

K_038

大型ディスプレイを用いた手書き文字・図形の認識システム Handwritten character and figure recognition system on large display

川田 貴俊†
Takatoshi Kawada

長尾 智晴†
Tomoharu Nagao

1. はじめに

人間にとって手書きで文書やスケッチを残すということは、直感的かつ容易な作業である。しかしコンピュータ内で扱われる電子的なデータと比べ、保存・編集・移動・複製が困難であるといった欠点も挙げられる。コンピュータにはこのようなメリットがあるが、標準的なインタフェースとなっている“キーボード+マウス”は、設置にスペースが必要といった空間的な問題と、使いこなすためにはある程度の慣れが必要という操作性の問題がある。

一方で近年の薄型テレビの技術開発によって、画面上で手書き入力可能で50型を超えるような大型ディスプレイが登場している。この装置には入力と出力を1台で行えるというメリットと、キーボードやマウスを通すといったことがないため直感的な操作が可能というメリットが挙げられる。

このような背景を受けて大型ディスプレイ上で手書き文字・図形データを認識・清書・電子化することへの需要が今後ますます高まると考えられる。具体的には、プレゼンテーションやブレインストーミングといった活発に意見交換が行われる場で発言やアイデアを書き留めたり、図を使って相手に説明したりといった使われ方が考えられる。

そこで本研究では手書き入力可能な大型ディスプレイを用いることを前提として、さらにその特性を活かした手書き文字・図形の認識システムを構築することを目的とした。

2. 手書き文字認識

本システムでは手書き文字の認識をDPマッチング^[1]で実現した。あらかじめ辞書データベースを用意しておき、それらと手書き文字を比較して、最も距離の小さい文字を認識結果として出力する。なお、比較は同じ画数の文字同士だけを対象とする。比較の前処理として、予め決めておいたサイズの正方形に外接するように、縦・横を同じスケールで伸縮し、正規化する。その後、2つの文字の先頭ストロークから順番に距離を求め、その総和を画数で割ったものを文字間の距離とした。

ストロークはプロットされた各点を始点とするベクトルの時系列データとして表現され、文字はストロークの時系列データとして表現される。図1に数字の“3”を書いたときのイメージを示す。

DPマッチングを行うときは、2種類の距離を定義し、それぞれに重みをつけて足し合わせたものを文字同士の距離とした。1つ目は2点間のユークリッド距離で、直感的な結果を得ることができる。2つ目は座標軸に対するベクトルの角度を使ったもので、角度差が小さければ距離はほぼ0

になり、差が大きくなると急激に距離が大きくなる距離を定義した。2種類の距離を導入することで、ストローク同士の位置関係とストロークの形を考慮した処理を実現することができる。

本システムでは大型ディスプレイを用いることを前提としているので、単一の文字認識だけではなく、単語の認識についても実装した。また日本語文章では“や”・“ゆ”・“よ”といった小さい文字も登場するのでそれらの認識も可能とした。

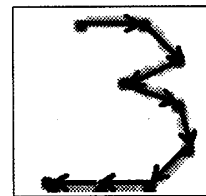


図1 文字のデータ構造

3. 手書き図形認識

本システムは大型ディスプレイを用いることを前提としたシステムなので、実用上認識することができると便利な図形ということで、円・楕円・自由曲線・矢印・表を認識対象とした。

手書き文字では書き順が個人によって変動することがある。手書き図形の場合はそれに加えて、ストローク方向の変動も多く発生する。それだけでなく、これらの変動は同じ人物が同じ図形を書いた場合でも頻繁に発生する。したがって、手書き図形認識をDPマッチングで実装するのは困難である。

そこで本システムでは、DPマッチングのように入力データをオンライン情報として扱うのではなく、手書き図形認識に適したオフライン情報に変換してから認識を行うことにした。具体的には図2に示すようなストロークの接続点（図中の円部分）に注目し、この接続点に情報をもたせてそれを使って認識処理を行う。

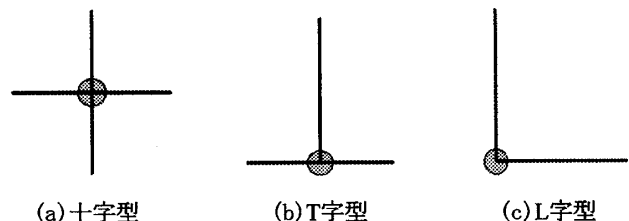


図2 ストロークの接続箇所

接続点のもつ情報としては、自分自身の座標・接続され

†横浜国立大学 大学院環境情報学府, Graduate School of Environment and Information Sciences, Yokohama National University

た2本のストロークのベクトル成分・およびその長さを選んだ。接続点のイメージを図3に示す。図に示すように、1つの接続点につながるストロークは常に2本だけである。図2(a), (b)のように1点に複数のストロークが集まっている場合はストロークの組み合わせの数だけ同じ場所に接続点が生じられる。

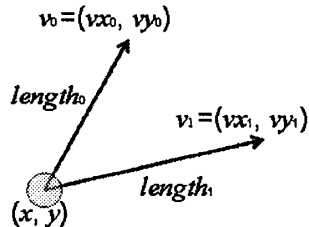


図3 接続点のデータ構造

ストロークが書かれてどのように接続点が生じられるかを、図2(a)の十字接続を例に解説する。十字が|, 一の順番に2画で書かれたとする。1画目が書かれただけでは接続点はないので、接続点は生成されない。次に2画目のストロークが書かれた時点でストロークの交差が発生するので、2本のストロークは分割され、計4本のストロークが書かれた状態になる。その結果、十字接続の場合は4本のストロークが1点で交わっていることになり、ストロークには $4C_2=6$ 通りの組み合わせがあるので、6個の接続点情報が生成される。同じようにT字型の場合は3本のストロークなので3個の接続点情報が生成される。また、L字のような折れ線が1画で書かれた場合は、ストロークが角の部分で分割され、2本の線分が接続した状態と同じになる。図4に矢印が書かれた場合の接続点の生成の様子を示す。矢印は図2(b)のT字型の変形と考えられるので、3つの接続点が生じられる。

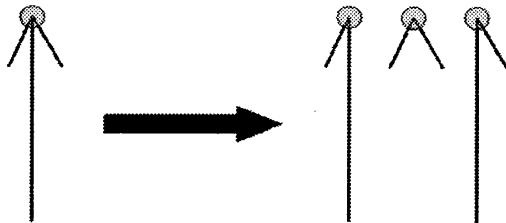


図4 矢印のデータ構造

次に、手書き図形が接続点情報に置き換えられた後、それがどの図形であると認識されるまでの処理手順について述べる。大まかな流れは手書き文字認識と同様で、予め登録しておいた辞書データベースの図形と手書きした図形を比較し、最も距離が小さい図形を認識結果として出力する。

先に述べたように、本システムは大型ディスプレイで入力を行うことを前提としているので、書かれる図形の大きさにはばらつきがある。このことから、文字の場合と同様に比較前に正規化を行う。正規化された図形同士の距離を求める手順の概略は次のようになる。

- ① 接続点をもつベクトルの成分と長さの情報を使って、接続点同士の距離 $dis1$ を求める
- ② 接続点間のユークリッド距離に応じて、 $dis1$ に重みをつけて $dis2$ とする

- ③ $dis2$ が最も小さくなるような接続点の組み合わせを探し、ベクトル情報を使ってさらに重みをつけ、その総和を2つの図形間の距離とする

図3のようなデータ構造は図形が複数のストロークから構成され、かつ全てのストロークが直線の場合に有効な構造である。しかし、本システムでは円・楕円・自由曲線も認識対象としているので、これらの図形に関しては別のデータ構造を考える必要がある。円・楕円・自由曲線は1筆書きすることができる図形で、実用上も複数画に分けて書かれることはないと考えられる。そこで、ストロークが書かれたら最初にこれらの図形かどうかを判断し、そうでなければ矢印・表の一部として扱うことにした。

4. 結果

図5に手書き文字・図形の認識例を示す。図5は“きょう”と“きょう”を書いて単語認識させた後、それぞれの単語を表で囲んだ状態である。

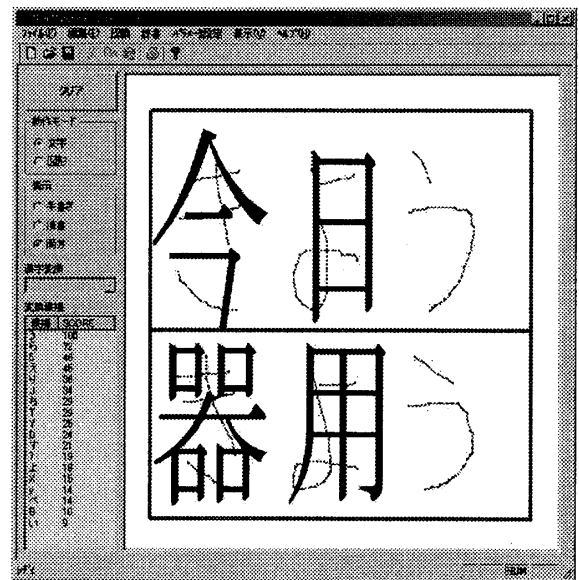


図5 文字と図形の認識結果

5. まとめ

大型ディスプレイという特性を活かした手書き文字・図形の認識システムを構築することができた。文字認識ではストロークの位置ずれと形を考慮した認識が行われ、図形認識ではストロークの順番、描画方向の変化に強い認識が可能であった。

参考文献

- [1] 酒井幸一, “デジタル画像処理入門,” コロナ社, pp. 81-90, 1997
- [2] 三苦寛人, 内田誠一, 迫江博昭, “固有変形を用いたオンライン文字認識手法,” 信学技法, PRMU2004-239, 2005
- [3] 伊藤等, 中川正樹, “Hidden Markov Model に基づくオンライン手書き文字認識,” 信学技法, PRMU97-85, 1997