大き

10

20

30

40

50

さを調整する必要がある。ステップ 10 - 18 においてその処理を行う。その調整のための拡大 / 縮小量は、ステップ 10 - 1 で求めた輪郭の太め量（細めの場合は負の値）の 2 倍分だけ太まっている。そこで、元の文字枠の X 幅を B_x 、輪郭の太め量の横の値を F_x とし、Y 幅を B_y 、輪郭の太め量の縦の値を F_y とし、太め処理を行った座標を (x, y) 、調整後の座標を (X, Y) とすると、

【0059】

【数 3】

$$(X, Y) = ((x + F_x) \times B_x / (B_x + F_x \times 2), (y + F_y) \times B_y / (B_y + F_y \times 2))$$

で計算することができる。そしてこの計算を 1 文字分すべての座標点列において処理することによって図 10 に示した太め処理を終了する。 10

【0060】

〔第 2 実施例〕

次に他の実施例について説明する。ここで説明するのはあるウエイトの書体を要求したとき、同一の書体がシステムに 2 つ以上存在したときの例である。このときは要求するウエイトを複数ある同一の書体の中からどのウエイトの書体を基にして処理するかが重要である。従って基準となるウエイトの書体が決定すれば、その後のウエイト変換処理は前記実施例で示したものと同様の処理ができるので、ここでは、基準となるウエイトの選択方法のみについて説明し、それ以外は省略する。

【0061】

20

図 19 は、その詳細な処理の流れを示すフローチャートである。

【0062】

まず、記憶装置にどの書体のどのウエイトが格納されているかを検索する必要がある。そこで、図 19 におけるステップ 19 - 1 において、各書体のヘッダ部に記憶されているヘッダ情報を参照してテーブルにデータ存在情報を格納する。図 20 にそのテーブル及びデータ存在情報を示す。図 20 では、記憶装置に、明朝体のウエイト 3 とウエイト 7、丸ゴシック体のウエイト 5、角ゴシック体のウエイト 6、楷書体のウエイト 4 とウエイト 8 が格納されている例である。ステップ 19 - 2 では、要求されたウエイトの情報がすでに ROM あるいはハードディスク装置等の記憶装置に格納されているか否かを判定する。判定はステップ 19 - 1 で作成されたテーブルを検索していき、要求された書体とウエイトの情報により、データが記憶装置に格納されているか否かが分かる。要求されたウエイトの情報が記憶装置に格納されていれば、ステップ 19 - 3 へ進み、格納されていなければステップ 19 - 4 へ進む。 30

【0063】

ステップ 19 - 3 は要求されたウエイトの情報がすでに記憶装置に格納されていた場合であり、太め / 細め処理を行う必要がないので記憶装置から座標データを読み込み、その座標データから出力サイズに応じて拡大縮小を行い、ビットマップフォントを作成して処理を終了する。

【0064】

ステップ 19 - 4 では、要求されたウエイトの情報が記憶装置にはない場合であり、太め / 細め処理を行って要求されたウエイトの文字を出力する必要がある。そこで、その太め / 細め処理を行う元となるデータを選択する必要がある。一般に細め処理を行う場合に比べて太め処理を行ったほうが品質的に劣化が少なく済むので、要求されたウエイトよりも小さいウエイトの情報が記憶装置に格納されているか否かをステップ 19 - 1 で作成したテーブルを基に判定を行う。そして要求されたウエイトよりも小さいウエイトの情報が記憶装置に格納されていれば、ステップ 19 - 5 に進み、要求されたウエイトの情報が記憶装置に格納されていなければステップ 19 - 6 に進む。例えばウエイト 5 の明朝体を出力したい場合には、ウエイト 3 の明朝体が記憶装置に存在するのでステップ 19 - 5 に進み、ウエイト 3 の角ゴシック体を出力したい場合にはそれより小さいウエイトの角ゴシック体は存在しないのでステップ 19 - 6 に進む。ステップ 19 - 5 においては太め処理の 40 50

元となるウエイトの書体の選択を行い、その太め処理のx方向、y方向のパラメータの設定を行う。ウエイト5の明朝体の出力要求の場合、ウエイト3の明朝体が存在するのでウエイト3とウエイト5の輪郭におけるx方向、y方向の差分を太めパラメータとしてセットする。ステップ19-6においては細め処理の元となるウエイトの書体の選択を行い、その細め処理の輪郭についてx方向、y方向のパラメータの設定を行う。ウエイト3の角ゴシック体の出力要求の場合、ウエイト6の角ゴシック体が存在するのでウエイト3とウエイト6の輪郭におけるx方向、y方向の差分を細めパラメータとしてセットする。

【0065】

以上のように、ステップ19-5およびステップ19-6で太めあるいは細めのパラメータのセットをした後は、前記第1の実施例で説明した通り処理を行い、所望のウエイトの文字を出力する。ここでは対象文字よりウエイトが小さい書体のデータが存在するか否かについて説明したが、対象文字よりもウエイトが大きい書体のデータが存在するか否かについて判定を行っても良い。また、対象文字のウエイトに最も近いウエイトのデータを選択して、そのデータを基にウエイト変換処理を行っても良い。

【0066】

〔第3実施例〕

次に他の実施例について説明する。ここで説明するのは前記第1、第2の実施例で説明した太め処理をグレースケールフォントに対しても適用できることを説明する。ビットマップフォントの各ドットの値が0または1の2値のフォントであるのに対して、グレースケールフォントは各ドットが0～3あるいは0～15あるいは0～255等多値を扱うことが可能な多値フォントである。そしてこのグレースケールフォントを生成するための方法として、一般には図21に示すように n^2 階調のグレースケールフォントを出力するとき、図3のステップ3-4において出力サイズ要求の縦方向、横方向をそれぞれ n 倍して、その出力サイズによってビットマップフォントを作成する。そして図21に示すように縦横 n ビットで分割していきそれぞれの $n \times n$ の分割矩形領域の中に何ビット1が含まれているかによってグレースケールの1ドットに対する多値の値が決定される。第1の実施例をグレースケールフォントに応用した場合、図22に示すフローチャートのようになる。このフローチャートは、前記第1の実施例で説明したビットマップフォント発生あるいは輪郭座標出力のフローとほとんど同様であり、それぞれのステップは図3のステップとほぼ一致しているので、ここではグレースケールフォント発生させることによって処理が異なるステップのところのみを説明する。異なるステップはステップ22-1、ステップ22-4であり、ステップ22-16およびステップ22-17は追加である。ステップ22-1においては、入力パラメータの取り込みであり、入力パラメータにグレースケール情報が追加される。パラメータは、文字コード、書体、ウエイト情報、文字の出力サイズ、グレイのレベル、出力形式、出力器の特性等である。文字コードとしてはJISコード、シフトJISコード、EUCコード、UNIコード等のあらかじめ対象となるシステムがどの文字コード体系によって定められているかによって決まる。また書体としては、明朝体、ゴシック体、丸ゴシック体等のシステムがあらかじめ内蔵しているデータあるいはオプションとして加えられたデータの中から選ばれる。そしてこの時のデータの内容は、グレースケールフォント発生のための特別なデータは格納されておらず、ビットマップフォント発生の際のデータとまったく同様である。ウエイト情報は、前記書体の線の太さ情報であり、ここでは極細、細、中、太、極太等の情報が与えられる。文字の出力サイズとしては、実際にフォントデータを出力する際にどれくらいの大きさで出力するかの情報であり、x方向、y方向のサイズが要求される。グレイのレベルは、グレースケールフォントを発生させる際に何階調でグレースケールフォントを作成するのかの情報である。これは4階調、16階調、256階調等出力器の特性に応じて設定される。出力形式は、所望するフォントの出力データ形式であり、座標データ出力、ビットマップフォント出力、グレースケールフォント出力、1ドットの表現形式等がある。1ドットの表現形式は例えばグレースケールフォントの場合は、1ドットを1バイトで表現するのか、あるいは2ドットや4ドットを1バイトで表現するのかといった表現形式の指定である。出力器の

10

20

30

40

50

特性は、グレイスケールフォントを作成する際にどのようにしてグレイ値を決定すると対象とする出力器に対しても最も最適なグレイスケールフォントが得られるかといった情報である。ステップ 22 - 4 においては、出力サイズとグレイスケールフォントのグレイレベルによって、ステップ 22 - 2 で読み込んだ座標データに対して拡大縮小処理を行う。このとき拡大縮小するための計算式は、要求出力サイズを (A_x, A_y) 、グレイレベルを n 、ステップ 22 - 3 で得られたそれぞれの座標値を (x, y) 、拡大縮小処理後のそれぞれの座標値を (X, Y) 、格納されているデータの文字枠のサイズを (M_x, M_y) とすると、

【0067】

【数 4】

$$(X, Y) = (x \times n \times A_x / M_x, y \times n \times A_y / M_y)$$

となる。

【0068】

ステップ 22 - 16、およびステップ 22 - 17 においては、ステップ 22 - 15 で作成されたビットマップフォントからグレイスケールフォントを作成する処理を行う。まずステップ 22 - 16 においては、入力パラメータとして得られた出力器の特性によって、グレイスケール変換テーブルの選択を行う。グレイスケールフォント変換テーブルは、図 23 に示すように、16 階調の場合 4×4 のマスクであらかじめ出力器の特性を表現するための値を格納しておく。図 23 の (a) の例では出力器の輝度特性がすべて均一になっている場合の例である。図 23 (b) の場合は、輝度がドットの中心部が高く周辺部が低い場合の例である。また図 23 (c) の場合は、輝度がドットの周辺部が高く中心部が低い場合の例である。これらの中から最も出力器の輝度特性に適したテーブルを選択する。そしてステップ 22 - 17 において、ステップ 22 - 16 で選択されたテーブルを元にグレイスケールフォントを作成する。この様子を図 24 を例にとって説明する。図 24 は、ステップ 22 - 15 で求められたビットマップフォントであり、その縦横のサイズはそれぞれグレイレベル n に対して、 n 倍で作成されている。従って、縦横をそれぞれ n で分割して、 $n \times n$ のます目を抽出し、そのます目の 1 つに注目し、そのます目のビットの値とステップ 22 - 16 で求められたテーブルのます目の値とをそれぞれ対応するます目どうしを掛算する。そしてその結果の合計を取ることによって、対象となるドットのグレイスケール値が求められる。図 24 の例では、16 階調のグレイスケールフォントを出力する場合の例であり、 4×4 のます目に対して図 23 (b) のテーブルが選択された場合の例を示した。この操作をすべてのます目に対して行うことによって、図 25 に示すようなグレイスケールフォントが生成される。そしてステップ 22 - 18 で出力形式に応じてグレイスケールフォントを格納して要求側へデータを返す。このとき出力形式が 1 ドットに対して 1 バイトの要求であればそれぞれのます目の値を 1 バイトに詰めて格納する。また隣り合う 2 点を 1 バイトに詰める要求であれば、1 ドットを 4 ビットに詰めてデータを格納し、要求側へ返して処理を終了する。

【0069】

〔第 4 実施例〕

第 1 実施例～第 3 実施例は、字を構成するストロークごとにそのストロークを構成する座標データでベクトルデータが表現されている場合におけるウエイト（太さ）の変換であった。

【0070】

これから説明する実施例は、ベクトルデータが、字の輪郭の座標データで表現されている場合についてのウエイトの変換である。その字形を図 26 に示す。これで分かるように、この輪郭は交差していない。図 26 に示されている字形を図 27 に示されるように、例えば太くする処理について説明する。ここで用いる字の輪郭の座標データの ROM 等に格納されているフォーマットは、例えば図 28 に示されている。これらは、ストロークごとにそのストロークを構成する座標データでベクトルデータを構成している字形を示している図 4～図 6 と比較すると理解が容易である。図 28 において、格納されているデータは、

10

20

30

40

50

文字の輪郭の特徴点を抽出した座標情報であり、それぞれの点に対して直線データ／曲線データ判別フラグ、輪郭開始点／終了点フラグ等の属性情報を持つ。そしてここで扱う曲線データの補間式は2次あるいは2次Bスプライン曲線であったり、2次あるいは3次ベジェ曲線であったりするが、どの補間式を使用しているかはあらかじめ決定されている。また、文字枠を示す座標の最小値は0、最大値は800等で表現されている。

【0071】

このようなデータにおいて、ウエイトの変換処理を説明したのが図29に示したフローチャートである。図29に示したフローチャートは、図3のフローチャートと同じ部分が多いので、相違する処理ステップのみ説明する。ステップ29-1からステップ29-7までは、図3におけるステップ3-1からステップ3-7までと、処理の対象とするデータは相違しているが、ステップ29-3の太め／細め処理を除いて、各ステップの処理は同様であるので説明を省略する。ステップ29-3の処理は後で図34のフローチャートを用いて詳細に説明する。

10

【0072】

ステップ29-8において、直線を発生させる処理を行うが、この処理は、図3のステップ3-8とは、直線の発生仕方は同様のDDAで行っているが、このとき2つの平面に対して行うことが異なっている。その1つの平面は塗りつぶし用XOR平面であり、もう1つの平面は輪郭用OR平面である。輪郭用OR平面は、塗りつぶし用XOR平面で抜けたビットを補うために設けている。これを図30の塗りつぶし用XOR平面を示す図及び図31の輪郭用OR平面を示す図を用いて説明する。図30の塗りつぶし用XOR平面には、1つのY座標に対してX座標を1つのみ打点する。これは、後で説明する塗りつぶしの処理（ステップ29-16）において、各ラインで左側から右側にスキャンしていき、奇数個の目の1から偶数個目の1までの間を1で塗りつぶすため、1つのY座標に対して1つのX座標にしておかないと、うまく塗りつぶすことができないためである。そして、このときの塗りつぶし用XOR平面に打点する方法は、対象とする座標点における塗りつぶし用XOR平面の値と1との排他的論理和（XOR）の論理演算を行い、その結果を塗りつぶし用XOR平面上の対象とする座標点に格納することで行う。そして、もう一つの平面は図31に示した輪郭用OR平面であり、この平面に対して打点するときは、1つのY座標に対して直線にかかっている全てのX座標を1とする。この輪郭用OR平面で、塗りつぶし用XOR平面で抜けたビットを補うことができる。

20

30

【0073】

ステップ29-9及びステップ29-10で曲線の発生させる処理を行うが、直線と同様に2平面に対して打点処理を行うことを除いて図3のステップ3-9及びステップ3-10と同様であるので、説明を省略する。

【0074】

ステップ29-11から29-14までのステップも図3のステップ3-11から3-14までのステップと同様の処理なので説明を省略する。

【0075】

ステップ29-15で塗りつぶし用平面に対して塗りつぶし処理を行う。この処理を図32を用いて説明する。図32の矢印に示すように、塗りつぶし用平面で各スキャンライン単位で左側からスキャンしていき、奇数個目の1から偶数個目の1までの間を1で塗りつぶす処理を行う。この処理を全てのスキャンラインに対して行う。ステップ29-16で、ステップ29-15で求めた塗りつぶしXOR平面と輪郭用OR平面とのORをとって1文字のビットマップデータを完成させる。この様子を示したのが図33である。ステップ29-17において例えば要求側の指定する記憶領域にステップ29-16で求めた1文字分のビットマップデータを返して処理を終了する。

40

【0076】

次に図34のフローチャートを用いて、図29のステップ29-3における太め／細め処理を詳細述べる。この処理は図10で説明した処理と重複する部分が多いので、異なる処理を行っている部分のみ説明する。

50

【 0 0 7 7 】

まず、文字を構成する輪郭において、外側の輪郭と内側の輪郭のそれぞれの場合によって太める／細めるパラメータが異なっているため、輪郭の区別を判別する必要がある。そして、外輪郭に対しては外輪郭太め／細めパラメータ、内輪郭に対しては内輪郭太め／細めパラメータを適用して太め／細め処理を行う。この処理が図 10 の処理と異なっている。この処理は、ステップ 34 - 1 からステップ 34 - 5 で行っている。

【 0 0 7 8 】

さて、輪郭を構成する外側の輪郭点列は左まわりで構成されているとする。そして内側の輪郭点は逆の右まわりで構成されているとする。この場合は輪郭点のまわり方を判定することにより、外側の輪郭であるか、内側の輪郭であるかを判定する。なお、この逆、即ち右まわりの場合を外輪郭、左まわりの場合を内輪郭としてもよい。

10

【 0 0 7 9 】

ステップ 34 - 1 からステップ 34 - 3 でその判定をするための前処理を行っている。この処理を図 35 を参照しながら説明する。まず、ステップ 34 - 1 において、図 35 に示すスタート点 S を取り出し、次にステップ 34 - 2 において、図 35 のスタート点 S の両側の 2 点 A , B を抽出する。点 A はスタート点 S の一つ前の点（その輪郭の最後の点）、点 B はスタート点の一つ後の点である。図 35 において、ベクトル A S 及びベクトル S B のなす角度を 2 分割する方向に前処理用の点を設ける。この点は、外輪郭であるとき即ち左まわりのときはベクトルの方向に向かって右側に、内輪郭であるとき即ち右まわりのときはベクトルの方向に向かって左側に、スタート点からある定められた値離れた点である。これがステップ 34 - 3 の処理である。そして、ステップ 34 - 4 において、この点を基に対象とする輪郭が外輪郭であるのか、内輪郭であるのか判定する。この処理を後で図 40 のフローチャートを用いて詳しく説明する。

20

【 0 0 8 0 】

そして、ステップ 34 - 5 でウエイトに応じて外輪郭の太さ及び内輪郭の太さのパラメータを決定する。太さを決定するパラメータは、外輪郭及び内輪郭に対して x 方向、y 方向それぞれ独立に値を持ち、横線の太め量及び縦線の太め量をそれぞれ独立に管理する。これは、例えば図 36 に示すように、明朝体を太めようとした場合、明朝体の横線はさほど太くせずに済むのに縦線は大きく太める必要があるため、x 方向、y 方向に対し、それぞれ異なる値を設定する必要があるからである。また、図 37 に示すように丸ゴシック体の場合は、横方向及び縦方向はほぼ同じ量だけ太らせている。この考え方は、図 10 のステップ 10 - 1 及び図 11 , 図 12 と同様である。図 38 に示すように、それぞれの書体とウエイトにおける横線と縦線に対して線の中心から外輪郭あるいは内輪郭までの標準値のデータを予めテーブルに持たせ、入力された対象座標データの書体及びウエイトの出力したい座標データのウエイトから外輪郭及び内輪郭の太め量をそれぞれ x 方向、y 方向に対して決定する。

30

【 0 0 8 1 】

ステップ 34 - 6 ~ ステップ 34 - 21 の処理は、対象とする座標データは相違しているが、処理の流れは図 10 のおける処理と同様なので説明を省略する。ただし、これらの処理で、外輪郭、内輪郭を考慮していることが図 10 の処理と異なっている。これは、鋭くない角のときの太め処理を行うステップ 34 - 10 において、外輪郭即ち左まわりのときは右側に移動する（図 39 参照）。なお、外輪郭が右まわりとなっているときは左側に移動する。

40

【 0 0 8 2 】

さて、図 40 のフローチャートを用いて、図 34 のステップ 34 - 4 における外輪郭／内輪郭判定処理について説明する。この処理を図 41 に示すように、輪郭を形成する座標点 A B C D E に対して、ある点 Z からの 2 点に対する角 θ_1 ~ 角 θ_4 を左まわりを正として角度を求めて、角 θ_1 ~ 角 θ_4 の合計の角度が 2π になるとき、ある点 Z は輪郭の内側にあると判定する。また、図 42 に示すように、同様に輪郭を形成する点 A ~ D とある点 Z とでなす角 θ_1 ~ 角 θ_4 の合計の角度が 0 である場合は、ある点 Z は輪郭の

50

外側にあると判定する。この処理を示したのが図40のフローチャートである。ある点Zは、図34のステップ34-1～ステップ34-3で求めているので、図40のフローチャートはその点からの角度の総和を求める処理を行う。ステップ40-1において、まず、角度の総和を0として初期化を行う。ステップ40-2において隣り合った2点を抽出し、ステップ40-3である点Zと抽出した2点とをベクトルで結び、間の角度を求める。その角度は外積を計算することで求めることができる。その求めた値をステップ40-4で角度の総和に加える。ステップ40-5では1輪郭の全ての点を2点ずつ抽出したか判定し、全ての点に対して計算を終了したときはステップ40-6に進み、終了していないときはステップ40-2の戻って再度計算を行う。ステップ40-6で、総和が0か2かを判定し、0であれば外輪郭と判定し外輪郭フラグを立て、2であれば内輪郭であると判定し内輪郭フラグを立てる。あとの処理では、このフラグで外輪郭であるか内輪郭であるかを判別する。

10

【0083】

〔第5実施例〕

第5実施例は、先に説明した第2実施例に対応するもので、交差しない輪郭により座標データが表現されている場合における同一の書体がシステムに2以上存在したときの例である。このときも図19のフローチャートと同様に、細め処理を行ったときに比較して太め処理を行ったときの方が品質の劣化が少ないので、要求されたウエイトより小さい輪郭の座標データを選択する。この処理は図19のフローチャートにおける処理と同様であるので説明を省略する。

20

【0084】

なお、このとき使用する図20に示したテーブルに対応するものを、図43に示す。

【0085】

〔第6実施例〕

第6実施例は、先に説明した第3実施例と対応するもので、交差しない輪郭により座標データが表現されている場合におけるグレースケールフォントに太め処理をした場合を説明するものである。この処理は図44のフローチャートに示されている。この図44のフローチャートは、図22に示されているフローチャートとは、図29のフローチャートと同様に2平面に対して処理をしていることが異っているだけで、同様の処理を行っている。従ってこの処理の説明を省略する。この図44の処理は、図3と図29との相違を理解して、図22を参照すると明白である。

30

【0086】

なお、本発明は複数の機器からなるシステムにおいて、達成されても良く、1つの機器からなる装置において達成されても良い。また、システムあるいは装置にプログラムを供給することにより、本発明が達成される場合にも適用されることは言うまでもない。

【0087】

【発明の効果】

以上説明したように、本発明によれば、入力された太字量または細字量に基づいて、輪郭データの座標値を変換し、調整すべき座標を抽出し、座標値を調整し、調整された輪郭データに基づいて太いまたは細い文字パターンを発生させるので、少ない文字データから肉厚量の異なる高品質な太字または細字文字を発生することができる。

40

【0088】

本発明によれば、入力された太字量または細字量に基づいて、変換するいずれかの文字データ候補を決定し、該決定された文字データに対応する輪郭データの座標値を変換し、該座標変換された輪郭データに基づいて、太いまたは細い文字パターンを発生しているので、文字データ資源を生かして最良の太字または細字の文字データを発生させることができる。

【0089】

本発明によれば、決定は、入力された太字量または細字量に近い肉厚量の文字データを変換する文字データ候補として、より厳密な太字または細字の文字データを発生させること

50

ができる。

【0090】

また、本発明によれば、変換は、対象座標に隣接する2点の座標値を参照して1つの変換座標を決定することにより行うので、バランスのとれた太いまたは細字の文字パターンを発生することができる。

【0091】

本発明によれば、変換は、抽出した各輪郭データを入力された太字量または細字量をx方向およびy方向に対して独立して可変することにより行うので、書体毎の特徴を生かした太字または細字の文字パターンを発生することができる。

【0092】

本発明によれば、判定は、対象座標と隣接する2点の座標値を参照して3点のなす角度および入力された値が太字量あるいは細字量であるかおよび対象書体によって決定することにより行うので、良好な品質の太字または細字の文字パターンを発生することができる。

【0093】

本発明によれば、座標調整は、太字量が指定されたときに対象点に新たな点を挿入することにより行うので、良好な品質の太字の文字パターンを発生することができる。

【0094】

本発明によれば、座標調整は、太字量が指定されたときに対象点に新たな点を挿入した後、角を丸めることによって行うので、書体毎の特徴を生かした良好な太字の文字パターンを発生することができる。

【0095】

本発明によれば、座標調整は、細字量が指定されたときに対象点を移動させることにより行うので、良好な品質の細字の文字パターンを発生することができる。

【0096】

本発明によれば、データの発生は、変換された輪郭データに基づいてビットマップフォント、輪郭データ、グレイスケールフォントのいずれかを出力して、種々の出力手段に適切なデータ形式で太字または細字の文字データを供給することができる。

【0097】

前記輪郭データは、交差するストロークによる複数の輪郭データで構成されていてもよいし、交差しない複数の文字の輪郭データで構成されていてもよい。

【0098】

従って、少ないメモリ容量で、様々な肉厚量の各書体の文字データを良好に発生できるという効果を奏する。

【図面の簡単な説明】

【図1】本発明の内部構成を示すブロック図である。

【図2】本発明の他の内部構成を示すブロック図である。

【図3】本発明のストロークによるアウトラインフォントに対する処理全体の流れを示すフローチャートである。

【図4】本発明に用いるストロークによるアウトラインフォントを説明する図である。

【図5】ストロークによる輪郭の座標データのフォーマットを示す図である。

【図6】本発明によって変換されたアウトラインフォントを示す図である。

【図7】座標データから塗りつぶしテーブルを作成する様子を示す図である。

【図8】3次ベジェ曲線を分解する様子を示す図である。

【図9】塗りつぶしテーブルからビットマップフォントを作成する様子を示す図である。

【図10】太め処理の詳細を示すフローチャートである。

【図11】明朝体の太め処理結果を示す図である。

【図12】丸ゴシック体の太め処理を示す図である。

【図13】太めパラメータを決定するために用いるテーブルを示す図である。

【図14】太め処理を行っている様子を示す図である。

【図15】太め量が正で鋭い角度のとき点を分割の様子を示す図である。

10

20

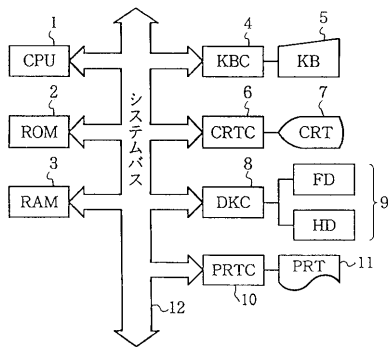
30

40

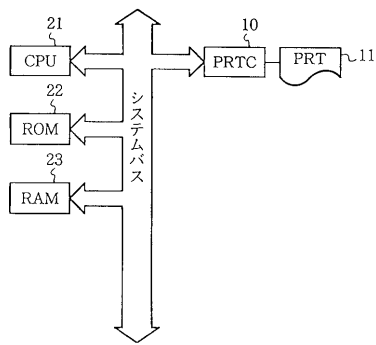
50

- 【図 1 6】分割した点を丸接合する様子を示す図である。
- 【図 1 7】太め量が負のときの鋭い角度の対応を示す図である。
- 【図 1 8】太め処理によって不都合が生じる様子を示す図である。
- 【図 1 9】本発明の他の実施例の全体の流れを示すフローチャートである。
- 【図 2 0】太めパラメータを決定するために用いるテーブルを示す図である。
- 【図 2 1】グレイスケールフォントにするために作成するビットマップを示す図である。
- 【図 2 2】本発明の他の実施例の全体の流れを示すフローチャートである。
- 【図 2 3】グレイスケールフォントに変換するための基本概念図である。
- 【図 2 4】グレイスケールフォントに変換するための様子を示す図である。
- 【図 2 5】グレイスケールフォントに変換された様子を示す図である。 10
- 【図 2 6】本発明に用いる交差しない輪郭によるアウトラインフォントを説明する図である。
- 【図 2 7】本発明によって変換された交差しない輪郭によるアウトラインフォントを示す図である。
- 【図 2 8】交差しない輪郭による座標データのフォーマットを示す図である。
- 【図 2 9】本発明の交差しない輪郭によるアウトラインフォントに対する処理全体の流れを示すフローチャートである。
- 【図 3 0】座標データから塗りつぶし用 X O R 平面を示す図である。
- 【図 3 1】座標データから輪郭用 O R 平面を示す図である。
- 【図 3 2】塗りつぶし用平面で塗りつぶしを行う様子を示す図である。 20
- 【図 3 3】ビットマップフォントを発生する様子を示す図である。
- 【図 3 4】交差しない輪郭による輪郭データの太め処理の詳細を示すフローチャートである。
- 【図 3 5】スタート点の仮移動を説明する図である。
- 【図 3 6】明朝体の太め処理結果を示す図である。
- 【図 3 7】丸ゴシック体の処理結果を示す図である。
- 【図 3 8】太めパラメータを決定するために用いるテーブルを示す図である。
- 【図 3 9】太め処理を行っている様子を示す図である。
- 【図 4 0】内輪郭 / 外輪郭を判定する処理を示すフローチャートである。
- 【図 4 1】内輪郭を判定を説明する図である。 30
- 【図 4 2】外輪郭を判定を説明する図である。
- 【図 4 3】太め又目ータを決定するためのテーブルを示す図である。
- 【図 4 4】グレイスケールフォントを生成するためのフローチャートである。
- 【符号の説明】
- 1 CPU (中央処理装置)
 - 2 ROM (読み出し専用メモリ)
 - 3 RAM (ランダムアクセスメモリ)
 - 4 KBC (キーボード制御部)
 - 5 KB (キーボード)
 - 6 CRTC (ディスプレイ制御部)
 - 7 CRT (ディスプレイ装置)
 - 8 DKC (ディスク制御部)
 - 9 外部記憶装置
 - 10 PRTC (プリンタ制御部)
 - 11 PRT (プリンタ装置)
 - 12 システムバス
- 40

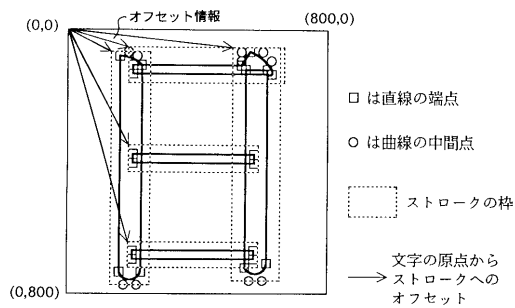
【図 1】



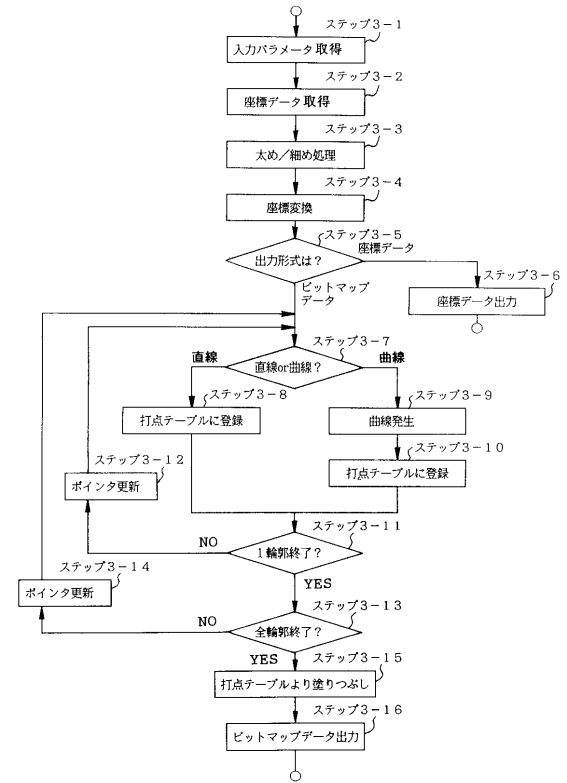
【図 2】



【図 4】



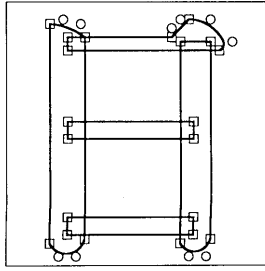
【図 3】



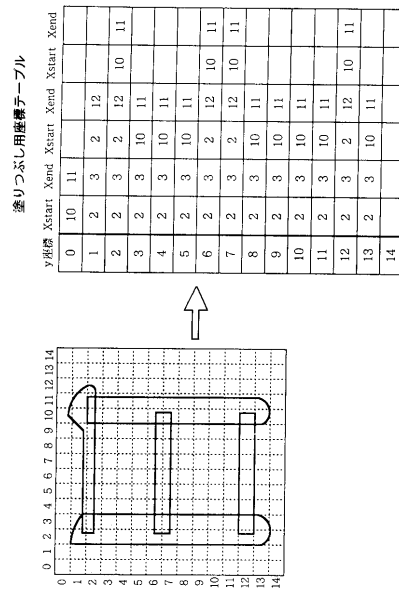
【図 5】

総輪郭数
第1輪郭終了点番号
第1輪郭へのオフセットX
第1輪郭へのオフセットY
第2輪郭終了点番号
第2輪郭へのオフセットX
第2輪郭へのオフセットY
...
第n輪郭終了点番号
第n輪郭へのオフセットX
第n輪郭へのオフセットY
X0座標値
Y0座標値
0番目の点属性
X1座標値
Y1座標値
1番目の点属性
X2座標値
Y2座標値
2番目の点属性
...
Xm座標値
Ym座標値
m番目の点属性

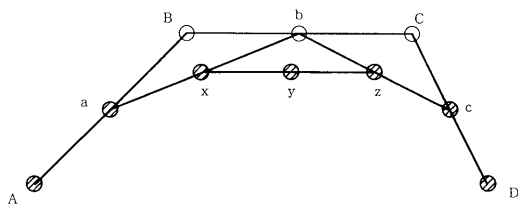
【図 6】



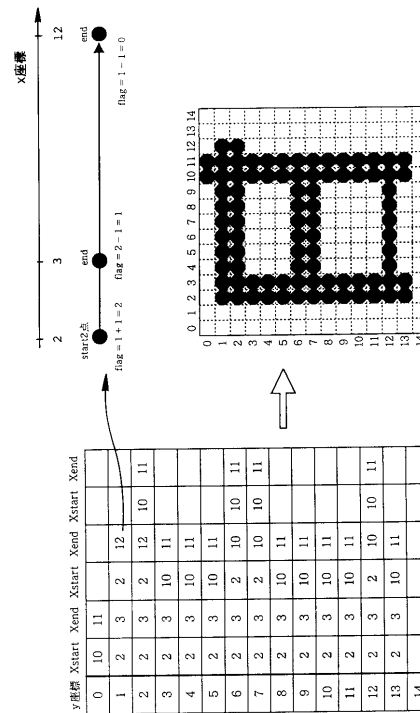
【図 7】



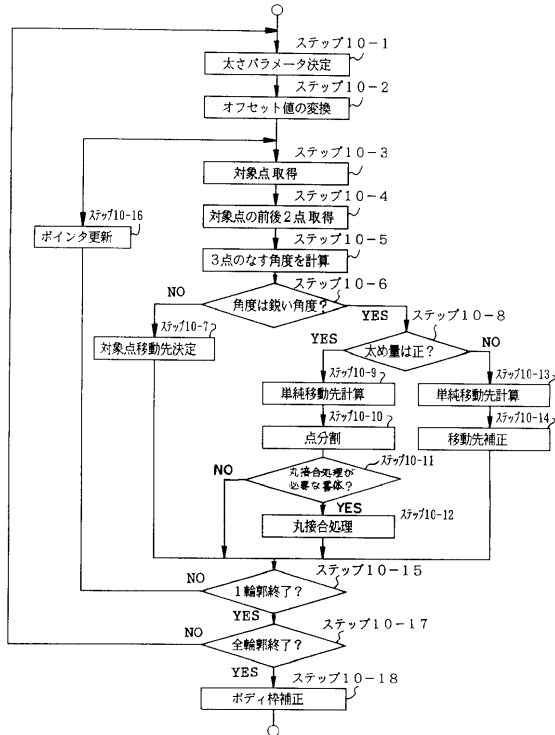
【図 8】



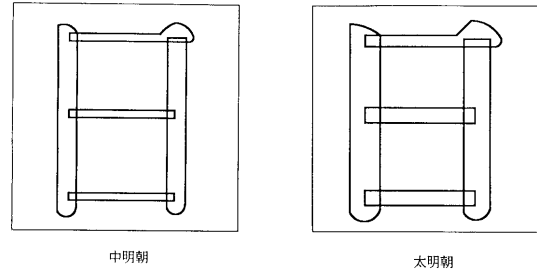
【図 9】



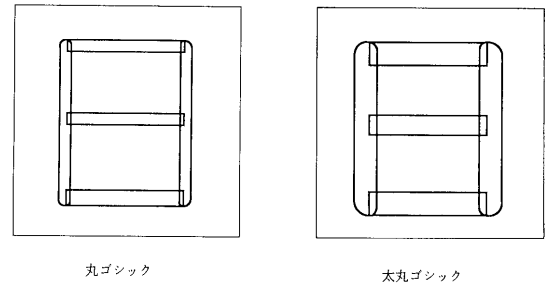
【図 10】



【図 11】



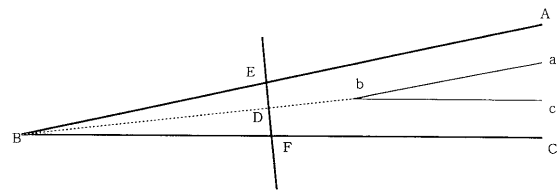
【図 12】



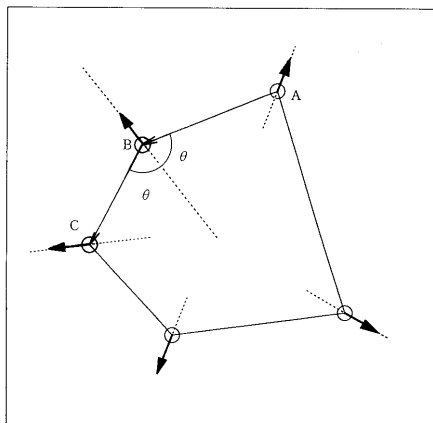
【図 13】

書体	明朝体		丸ゴシック体	
x/y	x	y	x	y
極細	6	10	10	10
細	9	15	15	15
中	12	20	20	20
太	15	25	25	25
極太	18	30	30	30

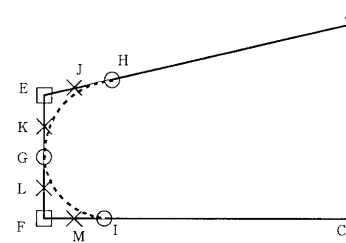
【図 15】



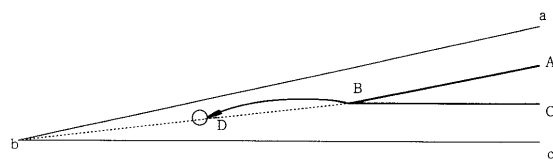
【図 14】



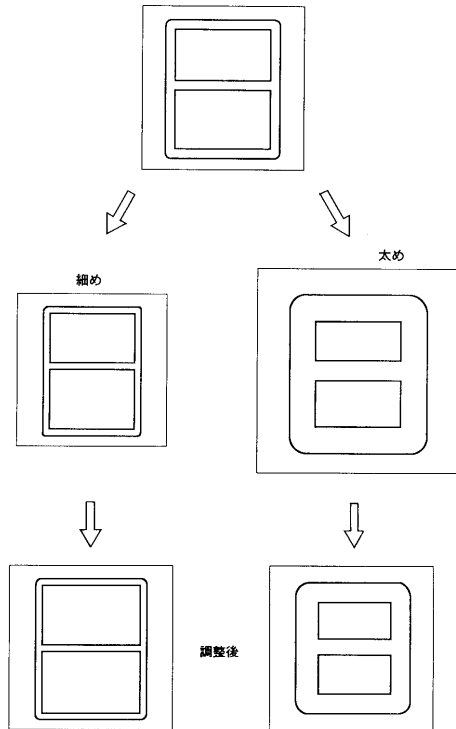
【図 16】



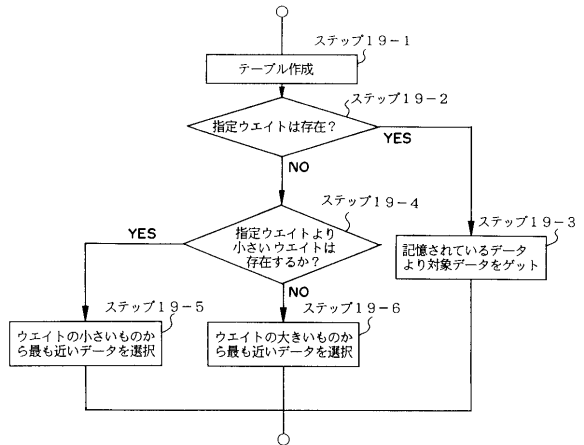
【図 17】



【図 18】



【図 19】

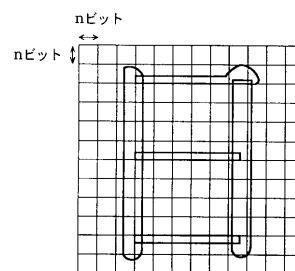


【図 20】

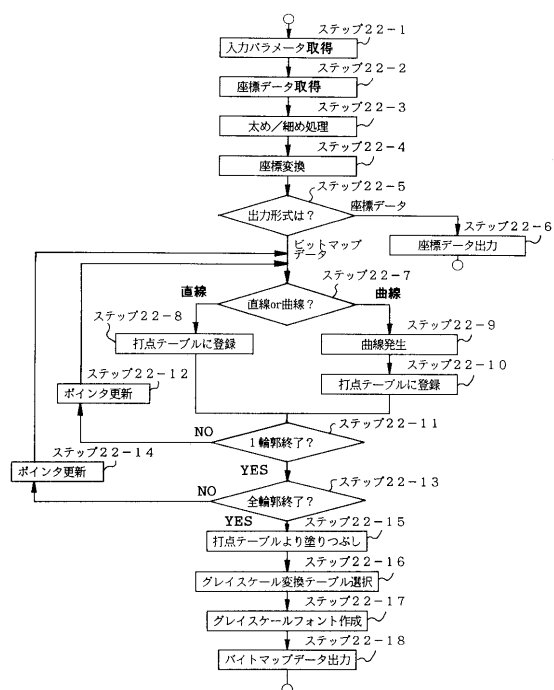
書体	明朝体			丸ゴシック体		
x/y	x	y	存在	x	y	存在
ウエイト 極細 1	3	5		5	5	
細 2	5	10		10	10	
中細 3	7	15	○	15	15	
中 4	10	20		20	20	
中太 5	12	25		25	25	○
太 6	14	30		30	30	
極太 7	16	35	○	35	35	

書体	角ゴシック体			楷書体		
x/y	x	y	存在	x	y	存在
ウエイト 極細 1	5	5		3	5	
細 2	10	10		6	10	
中細 3	15	15		9	15	
中 4	20	20		12	20	○
中太 5	25	25		15	25	
太 6	30	30	○	18	30	
極太 7	35	35		21	35	○

【図 21】



【 ㊦ 2 2 】



【 図 2 3 】

1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1

(a)

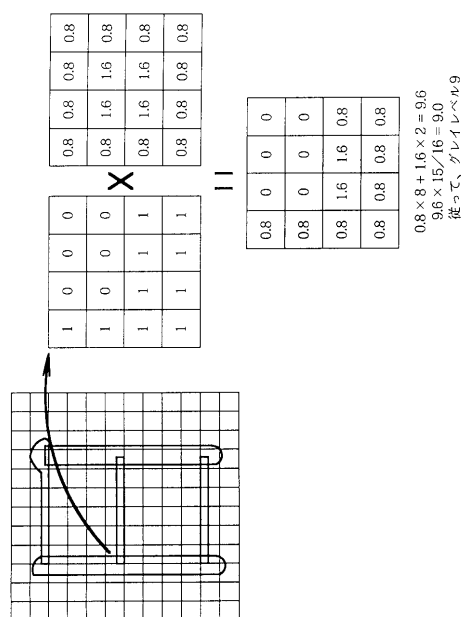
0.8	0.8	0.8	0.8
0.8	1.6	1.6	0.8
0.8	1.6	1.6	0.8
0.8	0.8	0.8	0.8

(b)

1.1	1.1	1.1	1.1
1.1	0.7	0.7	1.1
1.1	0.7	0.7	1.1
1.1	1.1	1.1	1.1

(c)

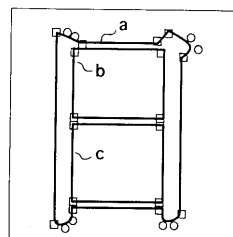
【 図 2 4 】



【 図 2 5 】

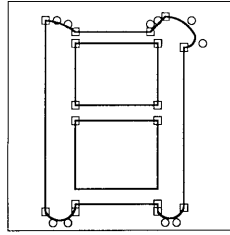
0	0	0	0	0	0	0	0	0	0
0	0	10	9	8	8	8	9	13	5
0	0	12	3	0	0	0	0	12	4
0	0	12	3	0	0	0	0	12	3
0	0	12	3	0	0	0	0	12	3
0	0	12	9	8	8	8	8	13	3
0	0	12	3	0	0	0	0	12	3
0	0	12	3	0	0	0	0	12	3
0	0	12	3	0	0	0	0	12	3
0	0	12	3	0	0	0	0	12	3
0	0	12	9	8	8	8	8	12	2
0	0	2	1	0	0	0	0	0	0

【 图 2 6 】



□ は直線の端点
○ は曲線の間接点

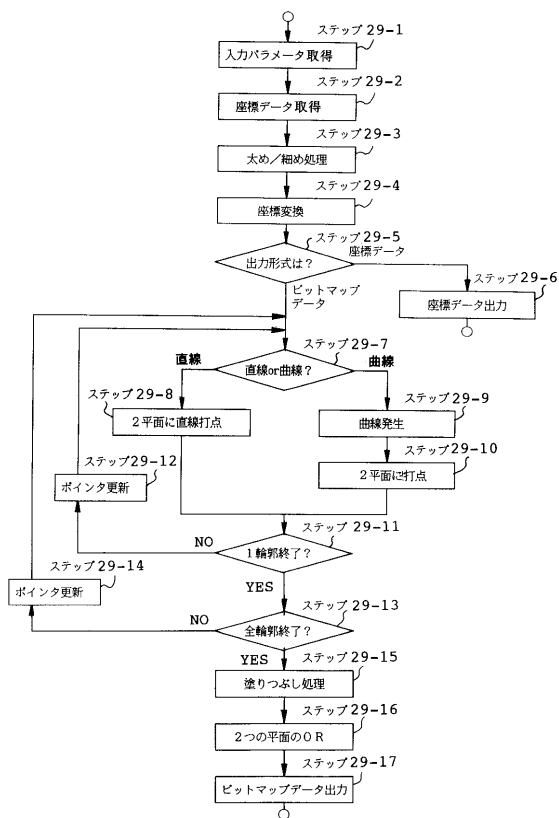
【図 27】



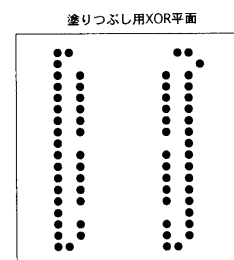
【図 28】

総輪郭数
第1輪郭終了点番号
第2輪郭終了点番号
...
第n輪郭終了点番号
X0座標値
Y0座標値
0番目の点属性
X1座標値
Y1座標値
1番目の点属性
X2座標値
Y2座標値
2番目の点属性
...
Xm座標値
Ym座標値
m番目の点属性

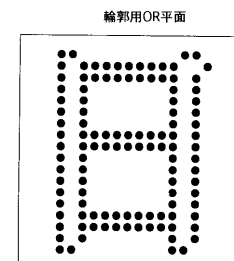
【図 29】



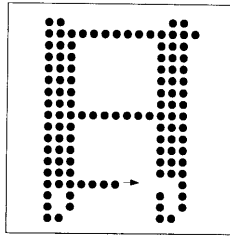
【図 30】



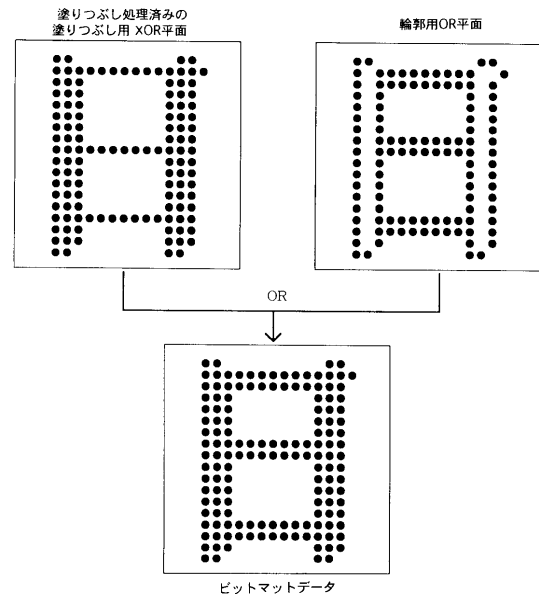
【図 31】



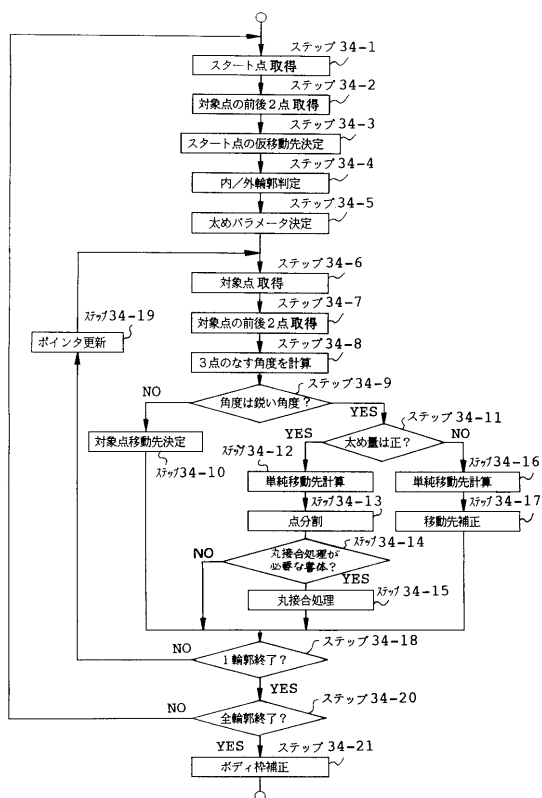
【図 3 2】



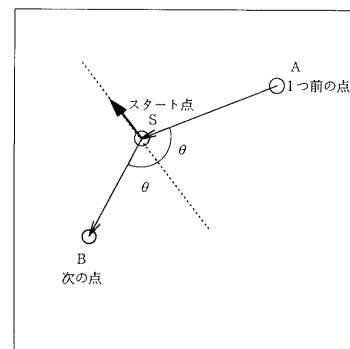
【図 3 3】



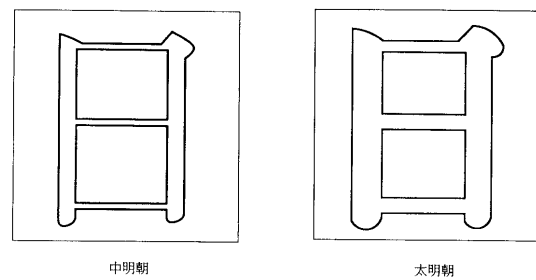
【図 3 4】



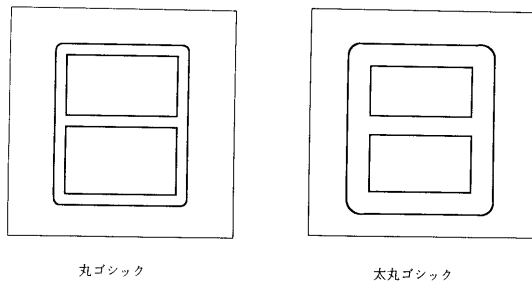
【図 3 5】



【図 3 6】



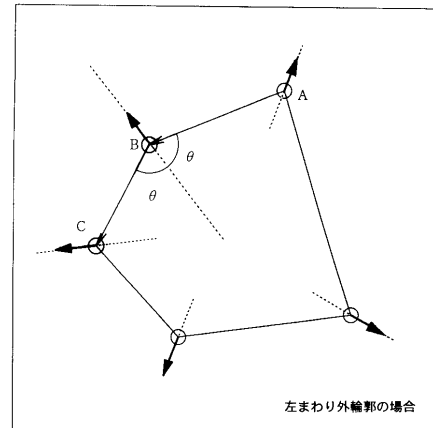
【図 37】



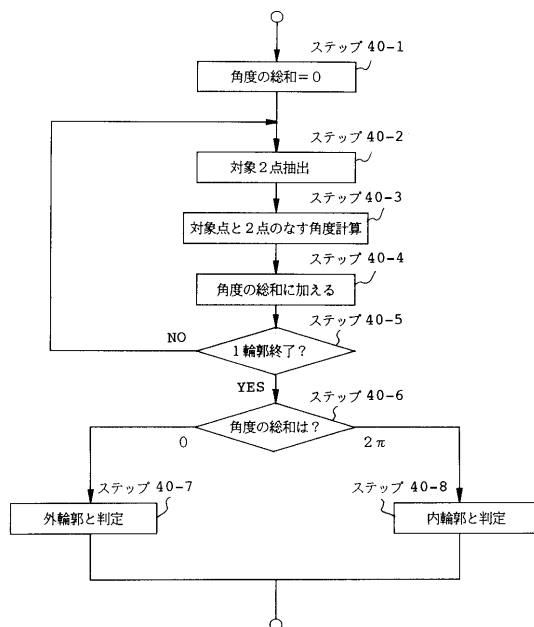
【図 38】

書体	明朝体				丸ゴシック体			
	外輪郭		内輪郭		外輪郭		内輪郭	
x/y	x	y	x	y	x	y	x	y
極細	6	10	5	8	10	10	8	8
細	9	15	7	13	15	15	13	13
中	12	20	10	18	20	20	18	18
太	15	25	13	23	25	25	23	23
極太	18	30	16	28	30	30	28	28

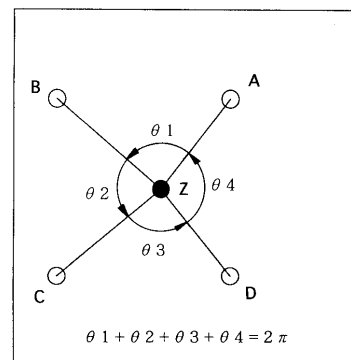
【図 39】



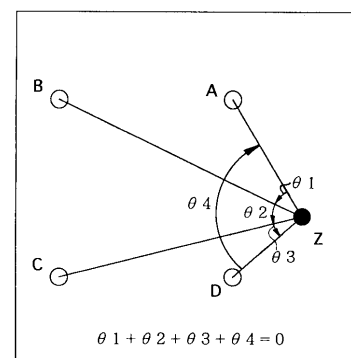
【図 40】



【図 41】



【図 42】

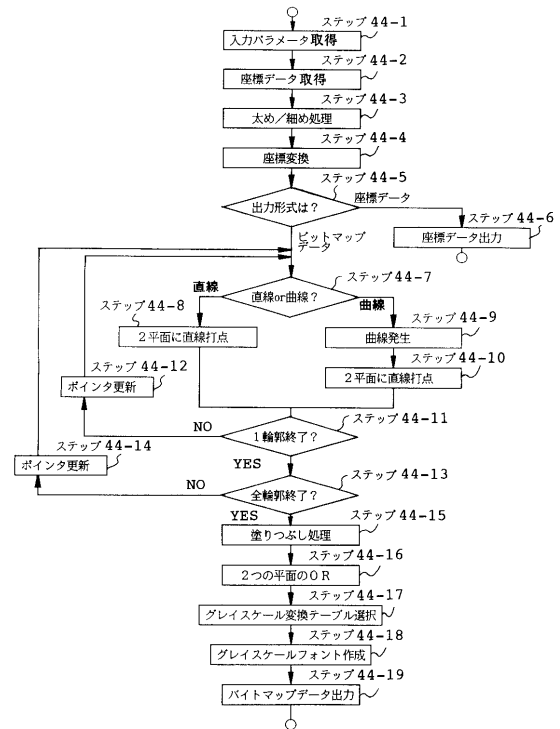


【図 4 3】

書体	明朝体				丸ゴシック体			
内/外/存在	外輪郭		内輪郭		外輪郭		内輪郭	
x/y	x	y	x	y	x	y	x	y
ウェイト極細 1	3	5	2	5	5	5	3	3
細 2	5	10	4	8	10	10	8	8
中細 3	7	15	6	13	15	15	13	13
中 4	10	20	9	18	20	20	18	18
中太 5	12	25	11	23	25	25	23	23
太 6	14	30	13	28	30	30	28	28
極太 7	16	35	15	33	35	35	33	33

書体	角ゴシック体				楷書体			
内/外/存在	外輪郭		内輪郭		外輪郭		内輪郭	
x/y	x	y	x	y	x	y	x	y
ウェイト極細 1	5	5	3	3	3	5	3	5
細 2	10	10	8	8	6	10	5	8
中細 3	15	15	13	13	9	15	7	13
中 4	20	20	18	18	12	20	10	18
中太 5	25	25	23	23	15	25	13	23
太 6	30	30	28	28	18	30	16	28
極太 7	35	35	33	33	21	35	19	33

【図 4 4】



フロントページの続き

- (56)参考文献 特開平06-138865(JP,A)
特開平02-205892(JP,A)
特開平05-119758(JP,A)
特開平05-173533(JP,A)
特開平05-143046(JP,A)

(58)調査した分野(Int.Cl.⁷, DB名)

G09G 5/00 - 5/42

G06F 3/14

17/21

B41J 2/485