(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第3563891号
(P3563891)
(45)発行日　平成16年9月8日(2004.9.8)　(24)登録日　平成16年6月11日(2004.6.11)

(51)Int.Cl.[7]　F I
G09G 5/24　G09G 5/24 620L
G09G 5/24 630H
G09G 5/24 630S

請求項の数 6　(全 20 頁)

(21)出願番号　特願平8-282461
(22)出願日　平成8年10月24日(1996.10.24)
(65)公開番号　特開平10-124026
(43)公開日　平成10年5月15日(1998.5.15)
審査請求日　平成15年5月6日(2003.5.6)

(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　100076428
弁理士　大塚　康徳
(74)代理人　100093908
弁理士　松本　研一
(72)発明者　師岡　秀和
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　小川　浩史

最終頁に続く

(54)【発明の名称】文字生成方法及び装置

(57)【特許請求の範囲】
【請求項１】
ボディとヘッドまたはテイルとを接続してストロークを生成し、生成したストロークを組み合わせて文字を生成する文字生成装置であって、
ボディが水平または垂直ボディ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ボディが水平または垂直ボディでないと判別された場合、ヘッドまたはテイルの接続部の両端点を結んだベクトルとボディの接続部の両端点を結んだベクトルとの角度を内積を用いて計算し、計算結果に基づきヘッドまたはテイルを回転し、回転したヘッドまたはテイルとボディとを接続し、
前記判別手段によりボディが水平または垂直ボディであると判別された場合、ヘッドまたはテイルの接続部の両端点を結んだベクトルとボディの接続部の両端点を結んだベクトルとの角度を内積を用いて計算することなく９０度単位の回転処理によりヘッドまたはテイルを回転し、回転したヘッドまたはテイルとボディとを接続する接続手段と
を有することを特徴とする文字生成装置。
【請求項２】
生成された文字を印刷する印刷手段を有することを特徴とする請求項１記載の文字生成装置。
【請求項３】
前記判別手段におけるボディが水平または垂直ボディであるか否かの判定は、ストロークのスケルトンデータのスタート座標とエンド座標とを用いて判別することを特徴とする請

求項１に記載の文字生成装置。
【請求項４】
ボディとヘッドまたはテイルとを接続してストロークを生成し、生成したストロークを組み合わせて文字を生成する文字生成方法であって、
ボディが水平または垂直ボディ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ボディが水平または垂直ボディでないと判別された場合、ヘッドまたはテイルの接続部の両端点を結んだベクトルとボディの接続部の両端点を結んだベクトルとの角度を内積を用いて計算し、計算結果に基づきヘッドまたはテイルを回転し、回転したヘッドまたはテイルとボディとを接続し、
前記判別ステップによりボディが水平または垂直ボディであると判別された場合、ヘッドまたはテイルの接続部の両端点を結んだベクトルとボディの接続部の両端点を結んだベクトルとの角度を内積を用いて計算することなく９０度単位の回転処理によりヘッドまたはテイルを回転し、回転したヘッドまたはテイルとボディとを接続する接続ステップと
を有することを特徴とする文字生成方法。
【請求項５】
生成された文字を印刷する印刷ステップを有することを特徴とする請求項４記載の文字生成方法。
【請求項６】
前記判別ステップにおけるボディが水平または垂直ボディであるか否かの形状の判定は、ストロークのスケルトンデータのスタート座標とエンド座標とを用いて判別することを特徴とする請求項４に記載の文字生成方法。
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】
本発明は、文字パターンを、文字の要素を組合わせて再構成する文字発生方法及び装置に関する。
【０００２】
【従来の技術】
従来、フォントデータ容量を削減するため、フォントを文字構成要素（以後ストローク）を最小単位とする基本文字構成要素に分割して記憶し、それを組み合わせて文字を発生するストローク合成フォント方式が用いられている。この方式は、文字を発生する過程でそれを構成するストロークを構築し、必要に応じて変倍等を行い、各ストロークを、文字を構成する要素として配置して最終的に文字が得られる。
【０００３】
【発明が解決しようとする課題】
しかしながら、ストローク合成フォント方式においては、文字発生に要する全処理時間のおよそ３０％をストローク構築処理が占めており、ストローク合成フォント方式は、文字の全輪郭型データをもつアウトラインフォント処理と比較し、ストロークを構築する処理が必要な分、文字パターンの展開時間の増大を招いていた。
【０００４】
本発明は上記従来例に鑑みてなされたもので、ストローク構築処理時間を圧縮して文字発生に要する時間を短縮できる文字処理方法及び装置を提供することを目的とする。
【０００５】
【課題を解決するための手段】
上記の目的を達成するために、請求項１に記載の発明は、ボディとヘッドまたはテイルとを接続してストロークを生成し、生成したストロークを組み合わせて文字を生成する文字生成装置であって、ボディが水平または垂直ボディ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ボディが水平または垂直ボディでないと判別された場合、ヘッドまたはテイルの接続部の両端点を結んだベクトルとボディの接続部の両端点を結んだベクトルとの角度を内積を用いて計算し、計算結果に基づきヘッドまたはテイルを回転し、回転し

たヘッドまたはテイルとボディとを接続し、前記判別手段によりボディが水平または垂直ボディであると判別された場合、ヘッドまたはテイルの接続部の両端点を結んだベクトルとボディの接続部の両端点を結んだベクトルとの角度を内積を用いて計算することなく９０度単位の回転処理によりヘッドまたはテイルを回転し、回転したヘッドまたはテイルとボディとを接続する接続手段とを有する。

また、請求項４に係る発明は、ボディとヘッドまたはテイルとを接続してストロークを生成し、生成したストロークを組み合わせて文字を生成する文字生成方法であって、ボディが水平または垂直ボディ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ボディが水平または垂直ボディでないと判別された場合、ヘッドまたはテイルの接続部の両端点を結んだベクトルとボディの接続部の両端点を結んだベクトルとの角度を内積を用いて計算し、計算結果に基づきヘッドまたはテイルを回転し、回転したヘッドまたはテイルとボディとを接続し、前記判別ステップによりボディが水平または垂直ボディであると判別された場合、ヘッドまたはテイルの接続部の両端点を結んだベクトルとボディの接続部の両端点を結んだベクトルとの角度を内積を用いて計算することなく９０度単位の回転処理によりヘッドまたはテイルを回転し、回転したヘッドまたはテイルとボディとを接続する接続ステップとを有する。

【０００６】

【発明の実施の形態】

［第１の実施の形態］

図１は本発明を適用した情報処理システムの基本的な構成を示す。本システムは日本語ワードプロセッサであっても良いし、ワークステーションあるいはパーソナルコンピュータ等のシステムであっても良い。図１において、１０１はＣＰＵ、即ち中央処理装置であり、この装置全体の制御及び演算処理等を行う。

【０００７】

１０２はＲＯＭ即ち読み出し専用メモリであり、システム起動プログラム及び文字変換用の文字パターン・データ等を記憶する。

【０００８】

１０３はＲＡＭ即ちランダムアクセスメモリであり、ＣＰＵ１０１により実行される後述するプログラムやその実行時に使用されるデータ、ＣＰＵ１０１の演算結果等を一時記憶する。

【０００９】

１０４はＫＢＣ即ちキーボード制御部であり、１０５のＫＢ即ちキーボードよりキー入力データ（文字コードや制御コード）を受け取りＣＰＵ１０１へ伝達する。

【００１０】

１０６はＣＲＴＣ即ちディスプレイ制御部であり、ＲＡＭ１０３に格納された表示情報、即ち、文字コードから変換されたビットマップイメージを読み出し、ＣＲＴ１０７に転送する。

【００１１】

１０７はＣＲＴ即ちディスプレイ装置であり、ＣＲＴＣ１０６よりビットマップイメージを受け取り表示画面に表示する。

【００１２】

１０８はＤＫＣ即ちディスク制御部であり、外部記憶装置１０９のデータ伝送等の制御を行う。

【００１３】

１０９はＦＤ即ちフレキシブルディスク装置、あるいはＨＤ即ちハードディスク装置、あるいはＣＤ－ＲＯＭ等の外部記憶装置である。外部記憶装置１０９にはプログラム及びデータを記憶させておき、ＣＰＵ１０１は必要に応じて記憶データを参照したり、またはＲＡＭ１０３にロードする。

【００１４】

１１０はＰＲＴＣ即ちプリンタ制御部であり、ＰＲＴ即ちプリンタ装置１１１の動作制御

を行う。
【００１５】
１１２はシステムバスであり、上述の構成要素間のデータ転送を行う。
【００１６】
図１２は、図１のシステム構成により本発明に係る文字生成処理を行う際のＲＡＭ１０３のメモリ構成を示した図である。本装置は、基本Ｉ／Ｏプログラム８１、ウインドウシステム等のオペレーティングシステム８２、文字パターンを発生する文字パターン発生プログラム８３あるいはそれを利用するアプリケーションがＣＰＵ１０１により実行されることにより動作する。基本Ｉ／Ｏプログラム８１は図１のＲＯＭ１０２に格納されている。オペレーティングシステム８２はハードディスク（ＨＤ）１０９－２に格納されている。そして電源がオンされた時に、基本Ｉ／Ｏプログラム８１中のＩＰＬ（イニシャライズプログラムローディング）機能によりＨＤ１０９からオペレーティングシステム８２がＲＡＭ１０３に読み込まれ、その動作が開始される。
【００１７】
本実施形態におけるプログラム及び関連データは、図１のＦＤ１０９－１あるいはＣＤ－ＲＯＭ１０９－３により供給される。図１３にこのＦＤ１０９－１あるいはＣＤ－ＲＯＭ１０９－３に格納されたファイルの構成を示す。そしてＦＤ１０９－１あるいはＣＤ－ＲＯＭ１０９－３から読出したプログラムファイル及び関連データファイルを一旦ＨＤ１０９－２へインストールすることにより、ＨＤ１０９－２からＲＡＭ１０３にロードし、ＣＰＵ１０１により実行することが可能となる。あるいは、ＨＤへインストールせずに、ＦＤあるいはＣＤ－ＲＯＭからＲＡＭ１０３に直接ロードして実行することも可能である。
【００１８】
図５は本発明にかかる文字処理装置を適用可能な第２のシステムの制御構成を示すブロック図であり、図１と同一のものには同一の符号を付してある。図１のこうせいに代えて、図５の構成の装置を用いてもよい。ただし、図５の装置にはファイルを記憶する画伊具記憶装置が備わっていないため、プログラムやフォントデータ等はＲＯＭ２１２に格納されることになる。この場合には、メモリマップはデータ類を除いて図１２と同様になる。
【００１９】
尚、プリンタ２１１１はレーザビームプリンタであっても良いし、インクジェットプリンタあるいは熱転写等のプリンタであっても良い。図５において、２１１はＣＰＵ、即ち中央処理装置であり、この装置全体の制御及び演算処理等を行うものである。
【００２０】
２１２はＲＯＭ即ち読み出し専用メモリであり、システム起動プログラム及び文字パターン・データ等の記憶領域である。２１３はＲＡＭ即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及びデータがロードされ実行される領域である。
【００２１】
２１１０はＰＲＴＣ即ちプリンタ制御部であり、２１１１はＰＲＴ即ちプリンタ装置である。２１１２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。
【００２２】
以上の構成からなる本実施例の動作を図６～９のフローチャートを用いて説明する。
【００２３】
図２は、ストローク合成フォント方式による文字生成のし方の例を示す。その文字の骨格となる横棒、縦棒といった基本文字構成要素（ストロークボディ）２０１と、その始点、終点部に局部輪郭（ストロークヘッド、テイル）２０２を組み合わせることによりストロークを形成し、ストロークを組み合わせることにより文字を構成するものである。基本文字構成要素のみでストロークを構成することも可能である。ストロークの形状を規定するデータは、図３に示すようにストロークの芯線を直線により表し、輪郭をそこからの距離で表現する「芯線（スケルトン）＋太さ」型であり、各ストロークは、幅情報（以後ウェ

イト情報）を持つ。
【００２４】
また、各ストロークには、ストロークＩＤが割り振られており、各文字コードに１対１に対応したストローク組み合わせ情報には、その文字を合成するためのストロークＩＤ及びストローク配置情報が記憶されており、それを用いてストロークの合成を行う。ストローク組み合わせ情報は、書体、文字コード、文字サイズ、ウェイト情報などに応じて組合わせるストロークを決定し、予め作成しておく。
【００２５】
図４は、ストローク組み合わせ情報の構造の例を示した図である。ＳＴＲＯＫＥ＿ＣＯＵＮＴはフォントを構成するストローク数を示し、ストロークＩＤはその文字に用いられるストロークの識別子を示す。図４の例では、図４（ａ）の文字は、ストロークＩＤが１，２，３の３つのストロークから構成されることがわかる。ストローク配置座標は、フォントデザイン座標系（本実施形態では、８００×８００とする）での各ストロークの芯線の両端点の座標ＳＴＲＯＫＥ＿ＳＴ，ＳＴＲＯＫＥ＿ＥＮＤを示し、ストロークＷｅｉｇｈｔはストロークの幅情報を示す。ストロークＩＤからＲＯＭ１０２あるいは外部記憶１０９格納されたストロークデータを検索することにより、各ストロークのストロークデザイン座標系での大きさＤ＿ＳＴＲＯＫＥ＿ＳＩＺＥ、ウェイト情報Ｄ＿ＳＴＲＯＫＥ＿Ｗｅｉｇｈｔ、制御点データＤ＿ＳＴＲＯＫＥ＿ＰＯＩＮＴ［　］、ストロークに接続する局部輪郭のヘッドデータＨ＿ＩＤ、テイルデータＴ＿ＩＤを読み出す。なお、局部輪郭のＩＤが示すデータは、ストロークデザイン座標系での、局部輪郭の制御点データ列である。また、局部輪郭のヘッドデータＨ＿ＩＤ及びテイルデータＴ＿ＩＤは、ストローク組み合わせ情報を作成する時点でストロークデータから読み出し、ストローク組み合わせ情報の一部として付加しておく。
【００２６】
図６～９は、本実施例に係るシステムの基本動作手順を示すフローチャートである。このフローチャートは、文字パターン発生プログラムの手順を示しており、ＣＰＵ１０１は、これらフローチャートの手順で処理を実行し、文字を生成する。
【００２７】
図６において、ステップＳ６－１は、本システムに搭載されている複数のフォントファイルよりストローク組み合わせ情報を読み出しＲＡＭ１０３上にセットする。
【００２８】
ステップＳ６－２では、書体、文字コード、文字サイズ、ウェイト情報より、読み出すべきストローク組み合わせ情報を決定する。
【００２９】
ステップＳ６－３では、ＲＡＭ３上にあるストローク組み合わせ情報より、組み合わせ情報を読み出す。
【００３０】
ステップＳ６－４では、ストローク組み合わせ情報により文字を生成して文字データを出力バッファにセットする。
【００３１】
図７は、図６のステップ６－４における文字生成処理を示すフローチャートである。
【００３２】
ステップＳ３－１は、展開を終了したストローク数と、作成している文字を構成するストローク数ＳＴＲＯＫＥ＿ＣＯＵＮＴとの比較を行い、一致していなければ全ストロークの展開が終了していないものとしてステップＳ３－２に進み、一致していれば文字生成は終了したものとして図６のステップＳ６－５に進む。
【００３３】
ステップＳ３－２は、配置座標算出処理であり、文字出力座標系でのストローク配置座標Ｏ＿ＳＴＲＯＫＥ＿ＳＴ、Ｏ＿ＳＴＲＯＫＥ＿ＥＮを、フォント座標系でのストロークデータから、

Ｏ＿ＳＴＲＯＫＥ＿ＳＴ＝ＳＴＲＯＫＥ＿ＳＴ　×　（ＯＵＴ＿ＳＩＺＥ／８００）
Ｏ＿ＳＴＲＯＫＥ＿ＥＮ＝ＳＴＲＯＫＥ＿ＥＮ　×　（ＯＵＴ＿ＳＩＺＥ／８００）
で求める。ただし、文字出力座標系における文字サイズを（ＯＵＴ＿ＳＩＺＥ×ＯＵＴ＿ＳＩＺＥ）とする。

【００３４】

ステップＳ３－３は、制御点データスケール変換処理であり、
Ｓｃａｌｅ＝ａｂｓ（Ｏ＿ＳＴＲＯＫＥ＿ＳＴ－Ｏ＿ＳＴＲＯＫＥ＿ＥＮ）／ａｂｓ（Ｄ＿ＳＴＲＯＫＥ＿ＳＴ－Ｄ＿ＳＴＲＯＫＥ＿ＥＮ）
によりスケールを求め、
Ｓｃａｌｅ　×　Ｄ＿ＳＴＲＯＫＥ＿ＰＯＩＮＴ［　］
によりストロークの制御点データのスケール変換を行い、得られた値を、Ｏ＿ＳＴＲＯＫＥ＿ＰＯＩＮＴ［　］にセットする。

【００３５】

ステップＳ３－４は、ストロークウェイト情報調整処理であり、文字出力座標系でのウェイトをＯ＿Ｗｅｉｇｈｔとすると、
Ｗｅｉｇｈｔ＿Ｓｃａｌｅ＝Ｏ＿Ｗｅｉｇｈｔ／Ｄ＿ＳＴＲＯＫＥ＿Ｗｅｉｇｈｔ
よりウェイト変換スケールを求め、ストロークのウェイト調整をＯ＿ＳＴＲＯＫＥ＿ＰＯＩＮＴ［　］に施す。

【００３６】

ステップＳ３－５は、ストローク種判定処理であり、ストロークＩＤ及びストロークのスケルトンデータのスタート点のｘ，ｙ座標とエンド点のｘ，ｙ座標とを比較することにより、ストロークをストロークのボディ部分が水平・垂直となっている単純ストロークであるか、ボディ部分が斜線となっている通常ストロークであるかを判定する。また、通常ストロークにはボディが曲線のストロークも含まれるが、そのようなストロークは、ストロークの輪郭を規定するコントロールポイントの存在により判定することができる。また、ストロークＩＤに対応して、そのストロークが単純ストロークか通常ストロークかを識別する識別子を予めストロークデータの一部として格納しておき、それに基づいて判定することもできる。

【００３７】

この判定の結果、単純ストロークの場合ステップＳ３－６へ、通常ストロークの場合ステップＳ３－７へ進んで、それぞれのストロークの種類に応じたストローク構築処理を実行する。

【００３８】

ステップＳ３－８は、ストローク構築処理に得られたボディデータへヘッド＆テイルデータをマージする処理である。

【００３９】

図８は、単純ストロークの生成手順を示すフローチャートである。ステップＳ４－１は、ストロークボディに接続するヘッドが存在するかどうか判定するステップであり、ストロークデータのＨ＿ＩＤよりヘッドの有無を判定し、ヘッドアリの場合ステップＳ４－２に、無しの場合ステップＳ４－３に進む。

【００４０】

ステップＳ４－２は、ヘッドを接続するためにストロークのスタート点を新たに再計算するステップである。単純ストロークでは、ストロークが縦型ストロークならばｙ軸方向の演算のみ、ストロークが横型ストロークならばｘ軸方向の演算のみというように、ｘ軸方向またはｙ軸方向どちらか一方の演算により対処している。

【００４１】

ステップＳ４－３は、ボディ部のウェイトデータよりヘッド部に接続するヘッドデータの制御点列のスケール変換をする処理であり、文字出力座標系でのストロークウェイトをＯ＿Ｗｅｉｇｈｔとし、デザイン座標系でのヘッド部分のウェイトをＤ＿Ｈｅａｄ＿Ｗｅｉｇｈｔとすると、

Ｈｅａｄ＿Ｗｅｉｇｈｔ＿Ｓｃａｌｅ＝Ｏ＿Ｗｅｉｇｈｔ／Ｄ＿Ｈｅａｄ＿Ｗｅｉｇｈｔ
よりヘッド部分のウェイト変換スケールを求め、ヘッド部分のウェイト調整をＯ＿ＨＥＡＤ＿ＰＯＩＮＴ［　］に施す。なお、ヘッド及びテイルの制御点データは、図１０に示されるように接続口が、デザイン座標系での座標軸に対し水平方向または、垂直方向に対し平行になるように配置されデータとして予め格納されている。

【００４２】

ステップＳ４－４は、ボディに接続するテイルが存在するかどうか判定ステップであり、ストロークデータのＴ＿ＩＤよりテイルの有無を判定し、テイルアリの場合ステップＳ４－５、無しの場合ステップＳ４－７に進む。ステップＳ４－５は、ステップＳ４－２と同様の手法により、ストロークのエンド点を新たに再計算するステップである。ステップＳ４－６は、ステップＳ４－３と同様の手法でテイルデータのスケール変換をする処理であり、文字出力座標系でのストロークウェイトをＯ＿Ｗｅｉｇｈｔと、デザイン座標系でのテイル部分のウェイトＤ＿Ｔａｉｌ＿Ｗｅｉｇｈｔよりウェイト変換スケールを求め、テイル部分のウェイト調整をＯ＿Ｔａｉｌ＿ＰＯＩＮＴ［　］に施す。

【００４３】

ステップＳ４－７は、ボディの生成を行うステップであり、ウェイトデータ及びストロークスタート点、エンド点よりボディを構成する４点を生成する。

【００４４】

ステップＳ４－８は、ボディに接続するヘッドが存在するかどうか判定するステップであり、ヘッドが存在する場合ステップＳ４－９へ、存在しない場合ステップＳ４－１０へ進む。ステップＳ４－９は、ヘッドの制御点データをボディに接続する処理であり、ヘッドデータがボディデータに接続可能となるように適宜９０度単位の回転処理を施す。ステップＳ４－１０は、ボディに接続するテイルが存在するかどうか判定するステップであり、テイルが存在する場合ステップＳ４－１１へ、存在しない場合には図８の処理を終了して図７のステップ３－８へ進む。ステップＳ４－１１は、テイルの制御点データをボディに接続する処理であり、テイルデータがボディデータに接続可能となるように適宜９０度単位の回転処理を施す。

【００４５】

図９は、通常ストロークの生成手順を示すフローチャートである。ステップＳ５－１は、ストロークボディに接続するヘッドが存在するかどうか判定ステップであり、ストロークデータのＨ＿ＩＤよりヘッドの有無を判定し、ヘッドありの場合ステップＳ５－２へ、無しの場合ステップＳ５－４へ進む。ステップＳ５－２は、ボディ部のウェイトデータよりヘッド部に接続するヘッドデータの制御点列のスケール変換をする処理であり、文字出力座標系でのストロークウェイトをＯ＿Ｗｅｉｇｈｔとし、デザイン座標系でのヘッド部分のウェイトをＤ＿Ｈｅａｄ＿Ｗｅｉｇｈｔとすると
Ｈｅａｄ＿Ｗｅｉｇｈｔ＿Ｓｃａｌｅ＝０＿Ｗｅｉｇｈｔ／Ｄ＿Ｈｅａｄ＿Ｗｅｉｇｈｔ
よりヘッド部分のウェイト変換スケールを求め、ヘッド部分のウェイト調整をＯ＿ＨＥＡＤ＿ＰＯＩＮＴ［　］に施す。

【００４６】

ステップＳ５－３は、ヘッドを接続するためにストロークのスタート点を新たに再計算するステップであり、ストロークのスケルトンのベクトルの方向にヘッド部分の大きさ量をアフィン演算等を用いて算出し、そのベクトル量だけスタート座標から減算することによりスタート点の再計算を行う。

【００４７】

ステップＳ５－４は、ボディに接続するテイルが存在するかどうか判定ステップであり、ストロークデータのＴ＿ＩＤよりテイルの有無を判定し、テイル有りの場合ステップＳ５－５に、無しの場合ステップＳ５－７に進む。ステップＳ５－５はステップＳ４－３と同様の手法により、テイルデータのスケール変換をする処理であり、文字出力座標系でのストロークウェイトをＯ＿Ｗｅｉｇｈｔと、デザイン座標系でのテイル部分のウェイトＤ＿Ｔａｉｌ＿Ｗｅｉｇｈｔからウェイト変換スケールを求め、テイル部分のウェイト調整を

Ｏ＿Ｔａｉｌ＿ＰＯＩＮＴ［　］に施す。
【００４８】
ステップＳ５－６は、ステップＳ５－３と同様の手法でストロークのエンド点を新たに再計算するステップである。
【００４９】
ステップＳ５－７は、ボディの生成を行うステップであり、ウェイトデータ及びボディ部のデザイン座標系でのウェイトデータよりボディ部を構成するコントロールポイントのスケールを行いボディを構成するコントロールポイントを生成する。
【００５０】
ステップＳ５－８は、ボディに接続するヘッドが存在するかどうか判定するステップであり、ヘッドが存在する場合ステップＳ５－９へ、存在しない場合ステップＳ５－１１へ進む。ステップＳ５－９は、ヘッド部の制御点データをボディに接続可能とするようにするためのデータの回転角を求める処理であり、図１１に示されるようにヘッドデータの接続部の両端点ＡとＢを結ぶベクトル$\bar{}$Ａとボディ部の接続部の両端点ＣとＤを結ぶベクトル$\bar{}$Ｂとのなす角θを、
ｃｏｓθ＝（$\bar{}$Ａ・$\bar{}$Ｂ）／（｜$\bar{}$Ａ｜・｜$\bar{}$Ｂ｜）
により求める。ステップＳ５－１０は、求められた回転角θを使ってヘッド部の制御点データに対し回転処理を行い、ボディに接続可能なデータに変換する。
【００５１】
ステップＳ５－１１は、ボディに接続するテイル部が存在するかどうか判定するステップであり、テイル部が存在する場合ステップＳ５－１２へ進み、存在しない場合には図７のステップＳ３－８へ進む。ステップＳ５－１２は、ステップＳ５－９と同様にテイル部の制御点データをボディに接続可能とするようにするためのデータの回転角を求める処理であり、ヘッド部の場合と同じ要領で回転角γを求める。ステップＳ５－１３は、求められた回転角γを使いテイル部の制御点データに対し回転処理を行い、ボディに接続可能なデータに変換する。
【００５２】
このような手順により、ストロークデータから文字パターンを生成する。
【００５３】
以上説明したように、本実施例によれば、ストロークのボディの長手方向が座標軸方向に沿っており、ヘッド部及びテイル部のスターとポイントの計算や、ボディとヘッド部及びテイル部との接続のための回転処理が複雑な計算なしにできる場合と、ボディが座標軸に対して０度及び９０度以外の傾きをもっていたり、曲線であるような場合とでストローク構築処理を切替える。こうすることで、ストロークの形状によっては処理を大幅に簡素化することが可能となる。一般的に、垂直方向あるいは水平方向のストロークを有する文字は非常に多く、このような場合分けを行うことで文字の発生を高速に行うことが可能となる。なお、本実施例のストローク構築処理の切り替えを書体単位で切り替えるようにしても、本実施例は有効である。
【００５４】
なお、本実施例では、文字発生プログラムをＲＡＭ１０３上にロードして実行させる例を示したが、この他にもＦＤ１０９から、本プログラム及び関連データを一旦ＨＤ１０９に格納（インストール）しておき、本プログラムを動作させる段にＨＤ１０９からＲＡＭ１０３上にロードするようにしても良い。
【００５５】
また、本プログラムを記録する媒体は、ＦＤ以外にもＣＤ－ＲＯＭ、ＩＣメモリカード等であっても良い。更に本プログラムをＲＯＭ１０２に格納しておき、これをメモリマップの一部となすように構成し、直接ＣＰＵ１０１で実行することも可能である。
【００５６】
図１５は、図１３に示した文字パターン発生プログラム実行ファイルのモジュール構成を示す図である。このモジュールは、少なくとも、文字を、それを構成するストロークの組

み合わせとして記憶し、各ストロークを基本文字構成要素と局部輪郭との組み合わせとしてベクトル形式で記憶したデータを用いて、文字を構成するストロークデータを読み出す読出工程のコードと、読み出したストロークが、所定の形状であるか判定する判定工程のコードと、判定された形状に応じた最速の手順で、前記前記局部輪郭と前記基本文字構成要素とを接続してストロークを生成するストローク生成工程のコードと、前記ストローク生成工程で生成されたストロークを組合わせて文字画像を生成する文字生成工程のコードの各モジュールを含む。

［第２の実施の形態］

図１６は本発明にかかるシステムの基本的な構成を示すブロック図である。本システムは日本語ワードプロセッサであっても良いし、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良い。図１６において、１０１はＣＰＵ、即ち中央処理装置であり、この装置全体の制御及び演算処理などを行うものである。１０２はＲＯＭ即ち読み出し専用メモリであり、システム起動プログラム及び文字パターン・データなどの記憶領域である。１０３はＲＡＭ即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及びデータがロードされ、実行される領域である。１０４はＫＢＣ即ちキーボード制御部であり、１０５のＫＢ即ちキーボードよりキー入力データを受け取りＣＰＵ１０１へ伝達する。１０６はＣＲＴＣ即ちディスプレイ制御部であり、１０７はＣＲＴ即ちディスプレイ装置であり、１０６のＣＲＴＣよりデータを受け取り表示する。１０９はＦＤ即ちフレキシブルディスク装置、あるいはＨＤ即ちハードディスク装置などの外部記憶装置であり、プログラム及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ロードする。１０８はＤＫＣ即ちディスク制御部であり、データ伝送などの制御を行うものであり、１１０はＰＲＴＣ即ちプリンタ制御部であり、１１１はＰＲＴ即ちプリンタ装置である。１１２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り、１１３はビットマップイメージを回転させるイメージ回転器装置である。

【００５７】

以上の構成からなる本実施例のシステムによる文字パターン生成の動作は、第１の実施の形態と同じく図６のフローチャートに示される。

【００５８】

図１７（ｂ）はストローク組み合わせ情報の構造を示した図である。要素としてＳＴＲＯＫＥ＿ＣＯＵＮＴはフォントを構成するストローク数を示し、ＳＴＲＯＫＥ＿ＳＴ、ＳＴＲＯＫＥ＿ＥＮＤ、ＳＴＲＯＫＥ＿Ｗｅｉｇｈｔは、それぞれフォントデザイン座標系（本発明では、８００＊８００とする）でのストロークの芯線の両端点を基準とした配置座標、幅情報を示している。ストロークＩＤよりストロークデータを検索することにより、各ストロークのストロークデザイン座標系での大きさＤ＿ＳＴＲＯＫＥ＿ＳＩＺＥ、ウェイト情報Ｄ＿ＳＴＲＯＫＥ＿Ｗｅｉｇｈｔ、制御点データＤ＿ＳＴＲＯＫＥ＿ＰＯＩＮＴ［　］を読み出す。

【００５９】

図１８は、本実施例における図６のステップＳ６－４での文字生成処理を示すフローチャートであり、図２０が文字発生処理の手順を示す概念図である。本実施形態の処理では、縦長のストローク、すなわち、長手方向が出力デバイスにより画像を出力する際の走査方向に直交するストロークのビットマップ画像を、長手方向が走査方向に沿った水平なストロークとして生成し、その後回転変換することで本来の縦長のストロークを生成し、文字を構成する。図２０の例では、「十」の時の縦線の画像を９０度回転された横長のストロークとして生成し、イメージ回転器１１３により９０度回転して本来の縦線に戻し、最後に別途生成した横線と合成して文字「十」を生成している。

【００６０】

図１８において、ステップＳ１８－１では、文字生成終了判定を、生成を終了したストローク数と、生成する文字に含まれるストローク数ＳＴＲＯＫＥ＿ＣＯＵＮＴとの比較を行うことで行う。全ストロークの展開が終了していなければステップＳ１８－２に、終了し

ていれば図６に戻ってステップＳ６－５に進む。
【００６１】
ステップＳ１８－２ではストローク構築処理を行い、図１９はその詳しいフローチャートである。
【００６２】
ステップＳ１９－１では配置座標算出を行い、文字出力座標系でのストローク配置座標Ｏ＿ＳＴＲＯＫＥ＿ＳＴ，Ｏ＿ＳＴＲＯＫＥ＿ＥＮを
Ｏ＿ＳＴＲＯＫＥ＿ＳＴ＝ＳＴＲＯＫＥ＿ＳＴ　×　（ＯＵＴ＿ＳＩＺＥ／８００）
Ｏ＿ＳＴＲＯＫＥ＿ＥＮ＝ＳＴＲＯＫＥ＿ＥＮ　×　（ＯＵＴ＿ＳＩＺＥ／８００）
で求める。なお、文字出力座標系での文字のサイズは（ＯＵＴ＿ＳＩＺＥ×ＯＵＴ＿ＳＩＺＥ）であるものとする。
【００６３】
ステップＳ１９－２は、制御点データスケール変換処理であり、
Ｓｃａｌｅ＝ａｂｓ（Ｏ＿ＳＴＲＯＫＥ＿ＳＴ－Ｏ＿ＳＴＲＯＫＥ＿ＥＮ）／ａｂｓ（Ｄ＿ＳＴＲＯＫＥ＿ＳＴ－Ｄ＿ＳＴＲＯＫＥ＿ＥＮ）
によりスケールを求め、
Ｓｃａｌｅ　×　Ｄ＿ＳＴＲＯＫＥ＿ＰＯＩＮＴ［　］
によりストロークのスケール変換を行い、得られた値をＯ＿ＳＴＲＯＫＥ＿ＰＯＩＮＴ［　］にセットする。なお、関数ａｂｓ（ｘ）は、ｘの絶対値を出力とする。
【００６４】
ステップＳ１９－３は、ストロークウェイト情報調整処理であり、文字出力座標系でのウェイトをＯ＿Ｗｅｉｇｈｔとすると、
Ｗｅｉｇｈｔ＿Ｓｃａｌｅ＝Ｏ＿Ｗｅｉｇｈｔ／Ｄ＿ＳＴＲＯＫＥ＿Ｗｅｉｇｈｔ
よりウェイト変換スケールを求め、ストロークのウェイト調整をＯ＿ＳＴＲＯＫＥ＿ＰＯＩＮＴ［　］に施す。
【００６５】
ステップＳ１９－４は、ストロークを結合して文字を生成する処理であり、基本文字構成要素（ボディ）を形成する制御点列に局部輪郭（ヘッド部，テイル部）の制御点列を接続することによりストローク輪郭を発生する。
【００６６】
このようにしてストロークが構築されたなら、ステップＳ１８－３の処理に進む。
【００６７】
ステップＳ１８－３は、ストローク種判定処理であり、ストロークのスタート点（Ｏ＿ＳＴＲＯＫＥ＿ＳＴ）及びエンド点（Ｏ＿ＳＴＲＯＫＥ＿ＥＮ）を用いて縦長か横長ストロークか判定するステップである。縦長ストロークの場合ステップＳ１８－４へ、横長ストロークの場合ステップＳ１８－１０へ進む。
【００６８】
ステップＳ１８－１０は、横長のストロークの輪郭点を発生させるステップであり、ステップＳ１８－１１は、発生させた輪郭点を用いて輪郭内部を塗り潰し、ビットイメージを文字発生バッファＢに保存する。
【００６９】
一方、ステップＳ１８－４は、ストロークの輪郭を規定しているコントロールポイントの回転処理を行うステップであり、ストロークのコントロールポイントに対し９０度回転処理を行う。ステップＳ１８－５は、コントロールポイントを用いてストローク輪郭点を発生させる処理であり、ステップＳ１８－６では、発生させた輪郭点を用いて輪郭内部を塗り潰し、ビットイメージを描画バッファＡに保存する。この工程で生成されるイメージが、例えば図２０のイメージ２０１である。ステップＳ１８－７は、発生したストロークのビットイメージをハードウェアを用いたビットマップイメージ回転器により回転させるステップであり、これにより得られるイメージが、例えば図２０のイメージ２０２である。ステップＳ１８－８は、回転させたビットイメージを文字発生バッファＢに重ね合わせ（

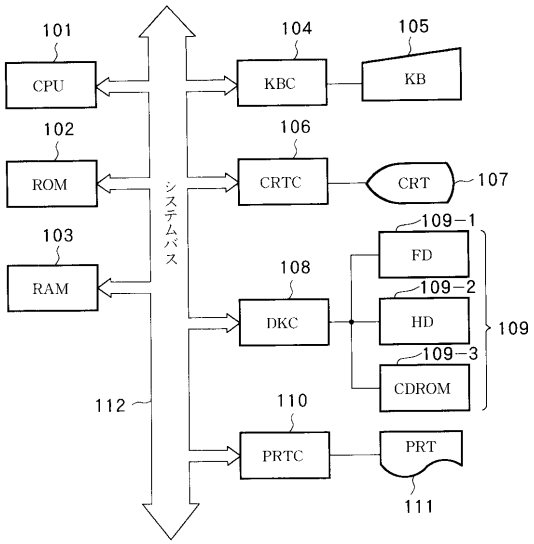
ＯＲ処理）を行うステップである。
【００７０】
こうして、縦長のストロークを次々と生成し、描画バッファＢに重ねあわせることでひとつの文字画像が生成される。生成された画像は、ステップＳ６－５の文字データ出力処理により、ＣＲＴ１０６やＰＲＴ１１０に所定の文字出力が行われる。
【００７１】
以上説明したように、本実施形態によれば、縦長ストロークを走査ライン方向に長い横形のストロークに変換し、横長のストロークをラスタライズしてビットマップ画像を生成する。通常走査方向は物理的に連続するメモリロケーションの方向と同一である。この場合、縦長、即ち長手方向が走査方向と直交する画像は、比較的短い（短手方向の画素数に相当）黒画素のクラスタが切れ切れのメモリロケーションに多数（長手方向の画素数に相当）格納されることになるが、横長、すなわち長手方向が走査方向に沿った画像では、比較的長い（長手方向の画素数に相当）連続したクラスタが小数（短手方向の画素数に相当）格納されることになる。従って、横長のストロークに対しては、連続したメモリロケーションにまとめて画像を展開することができるため、画像の輪郭内部の塗りつぶし処理を高速に行うことができ、文字の発生を高速に行うことが可能となる。
［第３の実施の形態］
次に、本発明の第３の実施の形態について述べる。
【００７２】
ここで説明するのは、前記第２の実施形態における描画バッファＡのサイズが、描画バッファＢと同一サイズ確保できる場合の文字生成処理である。したがって、図６のステップＳ６－４の詳細な処理手順として、第２実施形態の図１８の代わりに、本実施形態では図２１の手順で文字生成を行う。
【００７３】
図２１は、図６のステップＳ６－４における文字発生処理の詳細を示すフローであり、図２２が文字発生処理の動きを示す概念図である。
【００７４】
ステップＳ２１－１は、文字展開終了の判定を行い、展開を終了したストローク数と、展開している文字に含まれるストローク数ＳＴＲＯＫＥ＿ＣＯＵＮＴとの比較を行い、全ストロークの展開が終了していなければステップＳ２１－２に、終了していれば図６のステップＳ６－５に進む。
【００７５】
ステップＳ２１－２は、ストローク構築処理であり、詳細は第２実施例のステップＳ１８－２と同一のものである。
【００７６】
ステップＳ２１－３は、ストローク種判定処理であり、ストロークのスタート点（Ｏ＿ＳＴＲＯＫＥ＿ＳＴ）及びエンド点（Ｏ＿ＳＴＲＯＫＥ＿ＥＮ）を用いて、縦長ストロークか横長ストロークか判定する。縦長ストロークの場合、ステップＳ２１－４へ、横長ストロークの場合ステップＳ２１－７へ進む。ステップＳ２１－４はコントロールポイントの回転処理を行うステップであり、ストロークのコントロールポイントに対し９０度回転処理を行う。ステップＳ２１－５は、コントロールポイントを用いてストローク輪郭点を発生させる処理であり、ステップＳ２１－６は、発生させた輪郭点を用いて輪郭内部を塗り潰し、ビットイメージを描画バッファＡに保存する。
【００７７】
上記ステップＳ２１－２～ステップＳ２１－６を繰返して、バッファＡには本来縦長のストロークだけが、９０度回転された状態で、文字上での配置に相当する配置で生成される。
【００７８】
一方、ステップＳ２１－７は、横長のストロークの輪郭点を発生させるステップであり、ステップＳ２１－８は、発生させた輪郭点を用いて輪郭内部を塗り潰し、ビットイメージ

を文字発生バッファＢに保存する。
【００７９】
ステップＳ２１－１で、全ストロークの発生を終了したと判定されると、ステップＳ２１－９へ進む。ステップＳ２１－９は、ストロークの回転発生処理を行ったか判定するステップであり、回転発生を行ったと判定されるとステップＳ２１－１０へ進む。ステップＳ２１－１０は、描画バッファＡのイメージを回転器を用いて回転させるステップである。ステップＳ２１－１１は、ステップＳ２１－１０で回転させたバッファＡのビットマップイメージを文字発生バッファＢにＯＲ処理を行って重畳する。
【００８０】
係るシステムは、上述の処理を経てステップＳ６－５の文字データ出力を、ＣＲＴ１０６やＰＲＴ１１０に所定の文字出力を行う。
【００８１】
以上説明したように、本実施例によれば、縦ストローク用バッファＡ及び横ストロークバッファＢを用いて、文字を構成するストロークをどちらかのバッファに描画し、全ストローク発生後、縦ストロークバッファＡを回転器により回転させバッファＢに重ね合わせることにより文字の発生を高速に行うことが可能となる。本実施形態の処理によれば、第２の実施の形態に比べて、９０度回転したストロークを本来の向きに戻す際に、一括して行うために、処理がその分高速になる。
【００８２】
なお、文字を構成するストロークの縦ストロークの数が少ない時には第２の実施の形態の処理を用いて文字を発生し、縦ストロークが多い場合本実施形態用いたり、搭載メモリが少なく、描画バッファＡのサイズがＡ＜Ｂである場合に、第２の実施形態を、バッファＡ＞＝バッファＢとなるサイズのバッファが確保可能な場合本実施形態を用いるというように、文字生成の手順を切り替えて実施することもできる。
【００８３】
さらに、第１の実施の形態で説明した、ストロークの形状が単純か否かにより場合分けしてストロークを生成する処理と合わせれば、更に高速に文字画像生成を行うことができる。
【００８４】
【他の実施形態】
なお、本発明は、複数の機器（例えばホストコンピュータ，インタフェイス機器，リーダ，プリンタなど）から構成されるシステムに適用しても、一つの機器からなる装置（例えば、複写機，ファクシミリ装置など）に適用してもよい。
【００８５】
また、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を記録した記憶媒体を、システムあるいは装置に供給し、そのシステムあるいは装置のコンピュータ（またはＣＰＵやＭＰＵ）が記憶媒体に格納されたプログラムコードを読出し実行することによっても、達成される。
【００８６】
この場合、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。
【００８７】
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，ＣＤ－ＲＯＭ，ＣＤ－Ｒ，磁気テープ，不揮発性のメモリカード，ＲＯＭなどを用いることができる。
【００８８】
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だけでなく、その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ＯＳ（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

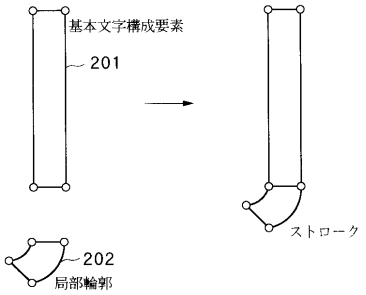
を行い、その処理によって前述した実施形態の機能が実現される場合も含まれる。
【００８９】
さらに、記憶媒体から読出されたプログラムコードが、コンピュータに挿入された機能拡張ボードやコンピュータに接続された機能拡張ユニットに備わるメモリに書込まれた後、そのプログラムコードの指示に基づき、その機能拡張ボードや機能拡張ユニットに備わるＣＰＵなどが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれる。
【００９０】
本発明を上記記憶媒体に適用する場合、その記憶媒体には、先に説明したフローチャートに対応するプログラムコードを格納することになるが、簡単に説明すると、図１５や、図２３のメモリマップ例に示す各モジュールを記憶媒体に格納することになる。
【００９１】
図１５の構成は前述の通りである。図２３においては、文字を、それを構成するストロークの組み合わせとして記憶し、各ストロークを基本文字構成要素と局部輪郭との組み合わせとしてベクトル形式で記憶したデータを用いて、文字を構成するストロークデータを読み出す読み出し工程のコードと、読み出したストロークデータに基づいて、ストロークが縦長か横長か判定する判定工程のコードと、縦長と判定されたストロークについては横長に変換し、横長と判定されたストロークはそのままでビットマップ画像を生成し、横長に変換されたストロークのビットマップ画像を元に戻すストローク生成工程のコードと、前記ストローク生成工程で生成されたストロークを組合わせて文字画像を生成する文字生成工程のコードが含まれていれば良い。
【００９２】
【発明の効果】
以上説明したように、本発明の文字生成方法及び装置は、ストロークの形状によりストロークの構築処理を切り替えることにより、ストローク構築処理を大幅に簡素化することが可能となり、文字の発生を高速に行うことが可能となる。
【００９３】
また、縦長のストロークを高速に発生可能なスキャンライン方向に長いストロークに変換して発生させることで、文字発生速度を大幅に向上させることが可能である。
【００９４】
【図面の簡単な説明】
【図１】本発明の第１の実施の形態に係るシステムの構成を示すブロック図である。
【図２】基本文字構成要素、局部輪郭データを規定する図である。
【図３】ストロークデータを説明する図である。
【図４】ストローク配置、属性データの例を示す図である。
【図５】本発明の第１の実施の形態に係るもうひとつのシステムの構成を示すブロック図である。
【図６】第１～第３の実施形態に係るシステムでの文字パターン発生手順を示すフローチャートである。
【図７】第１の実施の形態における文字生成処理を示すフローチャートである。
【図８】単純ストローク構築処理を示すフローチャートである。
【図９】通常ストローク構築処理を示すフローチャートである。
【図１０】デザイン座標系でのヘッド、テイルデータの配置を示す図である。
【図１１】ストロークボディとヘッド部のなす角度を示す図である。
【図１２】第１～第３実施形態のメモリ構成を示す図である。
【図１３】第１～第３の実施形態におけるプログラムを記録したＦＤあるいはＣＤ－ＲＯＭ等の媒体の内容構成図である。
【図１４】第１～第３の実施形態のプログラムを記録したＦＤあるいはＣＤ－ＲＯＭ等の媒体から、本プログラムを供給することを示す概念図である。
【図１５】第１の実施形態におけるプログラムを記録したＦＤあるいはＣＤ－ＲＯＭ等の

媒体における文字パターン発生プログラムファイルの構成図である。
【図１６】第２及び第３実施形態に係るシステムの構成を示すブロック図である。
【図１７】第２及び第３実施形態におけるストローク配置データの例を示す図である。
【図１８】第２実施形態のシステムでの文字生成処理を示すフローチャートである。
【図１９】第２及び第３の実施形態におけるストローク構築処理の手順を示すフローチャートである。
【図２０】第２の実施形態での文字生成の概念を示す図である。
【図２１】第３の実施形態のシステムでの文字生成処理を示すフローチャートである。
【図２２】第３の実施形態での文字生成の概念を示す図である。
【図２３】第２及び第３の実施形態におけるプログラムを記録したＦＤあるいはＣＤ－ＲＯＭ等の媒体における文字パターン発生プログラムファイルの構成図である。
【符号の説明】
１０１　ＣＰＵ
１０２　ＲＯＭ
１０３　ＲＡＭ
１０４　ＫＢＣ
１０５　キーボード
１０６　ＣＲＴＣ
１０７　ＣＲＴ
１０８　ＤＫＣ
１０９　外部記憶装置
１１０　ＰＲＴＣ
１１１　ＰＲＴ
１１２　システムバス
１１３　ビットイメージ回転器

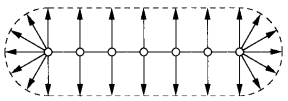
【図１】



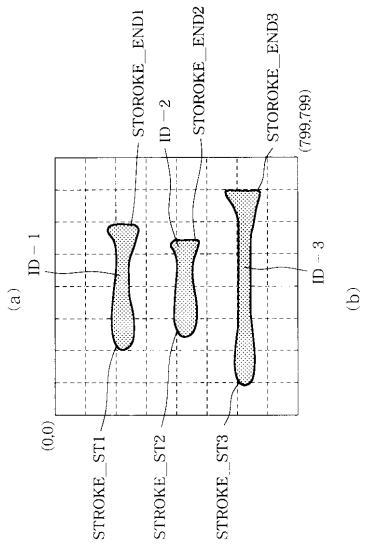
【図２】



【図３】



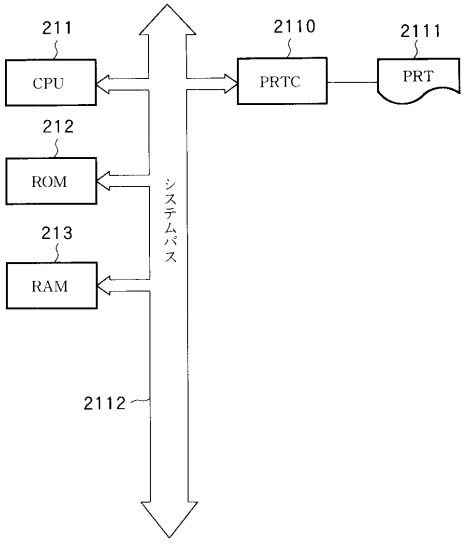
【図４】



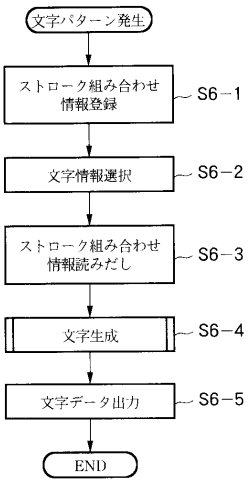
(b)

STOROKE_COUNT	3			
ストローク ID	ストローク配置座標	ストローク Weight	H_ID	T_ID
1	STROKE_ST1	STROKE_END1	50	H12	T14
2	STROKE_ST2	STROKE_END2	50	H23	なし
3	STROKE_ST3	STROKE_END3	50	なし	T31

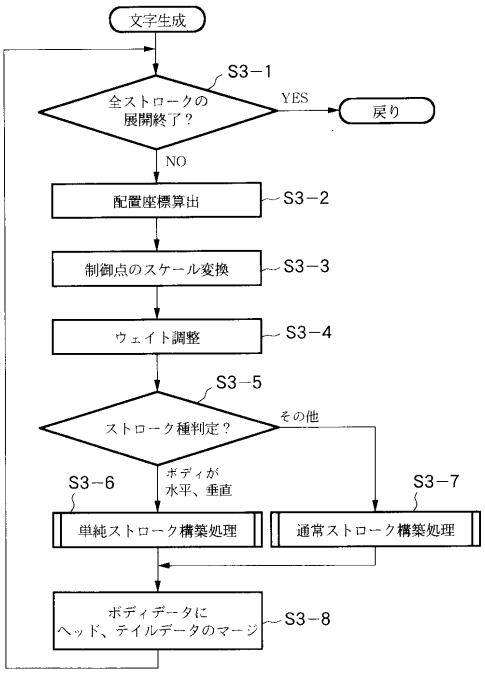
【図５】



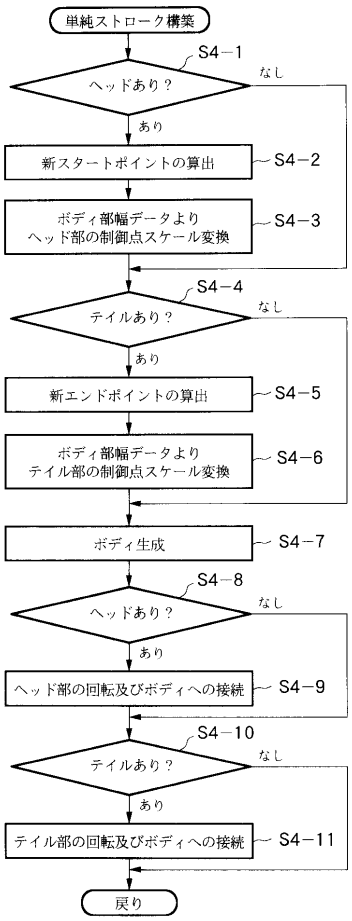
【図６】



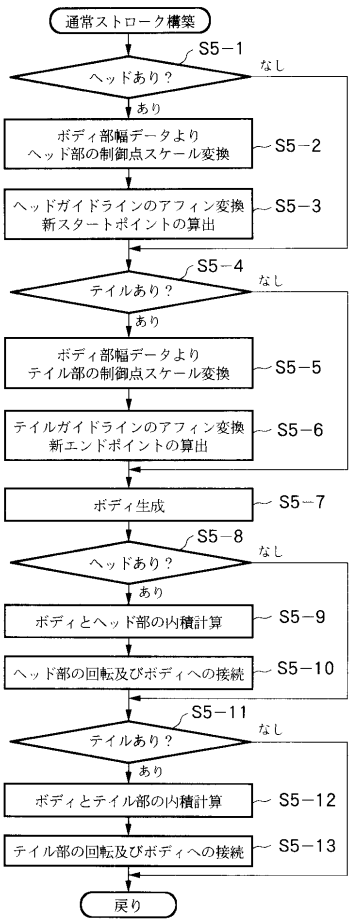
【図７】



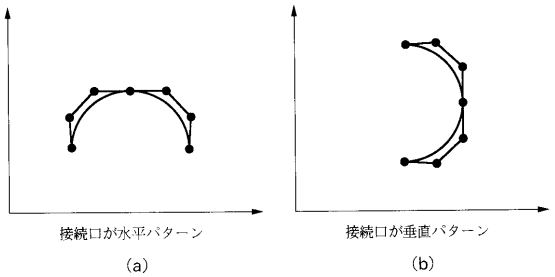
【図８】



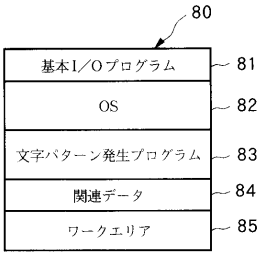
【図９】



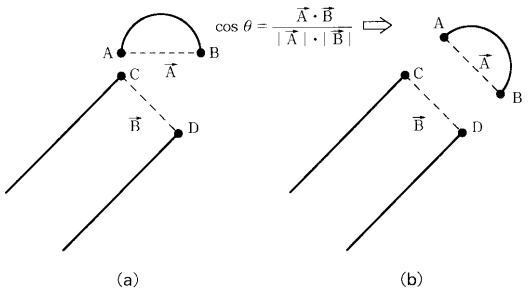
【図１０】



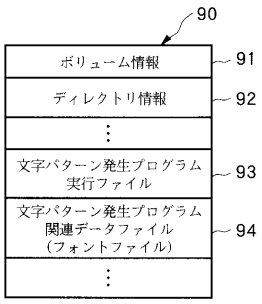
【図１２】



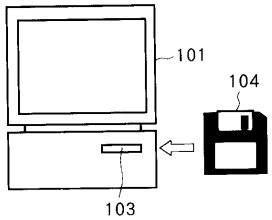
【図１１】



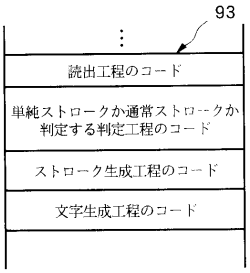
【図１３】



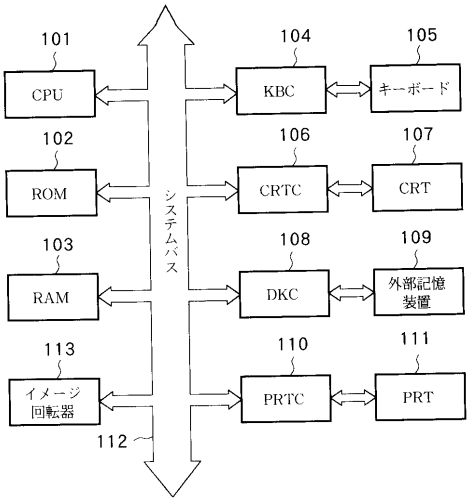
【図１４】



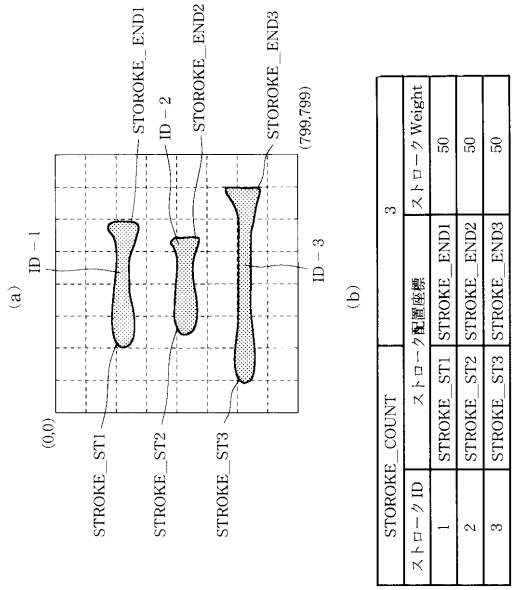
【図１５】



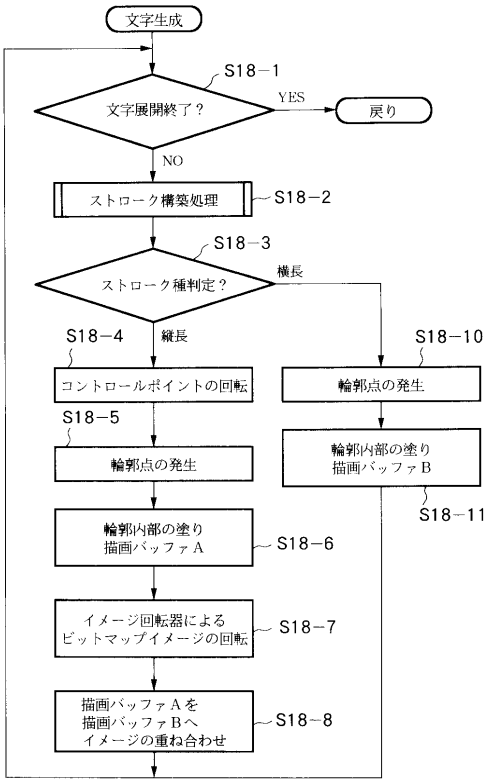
【図１６】



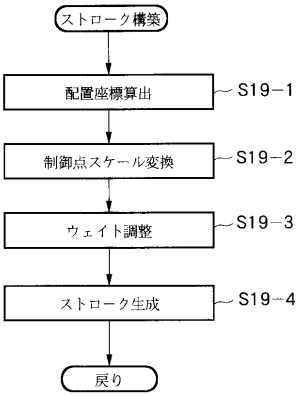
【図１７】



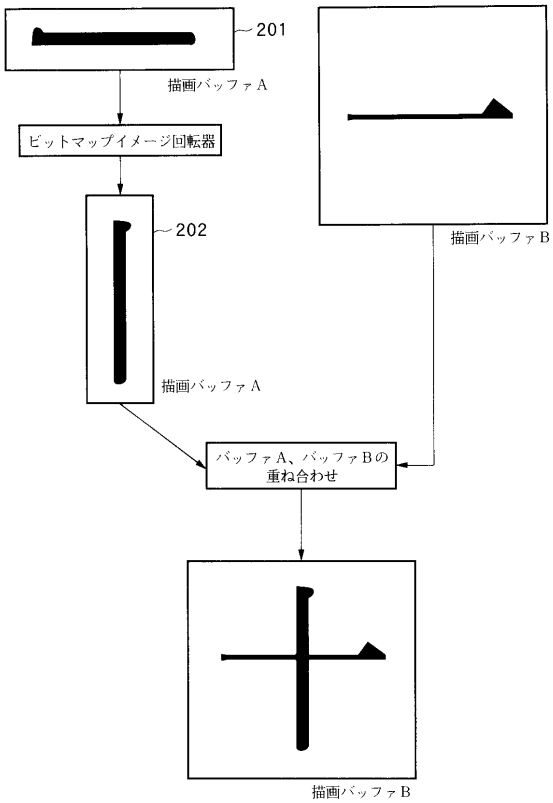
【図１８】



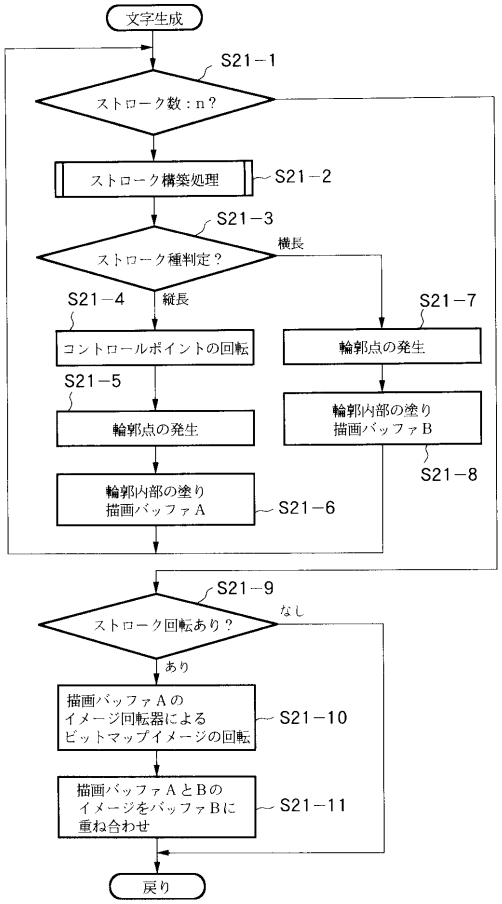
【図１９】



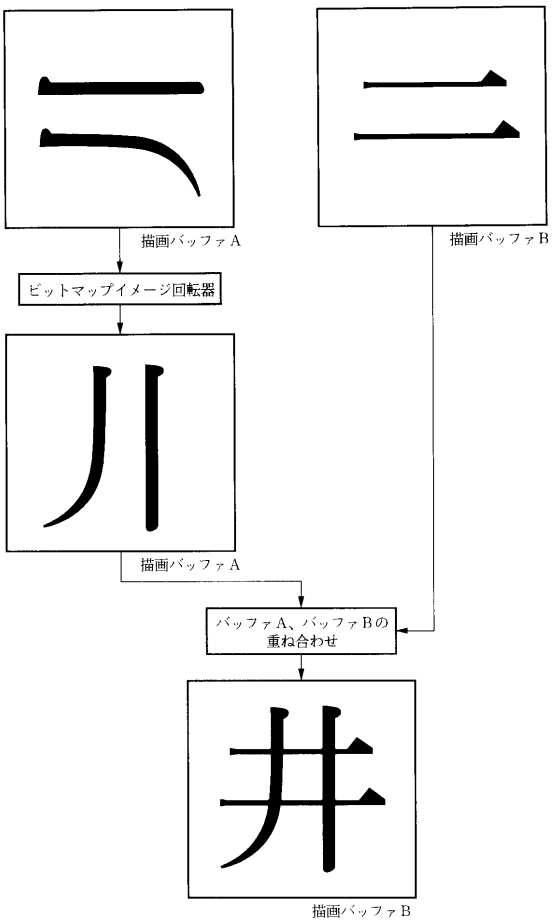
【図２０】



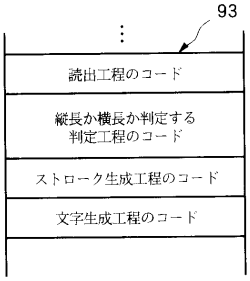
【図２１】



【図２２】



【図２３】


93
読出工程のコード
縦長か横長か判定する
判定工程のコード
ストローク生成工程のコード
文字生成工程のコード

フロントページの続き

(56)参考文献 特開平４－５６８９４（ＪＰ，Ａ）
特開平５－１６５４５７（ＪＰ，Ａ）
特開平９－１０６２７１（ＪＰ，Ａ）
特開平９－３０５１６２（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.[7]，ＤＢ名)
G09G 5/24