(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第3817281号
(P3817281)
(45)発行日　平成18年9月6日(2006.9.6)　(24)登録日　平成18年6月16日(2006.6.16)

(51)Int.Cl.			F I		
G06K	9/62	(2006.01)	G06K	9/62	G
G06K	9/46	(2006.01)	G06K	9/46	B

請求項の数 6　(全 16 頁)

(21)出願番号　特願平7-195233
(22)出願日　平成7年7月31日(1995.7.31)
(65)公開番号　特開平9-44599
(43)公開日　平成9年2月14日(1997.2.14)
審査請求日　平成14年7月5日(2002.7.5)

(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　▲高▼須　英司
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
(72)発明者　新井　常一
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

最終頁に続く

(54)【発明の名称】情報処理装置及び方法

(57)【特許請求の範囲】
【請求項１】
所定の入力装置より入力された手書きストロークから、当該ストロークを構成する点データ群Ｐ（０），…，Ｐ（ｋ）から特徴点を抽出し、当該特徴点に基づく情報を下位の処理に渡す情報処理装置であって、
前記入力装置より入力された手書きストローク情報である点データＰ（０）、…、Ｐ（ｋ）を記憶するストローク記憶手段と、
該ストローク記憶手段に記憶された点データにおいて、
前記点データの起点Ｐ（ｉ）（初期値ｉ＝０）、
参照点Ｐ（ｉ＋１＋ｊ）（初期値ｊ＝１）とし、
起点Ｐ（ｉ）とＰ（ｉ＋１）で定められるベクトルを基準ベクトル、Ｐ（ｉ＋１）と前記参照点Ｐ（ｉ＋１＋ｊ）で定められるベクトルを参照ベクトルと定義したとき、
前記基準ベクトルと参照ベクトルとの成す角度θと所定閾値とを比較する比較手段と、
該比較手段の比較結果、前記角度θが所定閾値以下である場合には、点Ｐ（ｉ＋ｊ）を削除対象に設定すると共に、前記変数ｊを“１”だけ増加させて前記比較手段を再実行し、
前記比較手段の比較結果、前記角度θが所定閾値を越えた場合、点Ｐ（ｉ＋ｊ）を特徴点として判断し、その際の変数ｊに基づいて前記変数ｉを更新すると共に、前記変数ｊを“１”で再初期化し、前記比較手段を再実行する第１の制御手段と、
前記第１の制御手段で得られた特徴点の個数が、注目ストロークに関してｎ個の連続し

たベクトル列を特定し得る数になったか否かを判別する判別手段と、
該判別手段によって、ｎ個の連続したベクトル列を特定し得る数より少ないと判別した場合は、前記所定閾値を減じた後、前記第１の制御手段による処理を再度実行する第２の制御手段と
を備えることを特徴とする情報処理装置。
【請求項２】
更に、前記判別手段によって、連続したベクトルの数が前記ｎ個を越えたと判別された場合は、前記第１の制御手段で得られた特徴点のうち、前記角度θの大きいものから上位ｎ＋１個の特徴点データを抽出する抽出手段を備えることを特徴とする請求項第１項に記載の情報処理装置。
【請求項３】
前記下位処理は文字認識処理であることを特徴とする請求項第１項に記載の情報処理装置。
【請求項４】
所定の入力装置より入力された手書きストロークから、当該ストロークを構成する点データ群Ｐ（０），…，Ｐ（ｋ）から特徴点を抽出し、当該特徴点に基づく情報を下位の処理に渡す情報処理方法であって、
前記入力装置より入力された手書きストローク情報である点データＰ（０）、…、Ｐ（ｋ）をストローク記憶手段に格納する格納工程と、
該ストローク記憶手段に記憶された点データにおいて、
前記点データの起点をＰ（ｉ）（初期値ｉ＝０）、
参照点をＰ（ｉ＋１＋ｊ）（初期値ｊ＝１）とし、
起点Ｐ（ｉ）とＰ（ｉ＋１）で定められるベクトルを基準ベクトル、Ｐ（ｉ＋１）と前記参照点Ｐ（ｉ＋１＋ｊ）で定められるベクトルを参照ベクトルと定義したとき、
前記基準ベクトルと参照ベクトルとの成す角度θと所定閾値とを比較する比較工程と、
該比較工程の比較結果、前記角度θが所定閾値以下である場合には、点Ｐ（ｉ＋ｊ）を削除対象に設定すると共に、前記変数ｊを“１”だけ増加させて前記比較工程を再実行し、
前記比較工程の比較結果、前記角度θが所定閾値を越えた場合、点Ｐ（ｉ＋ｊ）を特徴点として判断し、その際の変数ｊに基づいて前記変数ｉを更新すると共に、前記変数ｊを“１”で再初期化し、前記比較工程を再実行する第１の制御工程と、
前記第１の制御工程で得られた特徴点の個数が、注目ストロークに関してｎ個の連続したベクトル列を特定し得る数になったか否かを判別する判別工程と、
該判別工程によって、ｎ個の連続したベクトル列を特定し得る数より少ないと判別した場合は、前記所定閾値を減じた後、前記第１の制御手段による処理を再度実行する第２の制御工程と
を備えることを特徴とする情報処理方法。
【請求項５】
更に、前記判別工程によって、連続したベクトルの数が前記ｎ個を越えたと判別された場合は、前記第１の制御工程で得られた特徴点のうち、前記角度θの大きいものから上位ｎ＋１個の特徴点データを抽出する抽出工程を備えることを特徴とする請求項第４項に記載の情報処理方法。
【請求項６】
前記下位処理は文字認識処理であることを特徴とする請求項第４項に記載の情報処理方法。
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】
本発明は情報処理装置及び方法、詳しくは、入力された手書きストロークから、当該ストロークを構成する点データ群の中の特徴点を抽出し、当該特徴点に基づく情報を下位の処

理に渡す情報処理装置及び方法に関するものである。

【０００２】

【従来の技術】

従来、入力された筆跡を文字認識して、文字コードに変換する文字認識方法には数々の手法が考案されている。その手法の中にベクトルマッチング方式と呼ばれるものがある。この手法は以下のとおりである。

【０００３】

まず、入力された筆跡に対し、ペンダウン情報とペンアップ情報によって区切られた座標点列を１ストロークと呼び、そのストロークをベクトル化してストロークベクトルとして表現する。次に、複数のストロークベクトルの個々について、予め認識辞書に納められている標準ストロークベクトルと比較し、相違度を計算する。さらに、認識辞書には文字を構成するいくつかの標準ベクトルが定義されており、この定義された標準ストロークベクトル集合を参照して、入力されたストロークベクトル集合との一致度を計算して最も類似していると判断される文字を認識結果として出力するというものである。

【０００４】

ここで、ストロークをベクトル化する方法は、一般的に、入力された座標点を始点から終点までつないでストローク化した後、ストローク線分全体をｎ分割することによりｎ本のベクトルを作成している。このベクトル列がストローク特徴として抽出されている。ｎ分割の方法はストロークの全長をｎで割り、ストロークを等距離間隔で分割するものであった。

【０００５】

【発明が解決しようとする課題】

しかしながら、上記従来例のような、オンライン手書き文字認識方法ではストロークの全長をｎ等分した距離を元に分割点を決定していたが、類似文字、例えば「る」と「ろ」、「ぬ」と「め」のようにストロークの最終点近辺にその文字の特徴がある場合、単なる等距離間隔の分割では精度の良い分割ができず、その分割点によるベクトル列も類似文字間で識別が困難であるという問題点があった。

【０００６】

【課題を解決するための手段】

本発明は上記問題点に鑑みなされたものであり、入力された手書きストローク情報から、当該ストロークの精度良い特徴情報を抽出して下位の処理に渡す情報処理装置及び方法を提供しようとするものである。

【０００７】

この課題を解決するため、例えば本発明の情報処理装置は以下に示す構成を備える。すなわち、

所定の入力装置より入力された手書きストロークから、当該ストロークを構成する点データ群Ｐ（０），…，Ｐ（ｋ）から特徴点を抽出し、当該特徴点に基づく情報を下位の処理に渡す情報処理装置であって、

前記入力装置より入力された手書きストローク情報である点データＰ（０）、…、Ｐ（ｋ）を記憶するストローク記憶手段と、

該ストローク記憶手段に記憶された点データにおいて、

前記点データの起点Ｐ（ｉ）（初期値ｉ＝０）、

参照点Ｐ（ｉ＋１＋ｊ）（初期値ｊ＝１）とし、

起点Ｐ（ｉ）とＰ（ｉ＋１）で定められるベクトルを基準ベクトル、Ｐ（ｉ＋１）と前記参照点Ｐ（ｉ＋１＋ｊ）で定められるベクトルを参照ベクトルと定義したとき、

前記基準ベクトルと参照ベクトルとの成す角度θと所定閾値とを比較する比較手段と、

該比較手段の比較結果、前記角度θが所定閾値以下である場合には、点Ｐ（ｉ＋ｊ）を削除対象に設定すると共に、前記変数ｊを“１”だけ増加させて前記比較手段を再実行し、

前記比較手段の比較結果、前記角度θが所定閾値を越えた場合、点Ｐ（ｉ＋ｊ）を特徴

点として判断し、その際の変数ｊに基づいて前記変数ｉを更新すると共に、前記変数ｊを“１”で再初期化し、前記比較手段を再実行する第１の制御手段と、

前記第１の制御手段で得られた特徴点の個数が、注目ストロークに関してｎ個の連続したベクトル列を特定し得る数になったか否かを判別する判別手段と、

該判別手段によって、ｎ個の連続したベクトル列を特定し得る数より少ないと判別した場合は、前記所定閾値を減じた後、前記第１の制御手段による処理を再度実行する第２の制御手段とを備える。

【０００９】

また、この場合には、更に、前記判別手段によって、連続したベクトルの数が前記ｎ個を越えたと判別された場合は、前記第１の制御手段で得られた特徴点のうち、前記角度θの大きいものから上位ｎ＋１個の特徴点データを抽出する抽出手段を備えることが望ましい。これによって常にｎ本のベクトルデータを抽出できるようになる。

【００１０】

また、下位処理は、文字認識処理であることが望ましい。これによって、文字認識率を向上させることが可能になる。

【００１１】

【発明の実施の形態】

以下、添付図面にしたがって本発明にかかる実施形態の例を詳細に説明する。

【００１２】

＜第１の実施形態＞

図１は本第１の実施形態におけるオンライン手書き文字認識装置の機能構成を示すブロック図である。図示において、１０１は座標入力ペンを操作することで座標を入力するダブレット、ディジタイザ等の座標入力装置であり、本実施形態では、入力された文字データを獲得する文字入力部である。１０２はストローク格納部で、１０１の文字入力部にて得られた座標点を用いて、ペンダウン情報とペンアップ情報の間の座標点集合を一本のストロークとして格納する。１０３は特徴抽出部であって、ストローク格納部１０２中のストロークデータの中から特徴点となるデータを抽出する。また、１０４は特徴点消去部であり、特徴点抽出部１０４で得られた特徴点データからｎ個のベクトルを構成するためのｎ＋１個の点データを生成するため、不要なデータを消去する特徴点消去部である。１０５はベクトル化部であって、特徴点消去部１０５から得られたｎ＋１個の点データに基づきｎ個のベクトルデータを生成する。以上の如く、文字入力部１０１でもって１文字分の手書き文字を入力した場合、ペンダウンからペンアップまでのストローク情報が順次取り出され、ベクトル化部１０５には、各ストロークに対するｎ本のベクトルが格納される。

【００１３】

１０６は認識部であり、ベクトル化部１０５から各ストロークに対するベクトルデータを入力し、認識辞書部１０８に納められている基本ベクトルとの間で比較演算処理して文字認識する。１０７は、認識部１０６から出力された認識結果（文字コード）を出力する結果出力部である。尚、文字認識処理では、通常、最も確からしい候補を第１候補とし、次いで第２候補、第３候補、…として出力する。本実施形態でもこれに準拠する。

【００１４】

上記構成における特徴抽出部１０３の処理内容を図２、図３のフローチャートを用いて説明する。

【００１５】

図２は記入されたストローク（ペンダウンとペンアップの間の筆跡）の図であり、構成する座標点を図示している。座標点Ｐ０は始点、座標点Ｐｋは終点となる。従って座標点の個数は全部でｋ＋１個になる。

【００１６】

図３はストロークの特徴点を抽出する処理内容を示すフローチャートである。尚、以下の説明で、点Ｐｎから点Ｐ（ｎ＋１）に向かうベクトルをＶ（ＰｎＰ(n+1)）として表わす。

【００１７】

まず、ステップＳ３０１にて開始点カウンタｉを０にセットする。ステップＳ３０２にて対象点カウンタｊを１にセットする。そして、ステップＳ３０３では、変数ｉ，ｊの合計値がｋ以下であるか否か、すなわち、本処理が終了したか否かを判断する。否の場合には、以下のステップＳ３０４に進む。

【００１８】

ステップＳ３０４では、ベクトルＶ（ＰｉＰ(i+1)）とベクトルＶ（Ｐ(i+1)Ｐ(i+1+j)）の成す角度θが予め設定された閾値角度θt以下であるか否かを判断する。θ≦θtであれば、ステップＳ３０５に進み、座標点Ｐ（ｉ＋ｊ）を削除対象点としてマーキングし、ステップＳ３０６では対象点カウンタｊをインクリメントして次の座標点の削除チェックを行うためにステップＳ３０３に戻る。

【００１９】

以下、この処理を繰り返していくが、この期間中、２つのベクトルのなす角度θとθtの関係が、θ＞θt、という関係になっていると判断した場合には、ステップＳ３０７に進み、変数ｉを“ｉ＋１＋ｊ”で更新し、ステップＳ３０２に戻る。

【００２０】

さて、ステップＳ３０５にて全ての座標点のチェックを終了すると、本処理は完了する。この処理終了後、マーキングされた座標点は、角度変化の少ない連続している座標点であるので、この点群を削除して残されたものは角度の大きい部分の座標点となる。尚、始点Ｐ0と終点Ｐkは共に特徴点とみなすこととする。また、尚、上記処理における閾値角度θtはパラメータ化されており、可変である。理由は後述する。

【００２１】

上記処理内容を図４に従ってより詳しく説明する。今、点Ｐｉを基点とし、ｊ＝１の場合を想定する。この場合、２つのベクトルは、

Ｖ（ＰｉＰ(i+1)）、Ｖ（Ｐ(i+1)Ｐ(i+2)）

となる。

【００２２】

これら２つのベクトルのなす角度θ1が閾値角度θt以下であるとした場合、２つ目のベクトルＶ（Ｐ(i+1)Ｐ(i+2)）の終点Ｐ(i+2)の１つ手前の点Ｐ(i+1)は削除対象として設定される。この結果、変数ｊが１つインクリメントされるので（ｊ＝２となる）、次に、

Ｖ（ＰｉＰ(i+1)）、Ｖ（Ｐ(i+1)Ｐ(i+3)）

のなす角度θ2と閾値角度θtと比較されることになる。ここでθ2も閾値角度θt以下である場合には、点Ｐ(i+2)が削除対象として設定され、変数ｊがインクリメントされる（ｊ＝３）。この結果、次の２つのベクトルのなし角度θ3と閾値角度θtとが比較される。すなわち、

Ｖ（ＰｉＰ(i+1)）、Ｖ（Ｐ(i+1)Ｐ(i+4)）

このθ3が閾値角度θtより大きい場合には、点Ｐ(i+3)は特徴点（特異点）として設定される。

【００２３】

以上の結果、θ1,θ2が共に閾値角度θt以下であって、θ3が閾値角度θtより大きい場合、点Ｐ(i+1),Ｐ(i+2)が削除対象として設定されることになる。すなわち、ストロークを構成している各座標点の並びの曲がり具合が小さい場合には、それぞれの点が削除対象として設定されることになる。

【００２４】

尚、ステップＳ３０７では、変数ｉを“ｉ＋１＋ｊ”で更新し、ステップＳ３０２で変数ｊを“１”で初期化するので、点Ｐｉは図４における点“Ｐ(i+4)”の位置に移動し、そこから上述した処理が繰り返し実行されることになる。

【００２５】

さて、ストロークをｎ分割するためには（１ストロークからｎ個のベクトルを抽出するためには）、特徴点の個数はｎ＋１個必要になる。ところが、上記の処理を行った場合には、閾値角度θｔや注目しているストロークに依存して得られる個数が不定であり、一律に

ｎ＋１の点を得ることができない。そこで、本実施形態では、上記図３の処理を行う場合の閾値θｔを可変にし、それでもって適当な数の特徴点を抽出するようにした。この処理を図５のフローチャートに従って説明する。

【００２６】

先ず、ステップＳ５０１では特徴抽出処理で用いた閾値角度θｔに初期値として１８０°にセットする。ステップＳ５０２では、設定されている閾値角度θｔを用いて図３で説明した特徴抽出処理を行う。

【００２７】

この後、処理はステップＳ５０３に進んで、ステップＳ５０２で得られた特徴点の個数をチェックし、ｎ＋１個以上の特徴点数になったか否かを判断する。ここで、抽出された特徴点の個数がｎ＋１に満たない場合には、ステップＳ５０４に進み、閾値角度θｔを所定角度（例えば１°）減じ、ステップＳ５０２に戻る。

【００２８】

以下、閾値角度θｔを順次減じ、特徴点抽出部１０３による処理を行うと、得られる特徴点の数は増えていき、ついにはｎ＋１個以上の特徴点が得られることになる。このとき、処理はステップＳ５０３からステップＳ５０５に進み、得られた特徴点データを特徴点消去部１０４に出力し、本処理を終える。

【００２９】

次に、特徴点消去部１４５の処理動作を図６のフローチャートを用いて説明する。

【００３０】

先に説明した様に、特徴点抽出部１０３からは、少なくともｎ＋１個の特徴点が出力される（ｎ＋１個以上のの特徴点が得られる）。しかし、注目ストロークからｎ個のベクトルを抽出する場合、特徴点がｎ＋１個より多い場合には、その分の特徴点を除去することが望まれる。そこで、特徴点消去部１０４では、図６に示すフローチャートに従って処理することで、余分な特徴点を消去した。

【００３１】

先ず、ステップＳ６０１では特徴点の個数チェックを行い、特徴点の個数がｎ＋１個となった時点で終了する。特徴点の個数がｎ＋１個より多い場合は、ステップＳ６０２にて、隣接する特徴点の角度変化が最も少ない点は特徴点から外す処理を行う。この処理は特徴点抽出処理で用いている手法を利用しても良い。

【００３２】

以上の結果、特徴点消去部１０４からは、ストロークデータで表される軌跡の曲率が大きい付近の点は最終的に残り、曲率が小さい部分の点は消去され、最終的にｎ個のベクトルデータを生成するためのデータのみが生成される。

【００３３】

次に、ベクトル化部１０５の処理動作を図７のフローチャートに従って説明する。

【００３４】

ステップＳ７０１にてカウンタｉを“０”で初期化する。次いで、ステップＳ７０２ではｎ＋１個の特徴点すべてに対してベクトル処理したかどうかのチェックをしている。このチェック結果が否の場合には、以下のステップＳ７０３～７０５の処理を行う。

【００３５】

先ず、ステップＳ７０３では隣接する特徴点２点を通る直線の方向ベクトルを求める。方向ベクトルは図８に示す８方向ベクトルとする。求められた方向ベクトルはステップＳ７０４でベクトル値として保存される。ステップＳ７０５にて次の特徴点に移る処理を行ない、ステップＳ０２に戻る。

【００３６】

こうして、全ての特徴点に対してベクトル値が求められると、ステップＳ７０６にてｎ本のベクトル値が生成される。

【００３７】

文字認識部１０６は１つの文字入力に対する全ストロークの入力が完了した段階で、ベク

トル化部１０５から送られてきた全ベクトルデータに基づき文字認識処理を行なう。すなわち、入力した各ストロークのベクトルデータと辞書部１０８内の標準的なベクトルデータとを比較し、最も類似するとされる文字から順番に第１候補、第２候補、…とし、最終結果として結果出力部１０７に送る。

【００３８】

尚、説明が前後するが、入力されたストロークデータを構成している時系列な座標データがほぼ直線上に並んでいる場合（例えば直線もしくはそれに近い線を手書き入力した場合）、両端点は特徴点として処理されるものの、中間に位置する点の中では特徴点を特定することが難しい。しかし、ベクトル化部１０５から出力される注目ストロークに対するベクトルデータ（図８参照）としては、同じベクトルコードが連続するものとなるわけであるから、どの点を特徴点としても結果は同じである。従って、図５の処理では、１つの閾値角度θｔ（ほとんど０°に近い値）で一挙に多くの特徴点が抽出され、図６の処理に渡されるがステップＳ６０２では同じ角度のものが複数あった場合には、単純にいずれか１つを特徴点として決定すれば対処できるので、何等問題はない。

【００３９】

また、上記実施形態において、結果出力部１０７は、文字認識結果として、下位の処理部に渡す処理を行う。下位の処理としては、例えば文書処理等がある。要は、文字認識処理して得られた結果に基づいて処理するものであれば如何なるものであっても良い。

【００４０】

以上説明した様に、曲がりの程度が大きい、或いは曲がりの程度が大きい部分を有するストロークのデータ（座標データ群）については、その曲率が大きい付近の座標データを積極的に有効な点として処理するので、例えば平かなの「る」と入力した場合には最後の部分はもとより、ベクトルの向きが大きい部分に関しては確実に保持されるので、手書き文字の精度良い特徴ベクトルを常に得ることができ、結果として認識率を上げることが可能になる。

【００４１】

なお、本実施形態では、特徴抽出処理にベクトル角度を用いて特徴点を抽出したが、角度に限らず、ある一定距離内の座標点を消去して座標点を間引く処理を使用、併用してもかまわない。

【００４２】

＜第２の実施の形態＞

上述した実施形態（第１の実施形態）ではストロークの特徴を抽出する抽出処理を用いてｎ分割する手法を説明したが、本第２の実施形態ではストローク内のある範囲内における座標点密度を計算し、この密度に応じて分割数を変化させてｎ分割する手法を説明する。

【００４３】

図９は本第２の実施形態におけるオンライン手書き文字認識方法の機能構成を示すブロック図である。図９において９０１、９０２、９０６、９０７、９０８はそれぞれ、図１の１０１、１０２、１０６、１０７、１０８とほぼ同等の処理をするので、その動作の説明は省略する。

【００４４】

異なるのは、点密度計算部９０３を新たに設けた点である。以下、この点密度計算部９０３の処理内容を図１０のフローチャートに従って説明する。

【００４５】

まず、ステップＳ１００１では、ストロークの全長を計算する。ストロークの全長算出はストロークを構成する隣接座標二点間のユークリッド距離を求め、始点座標から開始して終点座標まで加算していくことで得られる。そっして、求まった全長をｎ等分し、その値をｌとする。

【００４６】

ステップＳ１００２でストローク上に長さｌを持つ座標点をマッピングし、この座標点をｎ分割点Ｐｉ（０≦ｉ≦ｎ＋１）と呼ぶことにする。ここで言う、「ストローク上」とは

座標点を結んでできる直線のことを指している（図１１参照）。
【００４７】
次にステップＳ１００３で第１のｎ分割点Ｐｉと第２のｎ分割点Ｐｉ+1（０≦ｉ≦ｎ＋１）の区間をＲｊ（０＜＝ｊ＜＝ｎ）とし、区間Ｒｊ内の座標点数をカウントする。カウントされた座標点数が点密度となる。いま、あるストロークの場合の４分割（ｎが４）の分割の様子と、区間別の点密度を図１２、図１３で示す。
【００４８】
次に分割点決定部９０４の動作を図１４を用いて説明する。
【００４９】
先ず、ステップＳ１４０１にて、点密度計算部で得られた各区間から点密度（全体の点数に対する区間の点数）が最大のものを探し出す。ステップＳ１４０２で、その区間の点密度が予め定められたしきい値より大きければステップＳ１４０３に進み、閾値以下の場合はその処理を終える。
【００５０】
さて、処理がステップＳ１４０３に進んだ場合には、点密度が閾値以下になるように再分割する。例えば、その区間の長さを求めて２等分し、新たな区間を作成して区間の点密度をそれぞれ算出する。点密度の算出方法は点密度算出部と同様の処理とする。この処理で分割数が増えたので（ｎ＋１個の区間が発生したので）、ステップＳ１４０４では点密度が最小の区間を探し出し、ステップＳ１４０５でそれに隣接する区間（両側にある場合には、点密度の低い区間）と合併する。また、合併した両区間を元々分割していたｎ分割点は削除する。これによって、再び注目ストロークはｎ分割状態に戻す。また、合併後の点密度は両区間の点密度平均を算出して求める。以下、このような動作を繰り返し、処理が終了すると、新しく生成されたｎ分割点が求まる。そして、分割点の両端にある点座標をベクトルとして有効な座標データとして出力する。この結果、人間が無意識にストロークの方向が変化する部分で筆記速度が遅くなり、その部分で多くの座標点が検出されることに対処し、精度良く分割することが可能になる。これ以降、求まったｎ分割点を用いてベクトル化し、第１の実施形態と同様の処理を行う。
【００５１】
＜第３の実施の形態＞
上記第１、第２の実施形態では、図１或いは図９に示す構成を有する装置で実現させたが、本発明はこれに限るものではなく、プログラムでもって実現させても良い。
【００５２】
図１５に第３の実施形態における装置のブロック構成を示す。図示において、１は装置全体の制御を司るＣＰＵ、２はブートプログラム等を記憶しているＲＯＭ、３はＣＰＵ１が処理するプログラムやデータを記憶するＲＡＭである。４は座標入力板であって、５は座標入力板４上で操作者による不図示の入力ペンでなぞった座標を検出する座標検出部である。尚、入力ペンには、座標入力板４に対して押下したか否かを検出するためのスイッチが設けられており、このスイッチの押下状況はＣＰＵ１に出力されるようになっている。６は、座標入力板４の下面に設けられた液晶表示器（ＬＣＤ）であって、７は液晶表示器６の駆動を制御するＬＣＤドライバである。
【００５３】
ここで、ＣＰＵ１は、ペンダウンを検出している間に、座標検出部５から順次入力される点座標データを線分で結ぶようＬＣＤドライバ７を制御することで、あたかも紙上に文字を書くかのごとく、ＬＣＤ６にはその筆跡を表示することが可能になっている。
【００５４】
８は例えばフロッピーディスクやハードディスク装置等の外部記憶装置であって、オペレーティングシステム、各種アプリケーションプログラムをはじめ、本実施形態にかかる文字認識処理にかかるプログラムを記憶している。そして、認識処理するときに参照される文字認識辞書や認識結果（文字コード）を記憶する領域も確保されている。
【００５５】

上記構成において、ＣＰＵ１は文字認識を行う場合には、文字認識にかかるプログラムをＲＡＭ３にロードし、そのロードしたプログラムに従って文字認識処理を行うことになる。

【００５６】

外部記憶装置８に記憶されている文字認識プログラムの例を図１６に示す。尚、図示では記憶媒体１６０を示しており、上記実施形態における外部記憶装置８に適合するものであるが、具体的には、フロッピーディスク、ハードディスク、ＣＤ－ＲＯＭ等情報を記憶する媒体であれば如何なる物であっても良いのは勿論である。

【００５７】

図示において、１６１は、入力したストロークの点データ群のベクトル方向の変化の度合を検出する検出モジュールであり、１６２は、検出されたベクトル方向の変化が所定の閾値と比較する比較モジュール、１６３は比較結果によりベクトル方向の変化が前記閾値以上の場合、当該ベクトルを構成する一部の点データを有効とし、所定変化以下の場合には当該ベクトルを構成する点データを構成する一部の点データを除去する制御モジュールである。

【００５８】

ここで、更に、記憶媒体には、更に、制御モジュールで得られた有効な点データの個数が、注目ストロークに関してｎ個の連続したベクトル列を特定し得るだけの数になったか否かを判別する判別モジュールと、ｎ個の連続したベクトル列を特定し得るだけの数が得られないと判別した場合、閾値を調整して、検出モジュール、比較モジュール、及び制御モジュールを再度実行する第２の制御モジュールを備えることが望ましい。

【００５９】

また、更に、記憶媒体は、第２の制御モジュールでもって調整された閾値に基づいて得られた有効な点データの数がｎ＋１個以上あると判別モジュールが判別した場合、得られた特徴点のベクトル変化の大きいものから上位のｎ＋１個の点データを抽出するモジュールを備えることが望ましい。

【００６０】

また、更に、下位処理として文字認識処理モジュールを備えることが望ましい。

【００６１】

以上の記憶媒体１６０は第１の実施形態に対応するものであるが、第２の実施形態に適応するものであれば、入力したストロークの点データ群の密度分布を検出する点密度検出モジュールと、得られた点密度の高低に基づいて区画を設定する区画設定モジュール、設定された各区画の両端点を特徴点として抽出する抽出モジュールを備えることになるのは、上記例からすれな容易に想到できよう。

【００６２】

この場合、点密度検出モジュールには、入力されたストロークの時系列の点座標データで表されるストローク長をｎ等分する分割モジュールと、分割した各区画に含まれる座標点の個数を計数する計数モジュールとを含むことが望ましい。

【００６３】

また、区画設定モジュールには、点密度検出モジュールで検出された各区画のうちの最大点密度と所定密度とを比較する比較モジュールと、最大点密度が所定密度以上であるとした場合、当該最大点密度の区画を、当該区画長で２等分することでｎ＋１個の区画を生成する第１の区画補正モジュールと、生成されたｎ＋１個の区画のうち、最低点密度の区画とその隣接する区画とを結合してｎ個の区画を再設定すると共に比較モジュールを再度実行する第２の区画補正モジュールを備えることが望ましい。

【００６４】

また、この場合にも、文字認識装置として機能させるためには、下位処理としての文字認識モジュールを備えることが望ましい。

【００６５】

以上説明した様に、本実施形態によれば、オンライン手書き文字を認識する場合に、入力

されたストロークデータから精度の良いベクトルデータを抽出することが可能になる。

【００６６】

尚、実施形態では、文字入力部を介して直接手書き情報が入力されるものとして説明したが、手書き情報を記憶した記憶媒体から、或いは、通信を介して手書き情報を入力しても良いので、上記実施形態に限定されるものではない。

【００６７】

また、上記第３実施形態で説明した様に、本発明は、入力したストローク情報から特徴点を抽出して、その特徴点に対応する情報を下位の処理に渡す技術に関するものであり、下位処理としてはどのようなものであっても構わない。しかしながら、下位処理として文字認識処理を採用した場合には、その認識率を向上させることが可能なので、文字認識処理に適応するのが望ましい。

【００６８】

また、上記の如く、本発明は、複数の機器から構成されるシステムに適用しても、１つの機器からなる装置に適用してもよい。また、本発明はシステム或は装置にプログラムを供給することによって実施される場合にも適用できることは言うまでもない。この場合、本発明にかかるプログラムを格納した記憶媒体が、本発明を構成することになる。そして、該記憶媒体からそのプログラムをシステム或は装置に読み出すことによって、そのシステム或は装置が、予め定められたし方で動作する。

【００６９】

【発明の効果】

以上説明したように本発明によれば、入力された手書きストローク情報から、当該ストロークの精度良い特徴情報を抽出して下位の処理に渡すことが可能になる。

【００７０】

【図面の簡単な説明】

【図１】本実施形態におけるオンライン手書き文字認識方法の機能構成を示すブロック図である。

【図２】本実施形態における記入されたストロークの例を示す図である。

【図３】第１の実施形態における特徴抽出処理を示すフローチャートである。

【図４】特徴抽出処理の原理を説明するための図である。

【図５】第１の実施形態における特徴点数の制御を示すフローチャートである。

【図６】第１の実施形態における特徴点消去部の動作を示すフローチャートである。

【図７】本実施形態におけるベクトル化部の動作を示すフローチャートである。

【図８】本実施形態における方向ベクトルを示す図である。

【図９】第２の実施形態におけるオンライン手書き文字認識装置の機能ブロック図である。

【図１０】第２の実施形態における点密度計算部の動作を示すフローチャートである。

【図１１】第２の実施形態におけるｎ分割点を示す図である。

【図１２】第２の実施形態におけるストローク分割の様子を示す図である。

【図１３】第２の実施形態におけるストローク分割後の点密度を示す図である。

【図１４】第２の実施形態における分割点決定部の動作を示すフローチャートである。

【図１５】第３の実施形態における具体的な装置構成を示すブロック図である。

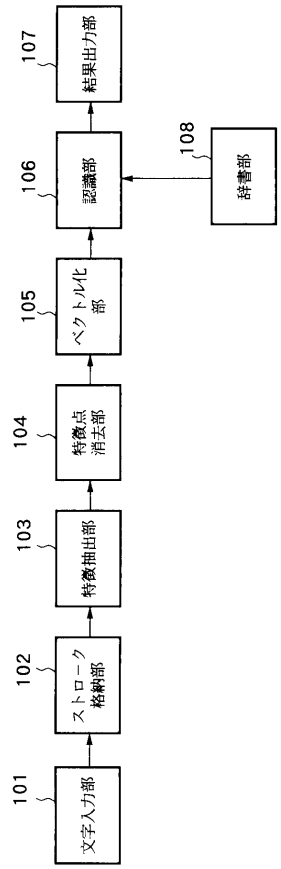
【図１６】第３の実施形態における記憶媒体に記憶されているプログラムモジュールを示す図である。

【符号の説明】

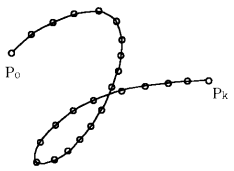
１０１、９０１　文字入力部
１０２、９０２　ストローク格納部
１０３　特徴抽出部
１０４　特徴点消去部
１０５、９０５　ベクトル化部
１０６、９０６　認識部

１０７、９０７　結果出力部
１０８、９０８　辞書部
９０３　点密度計算部
９０４　分割点決定部

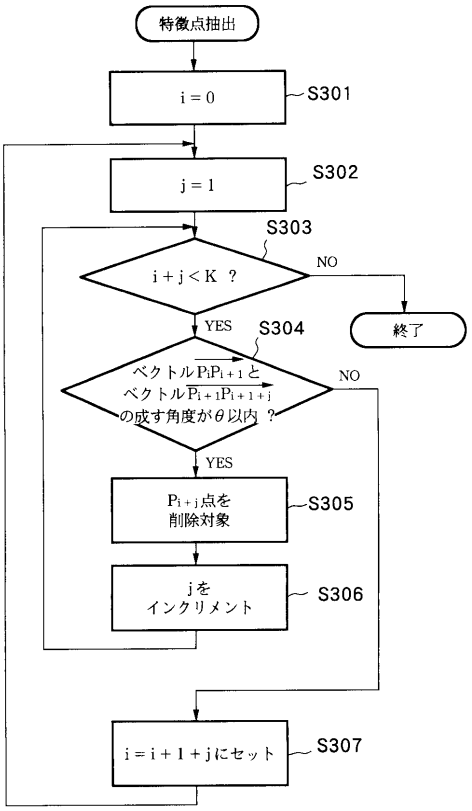
【図１】



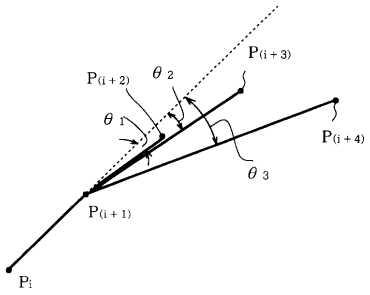
【図２】



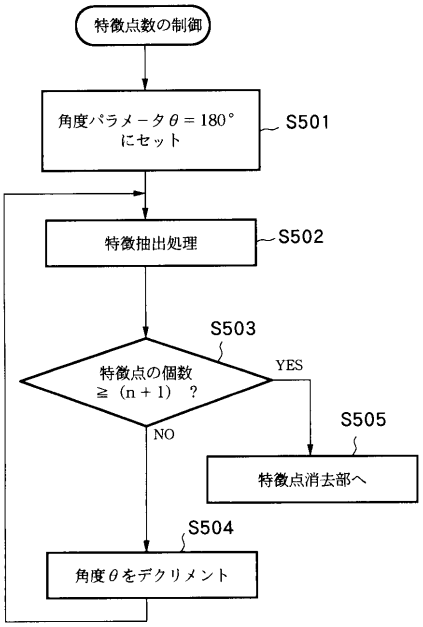
【図３】


特徴点抽出
i = 0
S301
j = 1
S302
S303
i + j < K ？
NO
終了
YES
S304
ベクトル $\overrightarrow{P_iP_{i+1}}$ と
ベクトル $\overrightarrow{P_{i+1}P_{i+1+j}}$
の成す角度が θ 以内 ？
NO
YES
P_{i+j} 点を
削除対象
S305
j を
インクリメント
S306
i = i + 1 + j にセット
S307

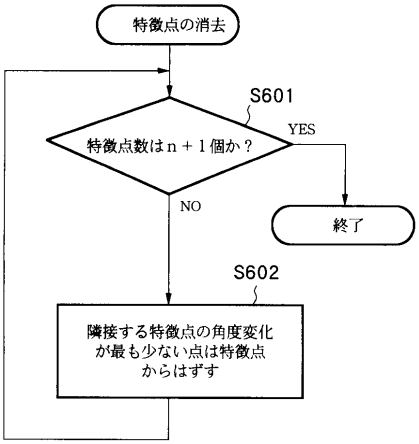
【図４】


$P_{(i+3)}$
$P_{(i+2)}$
θ 2
θ 1
$P_{(i+4)}$
θ 3
$P_{(i+1)}$
P_i

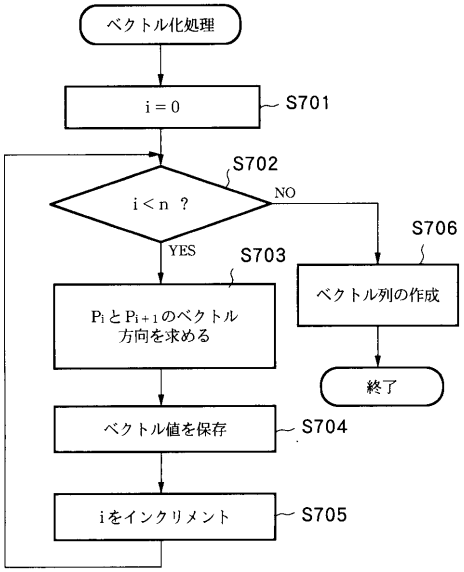
【図５】


特徴点数の制御
角度パラメータ θ = 180°
にセット
S501
特徴抽出処理
S502
S503
特徴点の個数
≧ (n + 1) ？
YES
NO
S505
特徴点消去部へ
S504
角度 θ をデクリメント

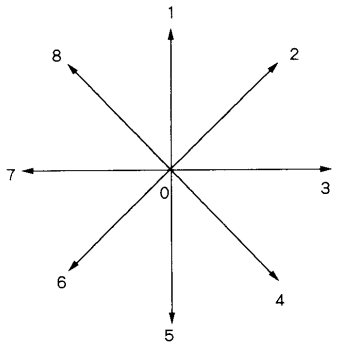
【図６】


特徴点の消去
S601
特徴点数は n + 1 個か？
YES
NO
終了
S602
隣接する特徴点の角度変化
が最も少ない点は特徴点
からはずす

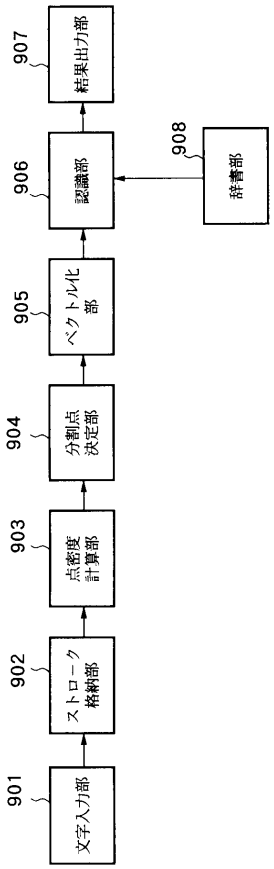
【図７】


ベクトル化処理
i = 0
S701
S702
i < n ?
NO
YES
S706
S703
Pi と Pi+1 のベクトル
方向を求める
ベクトル列の作成
終了
ベクトル値を保存
S704
iをインクリメント
S705

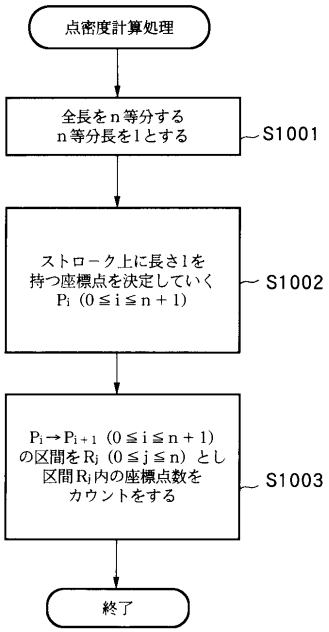
【図８】


1
2
3
4
5
6
7
8
0

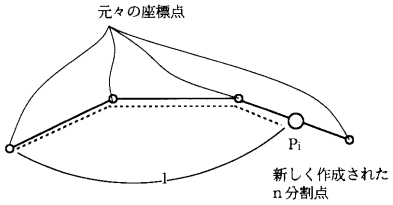
【図９】


907
結果出力部
908
辞書部
906
認識部
905
ベクトル化部
904
分割点決定部
903
点密度計算部
902
ストローク格納部
901
文字入力部

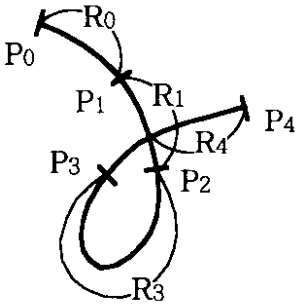
【図１０】


点密度計算処理
全長をn等分する
n等分長をlとする
S1001
ストローク上に長さlを
持つ座標点を決定していく
Pi (0 ≦ i ≦ n + 1)
S1002
Pi→Pi+1 (0 ≦ i ≦ n + 1)
の区間をRj (0 ≦ j ≦ n) とし
区間Rj内の座標点数を
カウントをする
S1003
終了

【図１１】



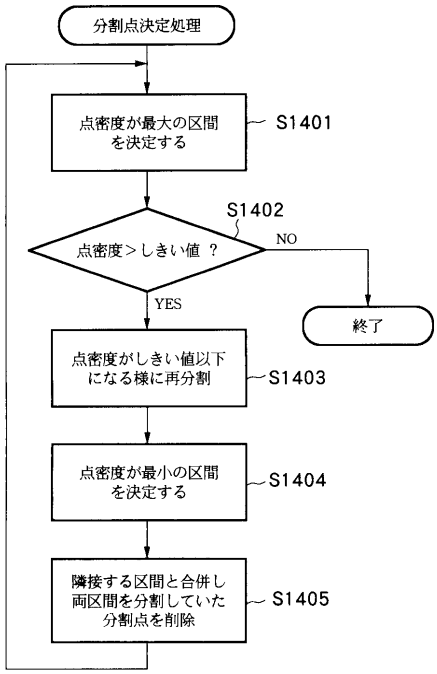
【図１２】



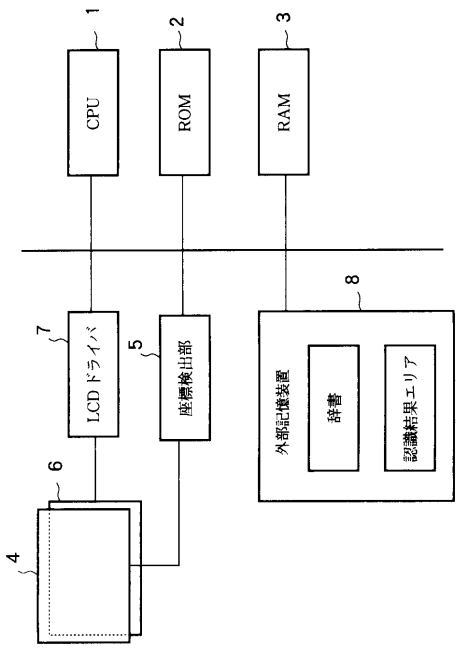
【図１３】

n = 4	R_0	R_1	R_2	R_3
点密度	15	15	20	30

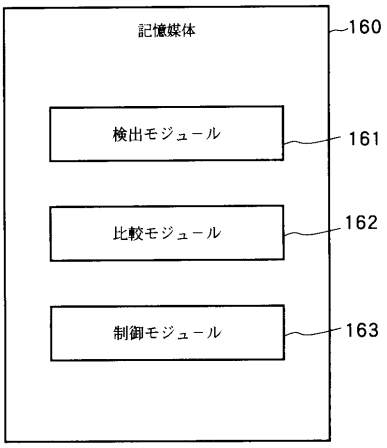
【図１４】



【図１５】



【図１６】



フロントページの続き

(72)発明者　吉井　裕人
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　佐藤　実

(56)参考文献　特開平０２－３１０７８３（ＪＰ，Ａ）
特開平０７－１６０８７０（ＪＰ，Ａ）
特開昭６２－１９０５７８（ＪＰ，Ａ）
特開昭６２－１５４０７８（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
G06K 9/00～9/76
G06T 3/00