(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第3653555号
(P3653555)
(45)発行日　平成17年5月25日(2005.5.25)　(24)登録日　平成17年3月11日(2005.3.11)

(51)Int.Cl.[7]　F I
G06T　7/00　GO6T　7/00

請求項の数 2　(全 14 頁)

(21)出願番号　特願平7-88302
(22)出願日　平成7年4月13日(1995.4.13)
(65)公開番号　特開平8-287250
(43)公開日　平成8年11月1日(1996.11.1)
審査請求日　平成14年3月28日(2002.3.28)

特許法第３０条第１項適用　電子情報通信学会

(73)特許権者　000001432
グローリー工業株式会社
兵庫県姫路市下手野１丁目３番１号
(74)代理人　100071054
弁理士　木村　高久
(72)発明者　亀山　博史
兵庫県姫路市下手野一丁目３番１号　グローリー工業株式会社内

審査官　松尾　俊介

(56)参考文献　特開昭６３－２９９４１３（ＪＰ，Ａ）
特開昭６３－２９８５２３（ＪＰ，Ａ）
特開昭６４－　７２２９４（ＪＰ，Ａ）

最終頁に続く

(54)【発明の名称】パターン認識方法

(57)【特許請求の範囲】
【請求項１】
２分木の各ノードにあらかじめ登録された辞書パターンと入力されたパターンとの距離を深さ優先探索によって計算し、最も距離値の小さいｋ個の辞書パターンのもつカテゴリーに基づいて、パターンの識別を行う方法において、
２分木の各ノードに辞書パターンを格納し、
左部分木の探索へ進むか進まないかをＬルール
ｄＬ－αＲＬ＜｜ｚ－q_k'｜が真ならば、左部分木を探索する
右部分木の探索へ進むか進まないかをＲルール
ｄＲ－αＲＲ＜｜ｚ－q_k'｜が真ならば、右部分木を探索する
０＜α＜１
ｄＬ（Ｒ）：左（右）子ノードに格納されたパターンとｚとの距離
ｚ　　：入力パターン
q_k'　　：探索途中での第ｋ番目の準最近傍点
ＲＬ，ＲＲ：パターンの勢力範囲
を適用するようにしたことを特徴とするパターン認識方法。
【請求項２】
前記ルールにおいて、αは０．５～０．６であることを特徴とする請求項１記載のパターン認識方法。
【発明の詳細な説明】

【０００１】
【産業上の利用分野】
本発明は、パターン認識方法に係り、特にパターン認識を高精度かつ高速化するための木構造に関する。
【０００２】
【従来技術】
従来、ｋ最近傍識別という方法が提案されている。この方法は、サンプルパターンの集合から入力パターンに最も近いものからｋ個を選び、それらのもつラベルに基づいて、入力パターンの分類を決定するという方法である。この方法はサンプルパターンの個数が十分である場合には、どんな分布のパターンにも適用でき高い識別能力を有することが知られている。またこの方法はパターンの確率分布に関する知識を必要としないノンパラメトリックな方法であり、また誤り確率がベイズ決定に近いという利点もある。
【０００３】
しかしながらｋ個の最近傍パターンを求めるには計算時間がかかるという問題がある。
【０００４】
そこで、階層的に分類されたサンプルパターンを木構造として表現しておき、木の探索を行ってサンプルパターンから入力パターンに近いものを選択し、選択されたパターンのコードでコード化（識別）する方法が提案されている（特開平４－２４８７２２号）。この方法では、木のルートよりはじめて、その子ノードに対応するパターンと、入力パターンの距離が最も近い子ノードを選んで進み、終端ノードまで繰り返したときに、最も距離の小さいパターンを選択するという方法がとられるため、サンプルパターン数のログのオーダーの距離計算回数で済み、非常な高速処理が可能である。
【０００５】
しかしながら、全探索とは異なり、探索されたパターンがすべてのパターンの内で最も近いという保証はなく、性能がよくないという問題がある。
【０００６】
またサンプルパターンの収容された２分木をルートから始めてルートに帰ってくるまで深さ優先順に全探索に等価な探索を行うことによって、最近傍パターンを見つける方法が提案されている（カランタリとマクドナルドの方法）。この方法では図１８に示すように、各ノードには各ノードのさすパターンから左右の部分木に収容された全パターンまでの距離の最大値（勢力半径）が付与されており、この勢力半径を利用して部分木に最小距離パターンが存在しないかどうかをチェックするルールを適用し、存在しないと判定されたときにはその部分木の探索を省略することによって、探索時間の高速化をはかることができる。
この方法では、図１９に示すように、データＰＬとｘとの距離を距離の公理を満たす関数（実際にはユークリッド距離を利用した）としてｄ（ＰＬ，ｘ）としたとき、ｄ（ｘ，ｙ′）＋L_{radius}≦ｄ（ＰＬ，ｘ）のとき、左部分木は探索しない（ルール１）。L_{radius}は勢力範囲、この図から半径L_{radius}の球の内部にはｙ′よりもｘに近いデータは存在しないことは距離の公理より明らかである。ここで、L_{radius}あるいはR_{radius}を小さくすることができれば、ルール１はより有効にはたらき、高速に探索することができる。
【０００７】
しかしながらこの方法では、文字認識に用いる特徴パターンの次元は数十次元以上と高く、高次元ではルールが有効に働かなくなり、探索の効率が悪いという問題がある。
【０００８】
これはパターン空間の広がり（体積）は次元数の指数関数として増加しているが、登録するパターンの個数は現実的には記憶容量のコストの制限からそのように大幅に増加させることはできない。このため高次元ではパターン空間は非常にスパースになっており、ルール１の探索途中での最近傍点までの距離ｄ（ｘ，ｙ′）はスパースな空間では平均的に大きいので、ルール１の不等式は成立しにくいと考えられるためである。２５６レベルで１次元を表現したとき、ｎ次元空間の体積は、256^nである。

【０００９】
【発明が解決しようとする課題】
このように、各ノードにデータと左右の部分木の勢力半径を付与した２分木を探索することによって入力データと最も近いデータをみつけるアルゴリズムの提案がなされているが、これらは手書き数字や、英文文字、かたかなに適用したところ十分な照合回数の削減を行うことはできないという問題があった。
【００１０】
本発明は前記実情に鑑みてなされたもので、手書き文字に対しても高精度を維持しつつ、高速で信頼性の高い認識を行うことのできる方法を提供することを目的とする。
【００１１】
【課題を解決するための手段】
そこで本発明では、あらかじめ登録された辞書パターンと入力されたパターンとの距離を計算し、最も距離値の小さいｋ個の辞書パターンのもつカテゴリーに基づいて、識別を行う方法において、各ノードに辞書パターンが格納された２分木を用い、部分木探索のルールを高次パターン空間の幾何学的構造に基づいて近似的に最近傍点探索を行うことができるように修正することによって照合回数を削減したことを特徴とする。
【００１２】
すなわち、ルートノードより始めて再びルートに戻るまで、以下を再帰的に行う。入力パターンｚと左右のノードに登録されたパターンとの距離ｄＬ，ｄＲを計算し，探索途中でのｋ番目までの準最近傍点q_i'（i=1,2,……k）を更新する。
【００１３】
ｄＬ＜ｄＲのとき、次の２つのルールをＬルールを先に、次にＲルールをと順次適用する。そうでないときはＲルール、Ｌルールの順に適用する。
【００１４】
Ｌルール
$dL-\alpha RL<|z-q_k'|$が真ならば、左部分木を探索する。
【００１５】
Ｒルール
$dR-\alpha RR<|z-q_k'|$が真ならば、右部分木を探索する。
【００１６】
ここでαは０より大きくαは１より小さいものとする。
【００１７】
ＲＬ，ＲＲ：パターンの勢力範囲である。
【００１８】
望ましくはこのαは０．５～０．６とする。
【００１９】
【作用】
すなわち、従来の最近傍点の探索は、次のようにして行われている。図１９に示すように、データＰＬとｘとの距離をｄ（ＰＬ，ｘ）としたとき、ｄ（ｘ，ｙ′）＋L_{radius}≦ｄ（ＰＬ，ｘ）のとき、左部分木は探索しない（ルール１）。L_{radius}は勢力範囲、この図から半径L_{radius}の球の内部にはｙ′よりもｘに近いデータは存在しないことは距離の公理より明らかである。
【００２０】
ここで、L_{radius}あるいはR_{radius}を小さくすることができれば、ルール１はより有効にはたらき、高速に探索することができる。
【００２１】
本発明では、ルール２を考える。これは、図１に示すように、ｄ（ｘ，ｙ′）＋αL_{radius}≦ｄ（ＰＬ，ｘ）のとき、左部分木は探索しない。０＜α＜１というものである。
【００２２】
ルール１が成立しないときにルール２が適用されたとき、その部分木の内に、ｙ′より近

いものがみつかる確率はαが１に近いときは低い。

【００２３】

ｙ′よりも近い点が存在する領域は図１に斜線で示した２つの球が重なる部分Ｃである。点ｘを中心とする球と点ＰＬと点ｘを結ぶ線分が交わる点をＣｐ とする。円はｎ次元超球、直線ＣｐＣｔはｎ次元超平面を示す。高次元空間では、Ｃｐが球の外周に近いところにあるとき（α＝ＣｐＰＬ／L_{radius}が１に近いとき）、Ｃの体積V_c は極めて小さい。直線ＸＰＬに垂直でＣｐを通る平面で球ＰＬを２分割したときのＣを含む方の領域Ｃ１の体積V_{C1}は

$$V_{C1}=\frac{\sqrt{\pi}^{n-1}}{\Gamma\left(\frac{n+1}{2}\right)}\int_{\alpha R}^{R}\left(R^2-x^2\right)^{\frac{(n-1)}{2}}dx=\frac{\sqrt{\pi}^{n-1}R^n}{\Gamma\left(\frac{n+1}{2}\right)}\int_0^{\theta_1}\sin^n\theta d\theta$$

ここでｎは空間の次元、Ｔ＝L_{radius}，Γ（ｔ）はガンマ関数、θ１はＸＰＬとＣｔＰＬのなす角度とする

ＰＬを中心とする球の体積は

$$V_S=\frac{\left(\sqrt{\pi}R\right)^n}{\Gamma(n/2+1)}$$

で与えられる。

【００２４】

そこでL_{radius}を半径とする球全体に対する前記２つの球が重なる部分Ｃの体積の比を知りたいが、その代わりにV_{c1}／V_s を縦軸、αを横軸にとり数値計算した結果を図２(a)および(b) に示す。ここでｎ＝１００回とし、(b) は縦軸を対数にとった場合のグラフである。αが１に近付くにつれて体積比は極めて小さくなっていくようすが、図３に示すようにｎ＝３のときに比べるとわかる。Ｃ部の体積はそれよりさらに小さい。

【００２５】

したがって、データがほぼ均一に分布していると仮定すると、ｃ部分に左部分木に属するデータが存在する確率は小さく、結局ルール１をルール２に変更しても探索ミスをすることは少ないことがわかる。

【００２６】

２分木の探索を行ったとき、ルール２が適用（厳密にはルール２の不等式がα＜１のみに成立する）された回数をｔとすると、真の最近傍点がみつかる（探索がエラーにならない）確率は、体積比を使って

（１－V_c ／V_s ）^{t} であるが

（１－V_{c1}／V_s ）^{t} ＜（１－V_c ／V_s ）^{t}

で見積もる。これは、V_{c1}＜V_c であるためである。

【００２７】

しかしながら、ｔを導き出すことは困難である。真の最近傍点がみつかる確率がｔの増加とともに小さくなる（エラーが増える）ので、最悪値はｔの最大値のときに与えられて、それは登録されたパターンの個数Ｎに等しい。

【００２８】

１００次元空間において、ｔ回ルール２が適用された場合の真の最近傍点が見付かる確率（探索率）を図４に示した。

【００２９】

αの増加に対して単調に増加する曲線が得られ、ｔが１００００回程度まで大きくても、αが０．５より大きい範囲で最近傍点が見付かる確率は１に近いことが予測できた。この図からαが０．５より大きいとき、真の最近傍点がみつかる確率は１に近く、精度を維持しつつ高速化をはかることができることがわかる。

【００３０】
【実施例】
次に、本発明の実施例について図面を参照しつつ詳細に説明する。
【００３１】
まず図５に示すような１３個の２次元パターンから２分木を作成する。図中数字はノード番号でありパターン番号でもある。ここで直線l_{12}はパターン１とパターン２との垂直２等分線（ｎ次元では垂直２等分超平面）を表す。この直線で空間は２つに分割されるが、パターン１（２）に近い点はこの直線のパターン１（２）と同じ側にある。l_{34}はl_{12}によって分割された空間をさらにパターン３と４にそれぞれ近い点が存在する領域として２分割する直線である。このように階層的に空間を分割していってできたものである。
【００３２】
まず、図６に示すようにｋ個の近傍点探索を行う。
まず被探索点ｘを入力する（ステップ１０１）。そして最短距離$NDIST_i$ ＝∞（ｉ＝１……ｋ）最近傍点q'_i（ｉ＝１……ｋ）を空にカレントノードN_c ＝ルートにセットする（ステップ１０２）。
【００３３】
そして木探索を行い、図７に示すような手続きサーチを再起的に起動する（ステップ１０３）。
【００３４】
次に手続きサーチについて説明する。
【００３５】
まずカレントノードN_c に格納された左右の部分木をL_{child}，R_{child} とし、L_{child}，R_{child} の勢力半径をL_{radius}，R_{radius}とするとともにL_{child}，R_{child} のパターンをＰＬ，ＰＲとする（ステップ２０１）。
【００３６】
そこでもし、L_{child}，R_{child} ともにＮＵＬＬであるならば、図６のステップ１０２に戻る（ステップ２０２）。
【００３７】
この後、N_c が指す左右の子ノードのパターンＰＬおよびＰＲの距離計算をする。ＤＬ＝ｄ（ｘ，ＰＬ）
ＤＲ＝ｄ（ｘ，ＰＲ）
ただし、もしＰＲがＮＵＬＬならばＤＲ＝∞とする。（ステップ２０３）
そして最短距離$NDIST_i$（ｉ＝１……ｋ）および最近傍点q'_i（ｉ＝１……ｋ）を更新する。最短距離$NDIST_i$（ｉ＝１……ｋ），ＤＬ，ＤＲをソートし、小さい順にｋ個まで値とパターン名を$NDIST_i$、q'_i に格納する（ステップ２０４）。
【００３８】
そしてＤＬ＜ＤＲであるか否かの判断をおこない（判断ステップ２０５）、ＤＬ＜ＤＲであるときには左部分木から先に探索を行う。
【００３９】
そして$NDIST_k + \alpha L_{radius} \leqq$ＤＬでなければ、カレントノードをＮｃ の左部分木$L_{child}$ に置き換えて、さらにサーチを起動する（ステップ２０６）。
【００４０】
さらに$NDIST_k + \alpha R_{radius} \leqq$ＤＲでなければ、カレントノードをＮｃ の右部分木$R_{child}$ に置き換えて、さらにサーチを起動する（ステップ２０８）。
【００４１】
一方ＤＬ＜ＤＲであるか否かの判断をおこなう判断ステップ２０５において、ＤＬ＜ＤＲでないと判断された場合には右部分木から先に探索を行う。
【００４２】
そして$NDIST_k + \alpha R_{radius} \leqq$ＤＬでなければ、カレントノードをＮｃ の右部分木$R_{child}$ に置き換えて、さらにサーチを起動する（ステップ２０７）。

【００４３】
さらに$NDIST_k + \alpha L_{radius} \leqq DL$でなければ、カレントノードをＮｃ の左部分木$L_{child}$に置き換えて、さらにサーチを起動する（ステップ２０９）。
【００４４】
このようにしたのちカレントノードを１つ前の値に戻して、手続きサーチが起動された時点に戻る（ステップ２１０）。
【００４５】
まず１回目の手続きサーチでは、ステップ２０１でL_{child}＝ノード１、R_{child}＝ノード２となる。L_{radius}はパターン１とパターン７との距離、R_{radius}はパターン１とパターン１３との距離となる。ＰＬ，ＰＲはそれぞれノード１およびノード２に格納されたパターンで図５の番号１および番号２が付された丸印で表される。
【００４６】
ステップ２０２ではL_{child}，R_{child}ともにＮＵＬＬでないのでステップ２０３に進む。
【００４７】
ステップ２０３においてＤＬはパターン１とｘとの距離、ＤＲはパターン２とｘとの距離として計算され格納される。
【００４８】
ステップ２０４において最近傍点q'_1はノード１のパターンとなる。
【００４９】
そしてステップ２０５においてＤＬ＜ＤＲであるのでステップ２０６に進む。
ステップ２０６において$NDIST_1 + \alpha L_{radius} \leqq DL$は、
$NDIST_1$はノード１に格納されたパターンとｘとの距離、L_{radius}はパターン１とパターン７との距離、ＤＬはノード１に格納されたパターンとｘとの距離であるので成立しない（図５の２つの円が重なっている）。そこで図８の矢印１のようにカレントノードN_cをノード１に換えて、さらに２回目のサーチを起動する。　２回目のサーチにおいて、ステップ２０１でL_{child}＝ノード３、R_{child}＝ノード４となる。そしてステップ２０２ではL_{child}，R_{child}ともにＮＵＬＬでないのでステップ２０３に進む。
【００５０】
ステップ２０３においてＤＬはパターン３とｘとの距離、ＤＲはパターン４とｘとの距離として計算され格納される。
【００５１】
ステップ２０４においてパターン１の方がパターン３およびパターン４よりｘに近いため、最近傍点q'_1は依然としてノード１のパターンとなる。
【００５２】
そしてステップ２０５においてＤＬ＜ＤＲであるのでステップ２０６に進む。
ステップ２０６において$NDIST_1 + \alpha L_{radius} \leqq DL$は、
$NDIST_1$はノード１に格納されたパターンとｘとの距離、L_{radius}はパターン３とパターン７との距離、ＤＬはパターン３とｘとの距離であるので成立しない。そこで、図８の矢印２のようにカレントノードN_cをノード３に換えて、さらに３回目のサーチを起動する。
【００５３】
３回目のサーチにおいて、ステップ２０１でL_{child}＝ノード７、R_{child}＝ノード８となる。そしてステップ２０２ではL_{child}、R_{child}ともにＮＵＬＬでないのでステップ２０３に進む。ステップ２０３においてＤＲ，ＤＬはそれぞれパターン７、パターン８とｘとの距離として計算され格納される。ステップ２０４において、パターン７の方がパターン１およびパターン８よりｘに近いのでq'_1はパターン７に更新される。
【００５４】
そしてステップ２０５においてＤＬ＜ＤＲであるのでステップ２０６に進む。
ステップ２０６において$NDIST_1 + \alpha L_{radius} \leqq DL$は、

$NDIST_1$ はノード７に格納されたパターンとｘとの距離、L_{radius}は０であるので等号が成立する。そこで今回のサーチは起動せずにステップ２０８に進む（図８の矢印４）。

【００５５】

ステップ２０８において、$NDIST_1$ ＋αR_{radius}≦ＤＲは、R_{radius}＝０により成立する。そこでサーチは起動せずにステップ２１０に進む

ステップ２１０でＮｃ をノード３に戻して３回目のサーチを抜けて、３回目のサーチが起動されたその直後すなわちステップ２０６の最後に戻る（図８の矢印５）。このとき最近傍点ｑ′$_1$ はノード７のパターンとなる。

【００５６】

そしてステップ２０８に進む。ここで最近傍点ｑ′$_1$ はノード７のパターン、R_{radius}はパターン４とパターン９との距離、ＤＲはノード４に格納されたパターン４とｘとの距離であるので成立しない。そこで図８の矢印６のようにカレントノードN_c をノード４に換えて、さらにサーチを起動する。

【００５７】

同様にして図８の矢印７，８，９，１０と探索をおこなっていく。矢印１１は図７のステップ２０８の処理でその不等式が成立し、$NDIST_1$ はノード９に格納されたパターンとｘとの距離、R_{radius}はパターン２と１３の距離、ＤＲはｘとパターン２との距離である。そしてサーチを起動せずにステップ２１０に進み、ステップ２１０でサーチを抜けると全部のサーチから抜けたことになり図６のメインフローでエンドとなる。

【００５８】

次にこのパターンから図８の２分木を作成する工程について説明する。

図９はそのメインフローである。まず木のルートノードをつくる（ステップ３０１）。ルートノードにパターンの登録はしない。勢力半径は∞にセットする。そしてカレントノードＮｃ をルートノードとし、サンプルパターンの集合をＳとする（ステップ３０２）。

【００５９】

そして図１０に示す木作成手続きを起動する（ステップ３０３）。

【００６０】

まず集合Ｓが空集合ならば戻る（ステップａ）。

【００６１】

Ｓがただ１個のパターンの集合であるならばこのパターンをＬノードに登録する。勢力範囲は０とする。そしてカレントノードのL_{child} がＬノードを指すようにする。R_{child}はＮＵＬＬとする。そして戻る（ステップｂ）。

【００６２】

ＳをＳ１とＳ２とに２分割し、それぞれの代表パターンＣ１，Ｃ２をノード・L_{node}とR_{node}に登録する。それぞれのノードに勢力半径を記憶する。カレントノードのL_{child} およびR_{child} がL_{node}とR_{node}を指すようにする（ステップｃ）。

【００６３】

このようにしてａ(1)，ｂ(1)，ｃ(1) と進むが、ここで(1) は１回目の処理を示す。そしてｃ(1) で、

Ｓ１＝｛Ｐ１，Ｐ３，Ｐ４，Ｐ７，Ｐ８，Ｐ９，Ｐ１０｝，Ｃ１＝Ｐ１

Ｓ２＝｛Ｐ２，Ｐ５，Ｐ６，Ｐ１１，Ｐ１２，Ｐ１３｝，Ｃ２＝Ｐ２

となる。

ここで集合ＳはＬＢＧアルゴリズムを適用し、分割された集合をＳ１′，Ｓ２′とするとともにＳ１′，Ｓ２′の重心をＣ１′，Ｃ２′とする。そしてＳのなかでＣ１′に最も近い点をＣ１とする。Ｓ－｛Ｃ１｝の中でＣ２′に最も近い点をＣ２とし、Ｃ１とＣ２を２個のクラスタ中心として再度クラスタリングを行う。このときの分割をＳ１，Ｓ２とする。

【００６４】

Ｃ１，Ｃ２をそれぞれ図８のノード１，２のデータとする。

【００６５】

WをＳ１の要素とするときの勢力半径max ｄ（Ｃ１，Ｗ）を計算してノード１の所定のメモリに書き込む。

【００６６】

そしてｄ(1) に進み、カレントノードＮｃ をＬ$_{node}$、集合Ｓ＝｛Ｐ３，Ｐ４，Ｐ７，Ｐ８，Ｐ９，Ｐ１０｝として木作成フローを起動する。（ステップｄ(1) ）　このようにしてａ(2) ，ｂ(2) ，ｃ(2) と進むが、ここで(2) は２回目の処理を示す。そしてｃ(2) で、

Ｓ１＝｛Ｐ３，Ｐ７，Ｐ８｝，Ｃ１＝Ｐ３

Ｓ２＝｛Ｐ４，Ｐ９，Ｐ１０｝，Ｃ２＝Ｐ４

となる。

Ｃ１，Ｃ２をそれぞれ図８のノード３，４のデータとし、勢力半径も計算して書き込む。

【００６７】

そしてｄ(2) に進み、カレントノードＮｃ をＬ$_{node}$、集合Ｓ＝｛Ｐ７，Ｐ８｝として木作成フローを起動する。（ステップｄ(2) ）

このようにしてａ(3) ，ｂ(3) ，ｃ(3) と進むが、ここで(3) は３回目の処理を示す。そしてｃ(3) で、

Ｓ１＝｛Ｐ７｝，Ｃ１＝Ｐ７

Ｓ２＝｛Ｐ８｝，Ｃ２＝Ｐ８

となる。

Ｃ１，Ｃ２をそれぞれ図８のノード７，８のデータとし、勢力半径はともに０とする。

【００６８】

そしてｄ(3) に進む。ここでＳは空集合として、木作成フローを起動する。（ステップｄ(3) ）

ａ(4) でただちに戻ってきてｅ(3) へと処理を進める。

【００６９】

このように再帰的に木作成フローを呼び出すことによって、パターンサンプルを階層的に分割していく。

【００７０】

ステップｅでは一方カレントノードＮｃ をＲ$_{node}$、集合Ｓ２からＰ２を除いた集合Ｓ２＝｛Ｐ２｝をＳとして木作成フローを起動する。

【００７１】

このようにして、木作成フローからすべて抜け出したとき、木の作成は終了する。

【００７２】

次に、この方法を用いた帳票読取りの一例について説明する。

【００７３】

図１１にそのフローチャートを示す。

【００７４】

まず画像入力（ステップ４０１）を行いこのデータを図１２に示すように、２値化し（ステップ４０２）、枠線位置検出および枠線除去（ステップ４０３）を行った後、輪郭追跡（ステップ４０４）を行い、図１３に示すように文字切り出し（ステップ４０５）を行う。ここで輪郭成分ｊ（ｊ＝１、２、……Ｊ）の内で、個別文字領域ｉに重心が含まれた輪郭成分を選びそれらを文字ｉを構成する輪郭データＣｉ（ｉ＝１，２，……Ｉ）として抽出する。

【００７５】

そしてｉが１からＩまで、４０６～４１１の認識処理がされたか否かを判断し（判断ステップ４０６）、Ｉまで処理されていれば認識結果を出力して終了する（ステップ４０７）。

【００７６】

一方、もしＣｉにデータがあれば（ステップ４０８）次のステップに進み輪郭データｉより方向成分特徴パターンＸｉを抽出する（ステップ４０９）。ここでは図１４(a) に示すようにメッシュ割りされた各領域の中で図１４(b) に示す４種類の方向指数のヒストグラ

ムをとる。文字パターンの位置の変動に対して安定した特徴を抽出するために、方向特徴面ごとにガウス分布に近い重みをもったマスク処理を行うことによりぼかし処理を施す。

【００７７】

またステップ４０８でＣｉにデータがなければ、認識結果は空白とし文字切り出しステップ４０５に分枝する。

【００７８】

ステップ４０９で、方向成分特徴パターンＸｉが抽出されると、木探索により参照パターンからパターンＸｉに近いｋ個のパターンを見つける（ステップ４１０）。

【００７９】

そして、選ばれたｋのパターンのカテゴリの中で、最も多いカテゴリがしきい値以上あれば、そのカテゴリーを個別文字領域ｉの文字の認識結果とする。そしてしきい値より小さければ、リジェクトとする（ステップ４１１）。

【００８０】

このようにして高精度の認識がなされる。

【００８１】

ところでこの方法を用いて１００００個のランダムデータを木に収容したときのαと照合回数との関係を測定した結果を図１５に示す。この図からα＝１すなわちカランタリとマクドナルドの方法では全く速度が変わらないことがわかる。そしてαの値と共に照合回数は単調増加することがわかる。

【００８２】

さらにこの方法において、αと探索成功率との関係を測定した結果を図１６に示す。この図から図５に示した予測どおり、αの値と共に探索成功率は単調増加することがわかる。

【００８３】

さらに、手書き数字による認識実験により、認識精度と照合回数の評価を行った。その結果を図１７に示す。（各曲線の右端のマークがα＝１．０、その左隣がα＝０．９５，０．９０，０．８５，……と０．０５刻みで０まで認識したときの一部をプロットしている）

αを１から０まで減少させたとき、照合回数は大幅に減少させることができ、しかも認識精度の低下は非常にゆるやかであった。

【００８４】

ちなみに２０７３５回の全数照合回数に対し、

α＝１．０　　９７．８０％，　２．００％，　０．２０％　　６１８９回

α＝０．４　　９７．７８％，　１．９８％，　０．２４％　　３７９回

αの値をどの程度にしたときに、認識精度の低下がゆるやかに維持されるかについては、理論的説明はできないが、０．５～０．６程度のときによいことは図１７からあきらかである。

【００８５】

なお、手書き数字、手書き英文文字、手書きかたかなについても同様の効果が確認され、認識精度と認識速度の調節を簡単に行うことが可能となる。

【００８６】

【発明の効果】

以上説明してきたように、本発明によれば、高精度を維持しつつ認識処理の高速化をはかることが可能となる。

【図面の簡単な説明】

【図１】本発明の方法を示す説明図

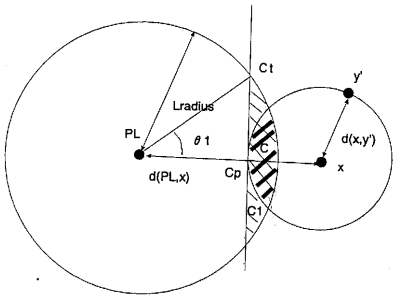
【図２】１００次元空間での切り取り部分と全球の体積比および(b) は縦軸を対数にとった場合の関係を示す図

【図３】３次元空間での切り取り部分と全球の体積比を示す図

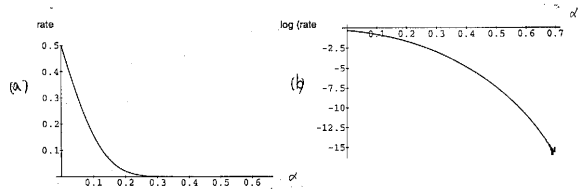
【図４】１００次元空間において、ｔ回ルール２が適用された場合の真の最近傍点が見付かる確率（探索率）を示す図

【図５】本発明実施例の１３個の２次元パターンから２分木を作成する工程を示す説明図
【図６】ｋ個の近傍点探索を行うためのフローチャートを示す図
【図７】手続きサーチを示すフローチャートを示す図
【図８】本発明の方法で得られるツリーと探索過程を示す図
【図９】２分木作成のフローチャートを示す図
【図１０】木作成手続きを示すフローチャートを示す図
【図１１】本発明の方法を用いた認識処理を示すフローチャート図
【図１２】入力画像の二値イメージを示す図
【図１３】文字切り出し後のイメージを示す図
【図１４】輪郭文字パターンの正規化とメッシュ分割を示す図および輪郭の方向指数を示す図
【図１５】αと照合回数との関係を示す図
【図１６】αと照合成功率との関係を示す図
【図１７】ｋ＝１，２，３……のときの照合回数と照合成功率との関係を示す図
【図１８】カランタリとマクドナルドの方法を示す説明図
【図１９】カランタリとマクドナルドの方法を示す説明図
【符号の説明】
L_{radius}　勢力半径
R_{radius}　勢力半径

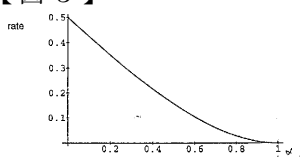
【図１】



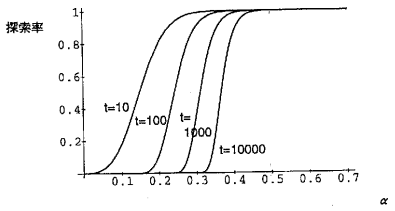
【図２】



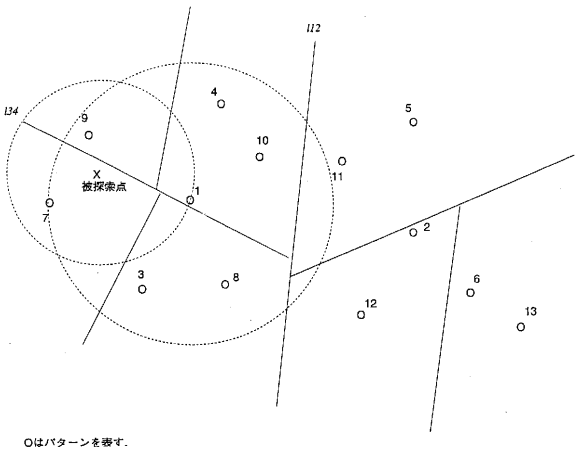
【図３】



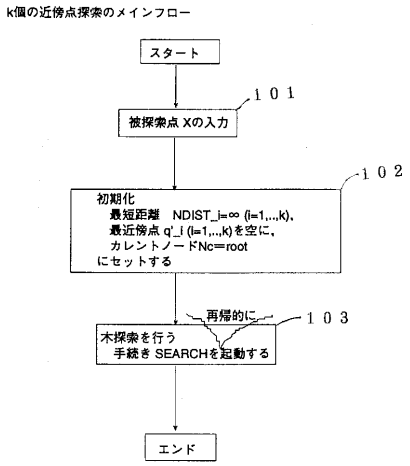
【図４】



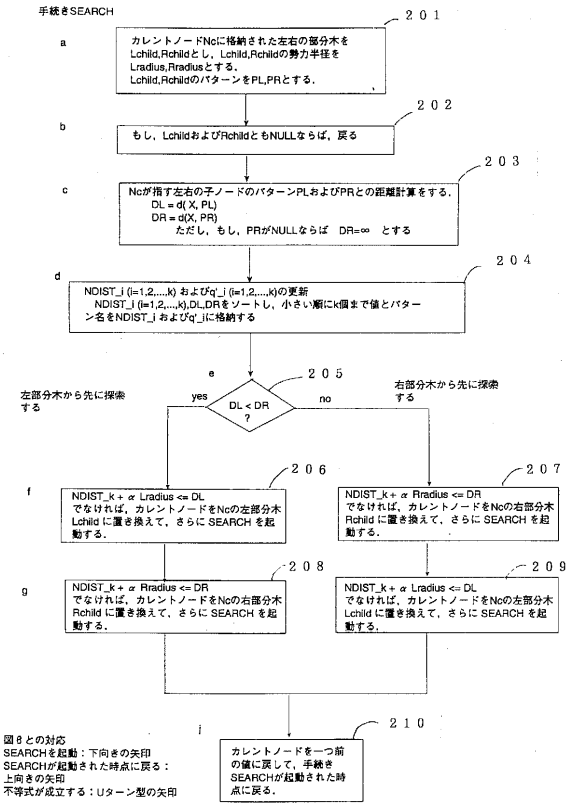
【図５】



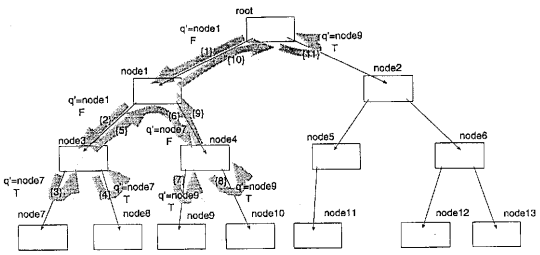
【図６】



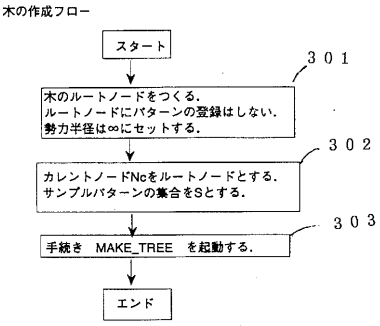
【図７】



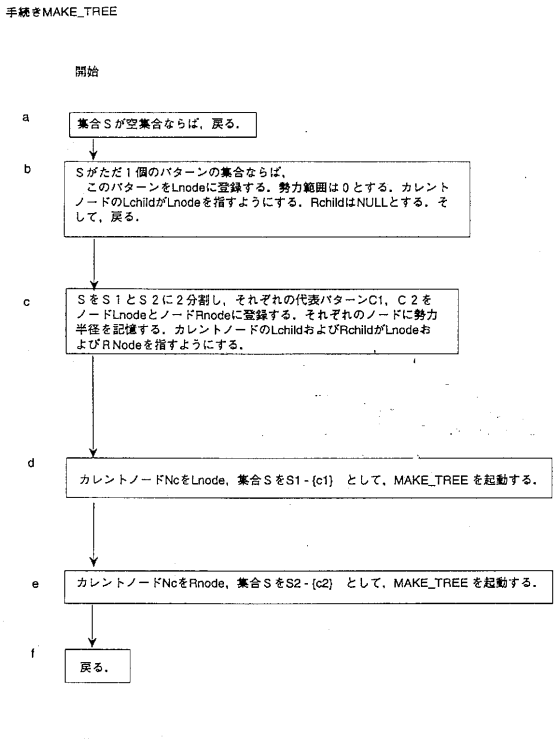
【図８】



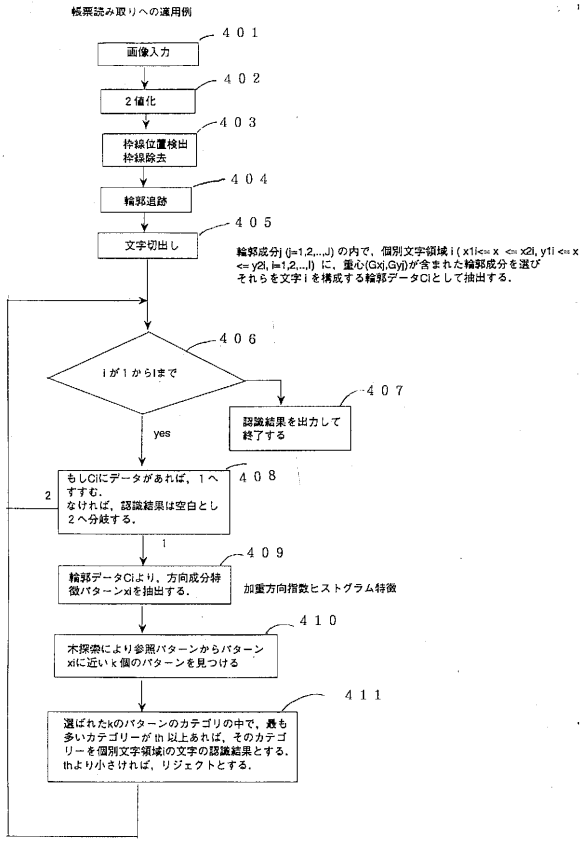
【図９】



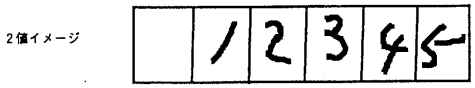
【図１０】



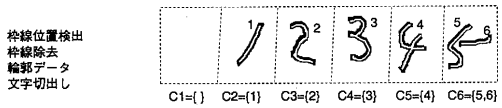
【図１１】



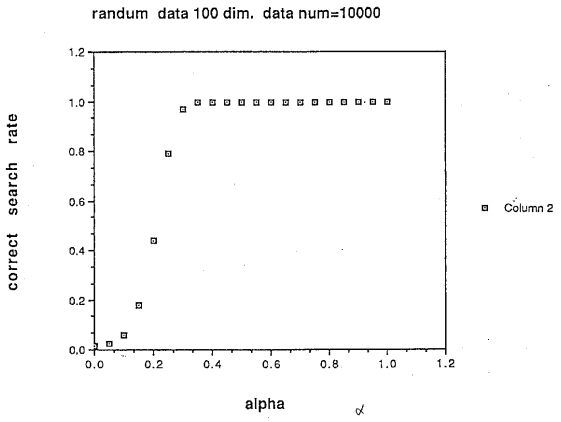
【図１２】



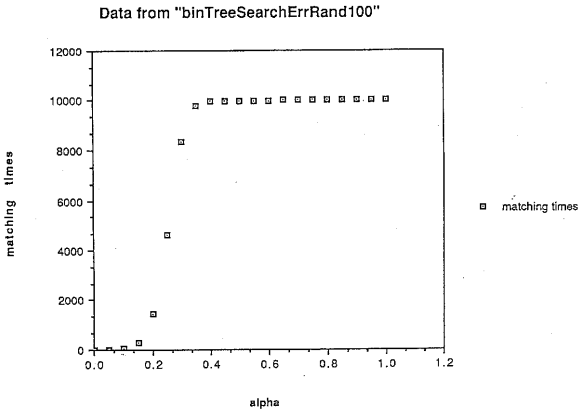
【図１３】



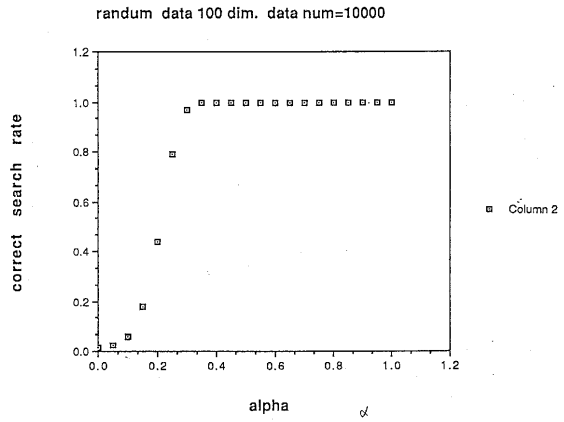
【図１４】



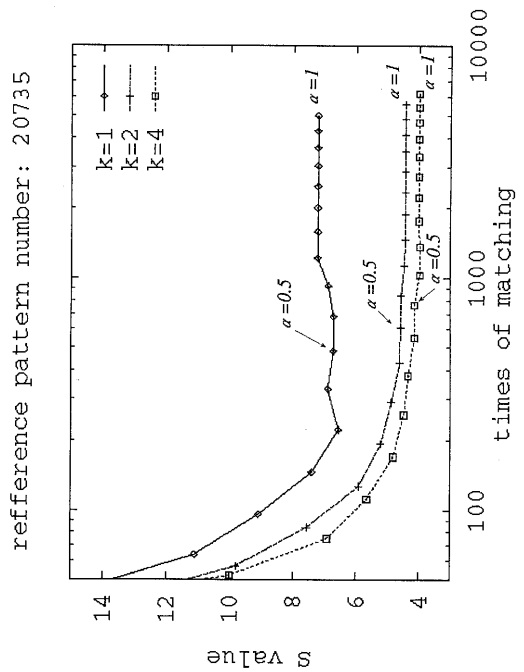
【図１５】



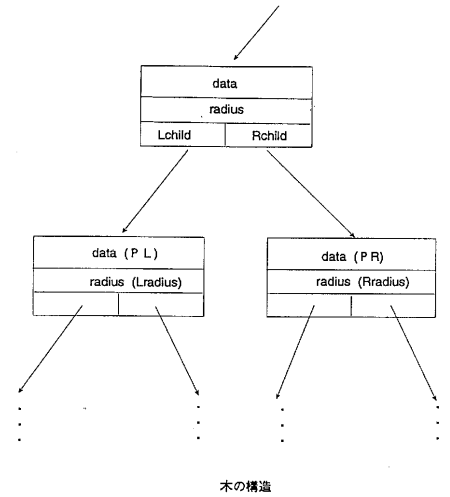
【図１６】



【図１７】



【図１８】



木の構造

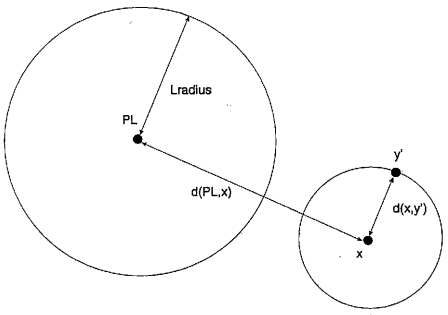
dataには、n次元のデータ(x1,x2,...,xn)およびカテゴリを格納，あるいはそのデータを指すポインタを格納する．

x1	x2	x3	・・・	xn	label

【図１９】

部分木探索の不必要性のチェック方法

Kalantari と McDonaldの文献で使用されている．



フロントページの続き

(58)調査した分野(Int.Cl.[7], ＤＢ名)
G06T 7/00