

2G-1 黒画素方向性特徴のずらしマッチングによる印刷文字認識方式の開発

岡本 卓哉 樋野 匡利  
(株)日立製作所システム開発研究所

1. はじめに

近年、各種データ、特に紙で管理されていた情報の電子化が進められており、印刷文書認識に対するニーズが増大している。こうしたことから、我々は印刷文字の特徴を生かした、効率的な文字認識アルゴリズムを検討した。

印刷文字認識のアルゴリズムを検討するにあたって考慮すべきことは、同一文書中に様々なフォント(字体、サイズ)が混在することである。したがって、フォントの多様性に対応できる、認識アルゴリズムが必要となる。フォントの違いによって、2次元的な文字パターンの変動が発生するが、認識精度の向上のためには、この変動を吸収する必要がある。

本方式は、文字パターンに適応した特徴抽出と、1次元方向のずらしにより、2次元方向のずらしを擬似的に行なう。この方式により、効率的で高精度な認識を実現する。

2. 方式の概要

印刷文字の字体の違いなどによる、文字パターンの変動は、ストロークの位置ずれ、線幅の違いが主なものであり、ストロークの方向性の変動は小さい。

このような印刷文字の持つ特徴を利用し、少ない処理量で、高精度な認識が可能なアルゴリズムを検討した。本方式では、文字パターンを方向性特徴に分解し、それぞれの方向性特徴について、特徴の方向性に垂直な方向(縦方向性特徴なら左右方向)についてのみ、線幅の変動、位置ずれを許すことで、2次元的なパターンの変動を、1次元的なずらしの組合せにより吸収することで、ずらし量の増加に伴う処理量の増加を1次元的な増加に抑えた文字認識方式となっている。

3. 特徴抽出

パターンマッチングの基となる特徴は、黒画素方向性特徴を用いた。黒画素方向性特徴の求め方は以下の通りである。

まず、文字パターン中の各黒画素について、その画素が含まれる、縦横斜めの4方向の連続した黒画素の長さ(ラン長)を求める。ある黒画素に注目したときの、縦方向のラン長をV、横方向のラン長をH、右斜め方向のラン長をR、左斜め方向のラン長をLとすると、(V, H, R, L)をベクトルと考え、式1を使って、長さ1に正規化することで、方向性

特徴(v, h, r, l)が得られる。白画素の位置の方向性特徴は、長さ0のベクトル(0, 0, 0, 0)とする。

$$(v, h, r, l) = (V, H, R, L) / \sqrt{V^2 + H^2 + R^2 + L^2} \dots \dots \text{式1}$$

図1に、ラン長抽出の原理と、画素ごとの特徴抽出で得られた各方向性特徴画像を示す。

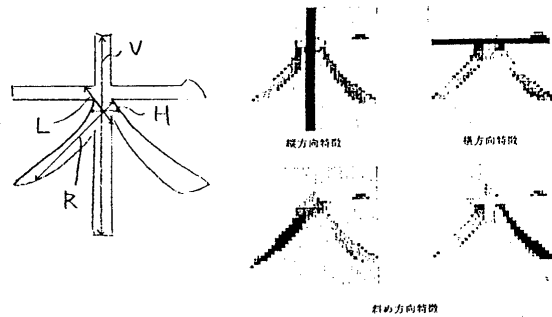


図1 方向性特徴抽出

先に求めた方向性特徴のうち、縦方向性特徴と横方向性特徴をパターンマッチングに用いる。それぞれの特徴画像に対して、一定数にメッシュ分割し、メッシュ内の画素の平均値をパターンマッチングの特徴量とする。

マッチングの処理量を小さくするためには、メッシュ分割数は、できるだけ少ない方が良い。しかし、高精度な認識のためには、細かな部分の特徴が明確になるように、メッシュ分割数は、多くした方が良い。

少ないメッシュ数で、高精度な認識を実現するためには、文字パターンに適応した特徴のボケの少ないメッシュ分割が必要となる。

本方式では、まず、図2に示すように、先に求めた方向性特徴画像に対して、注目している方向に平行な画素境界線を挟んだ特徴の差分を計算し、これを上記方向に直交する軸に加算投影する。この差分の投影値が大きな位置から順に、メッシュ分割線を設定する。このように分割線を設定することで、各メッシュ内の画素は、類似した方向性特徴の値を持つことになる。

さらに、メッシュ分割数を一定にするために、分割線を追加する。追加する分割線は、元の文字パターンのバランスを、できるだけ保存するために、均等分割位置とのずれが最も小さくなるように決定する。

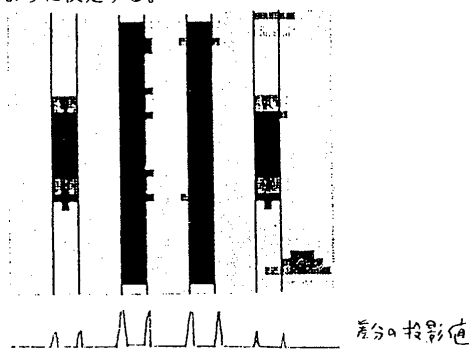


図2 メッシュ分割位置の設定

上記の方式で設定した分割線に直交する側の分割線は、マッチングの際に、ずらしを行わないことから、ボカシの効果を狙い、分割数を少なくし、均等分割とした。

特徴数は、文字のストローク数を検討した結果、縦方向性特徴については、横16×縦8の分割数とし、横方向性特徴については、横8×縦16の分割数とした。

#### 4. パターンマッチング

パターンマッチングは、少ない特徴量での絞り込みから順に細かな特徴量による詳細認識を行なう、階層的なマッチングを用いる。まず、文字の複雑度を表わす特徴で絞り込み、次に、メッシュ分割して得た特徴を投影した値を用いて、さらに絞り込む。最後に、残った文字について、メッシュ分割して得られた特徴量による詳細認識を行なう。

##### (1) 複雑度による絞り込み

複雑度は、図2に示した、差分の投影値を合計した値を用いる。文字サイズに依存しないように、この値を、図2に示した縦方向性特徴の場合は、文字の高さで割る。横方向性特徴についても同様な方法で複雑度が得られる。

絞り込みは、認識すべき文字パターンから得られた、縦、横の複雑度を、辞書の複雑度と比較し、差が閾値以内のものだけを取り出すことを行なう。

##### (2) 1次元投影値による絞り込み

メッシュ分割で得られた、縦横の特徴量の1次元投影値を用いて絞り込む。これにより、ずらしを行わない方向の位置情報を縮退した特徴量が得られる。

縦方向性特徴については、縦の1列ごとに投影し、横方向性特徴については、横の1行ごとに投影した値を求める。この投影値の並びに対して、縦、横それぞれについてDPマッチングを行ない、上位を取りだすことで絞り込みを行なう。

##### (3) 詳細認識

絞り込みで残った文字について、投影値では失われてしまう位置情報も含めて、擬似的2次元マッチングを行なう。

縦方向性特徴については、縦に並んだ1列ごとのメッシュ単位で、左右方向へのずらしマッチングを行なう。横方向性特徴についても同様に、横に並んだ1行ごとのメッシュ単位に上下方向へのずらしマッチングを行なう。ずらしマッチングには、(2)と同様にDPマッチングを用いる。(2)、(3)とも、ずらしの範囲は、2メッシュとした。

したがって、このマッチングは、図3に示した整合枠の中で、最もコストの少ない経路を探す問題になる。図3のマトリクスの各点が、それぞれの特徴を対応づけたときのコスト値となる。

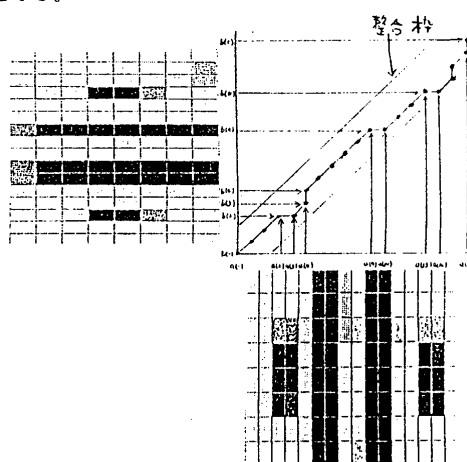


図3 文字パターンのDPマッチング

#### 5. 評価結果

本方式による文字認識の認識実験を写植、ワープロ等様々な出力によるゴシック体、明朝体に対して行なった。認識率は、字体の違いに関係なく、普通の品質の文字に対して、平均で約97%であった。誤認識の多くは、“王”、“玉”などの細かな違いが分からなかったものと、かな文字の間違いで、全体の9割を占めた。これは、本方式の擬似2次元マッチングが、縦/横の直線部分には有効であるが、かな文字の曲線部分などの大幅な変形には十分対応しきれない点、微かな部分が特徴量に十分反映できない点にあると考える。

#### 6. まとめ

本方式による、文字パターンより抽出した方向性特徴に対して、この特徴に対応したメッシュ分割方式と、擬似的2次元ずらしマッチングによる文字認識が、フォントの多様性に対応する認識方式として、有効であることを確認した。

#### 参考文献

- [1] パターン認識：白井良明編、オーム社 1987.8