重なり文字を含む手書き文字画像からの文字切り出し

西　大地 [a)]・横道　政裕 [b)]

Handwritten Character Segmentation from Images with Overlapping Characters

Daichi NISHI and Masahiro YOKOMICHI

Abstract

In the handwritten character recognition from image, it is necessary to extract each character from the string. However, if a character comes in contact with another one, it is difficult to determine at where they should be separated. In this report, a segmentation method is proposed. The method is based on thinning the background region. Two methods to choose the best suited segmentation path is analyzed. The former uses the distance between the segmentation paths and the center of the image. The latter is based on a image feature used in character recognition. Their performances are examined by real image experiments.

Keywords: Character Segmentation, Offline Character Recognition, Background Thinning, P-LM

1. はじめに

近年では、携帯電話やゲーム機にもタッチパネルが組み込まれており、それに文字を書きこませて処理をするオンライン手書き文字認識が普及している。スマートフォンには名刺読み取りのアプリケーションなどもあり、さらには文字の認識だけではなく楽譜認識など光学文字認識技術は発達している。このように文字認識技術は日常生活に身近なものとなっている。

しかし、そのような文字認識技術にもまだ課題はある。上で述べたものは、認識する文字の種類に傾向があったり、専用の媒体に書き込ませたりするものであり、たとえば、紙などに書かれた手書きの文章を読み取らせるオフライン手書き文字認識は様々な問題があり、まだあまり実用化されていない。

オンライン手書き文字認識では機械に直接文字を書き込んでいくため、文字の形だけではなくストローク情報を文字認識処理に使うことができる。さらに、入力するタッチパネルにも大きさに制限があり、その範囲内に文字を書き込んでいくため文字と文字の区切りを特に意識する必要がなく、高い認識率が得られる。それに比べ、オフライン手書き文字認識ではすでに書かれている文字を対象に認識するためストローク情報を利用することは難しく、認識率は低くなる。

他に、文字を認識する際は 1 文字ごとに認識を行うため、文字列領域をテキストデータに起こす場合にはまず、文字と文字の区切りがどこにあるのかも判断していかなくてはならないという問題がある。日本語の文字は、アルファベットと違い 1 文字を構成する線が離れているものがあり文字の切出しを難しくしている。たとえば、横書きの文字列に「群」という文字があった場合、切出し方によっては「群」とも「君」と「羊」という 2 つの文字の並びとも判断されてしまう。文字と文字が接触してしまうことで正確に切り出せないこともある。現在、郵便物の住所の読み取りなどにオフライン文字認識技術が用いられているが、これは書かれる文字列にある程度パターンがあることと、「都道府県」や「市町村」などのほぼ必ず用いられる文字に着目することで、文字区切りの認識精度を高めていることで成り立っている。

これらのような問題があるため、オフライン手書き文字

a) 情報システム工学専攻修士 2 年生

b) 環境ロボティクス学科担当　准教授

認識技術はオンライン手書き文字認識に比べ精度が低く、普及していないのが現状である。

本稿では、これらの問題点の中から、静止画像の文字認識を行う際に必要な、文字列からの文字切出しの問題に着目し、背景領域の細線化を用いて接触文字の切出しパスを判定し、文字の縦横比と文字認識の際の尤度をもとに切出す処理を行うか判定する手法を提案する。

2. 文字認識と文字の切り出し

オフライン文字認識の処理プロセスの流れを説明する。まず、スキャナなどで取り込んだ手書き文字の画像を文字認識に適したデータに変換する前処理を行う。画像処理の内容としては、画像全体から文字列の部分を取り出す文字列領域抽出を行う。次にノイズの除去を行い、白黒の 2 値化処理を行う。

この画像処理によって得られた文字列領域のデータを用いて文字認識処理を行う。文字認識処理の流れは、以下の通りである。最初に文字列領域からを 1 文字毎に文字領域を切出す。そして、文字データごとに文字認識処理を行い、得られた結果をテキストデータとして出力する。

2.1 文字切出し

文字認識は、1 文字を対象に認識をする手法が一般的である。そのため、文字認識で文字列を処理するためには、文字列から 1 文字ずつ切出す必要がある。新聞や小説などは文字と文字の間にスペースが入るため文字列の方向に対して垂直に見たときの画素の数を調べ、黒画素のない列を文字と文字の間のスペースとし、区切ることができる。図 1 は文字列と、その文字列の方向に対し垂直に見たときの黒画素の数のヒストグラムである。しかし、手書き文字などの場合は人によっての癖があり、文字ごとのスペースは人ごとに異なる。アルファベットと違い、ひらがな、カタカナ、漢字には 1 つの文字でも間にスペースがあるものがあり、この問題を難しくしている一因となっている。さらに、図 2 のように文字と文字の間に黒い画素のない列がない場合や、文字が接触した場合、スペースで切出すことはできなくなり、文字の切出しは難しいものとなる。

2.2 文字切出しの手法

まず、文字列画像から画像の縦の長さをもとにして、白

図 1 文字列と各列の黒画素のヒストグラム

図 2 文字の境界が垂直方向の線分で分けられない例

ピクセル列で画像を切出す。その後、切出した画像の中から縦の長さに対しての横の長さの比率が一定の値を上回る矩形を接触文字として判断し、背景領域の細線化を用いて切出しを行い 1 文字ずつの画像にする。しかし、背景領域の細線化を用いた際、区切るラインが文字の中を通ることがある。この章では、背景領域の細線化を用いた文字の切出しの処理手順について述べる。そしてその際に起こり得る問題とそれらの解決策について述べる。

2.2.1 画素数を用いた切出し

日本語には、ひらがな、カタカナ、漢字がある。これらには、文字の中に白ピクセル列が存在するものが存在する。そのため、白ピクセル列を文字の区切りと考え、左から右に白ピクセル列を探していくと、文字の中にある白ピクセル列で区切ってしまう問題が発生してしまう。そこで、日本語の文字は一般的にある程度正方形に似た四角形の範囲に収まることに注目し、文字列画像の縦幅の値を用い、文字切出しを行う手法を考える。

対象の文字列画像からすべての黒画素を含むような矩形に切出す。その後、その矩形の縦の長さをもとに区切るラインを設定する。そのライン上に黒ピクセルがない場合、そのラインで切出す。もし、区切るライン上に黒ピクセルが存在した場合は、そのラインをもとに左右に 1 ピクセル列ずつずらしながら白ピクセル列を探し、その左右の白ピクセル列を切出し候補とする。その後、左右の切出し候補の列で切出し、それらの矩形に、再度黒画素を含むような

矩形に切出す処理をする。その後、その二つの矩形の縦横の長さで、長い方に対する短い方の比の値が 1 に近い方の切出し候補のピクセル列で切出す。

2.2.2 背景領域の細線化

2.2.1 項で述べた手法では、文字と文字の間に縦の白ピクセル列がないと文字を切出すことはできない。そこで背景領域の細線化を行うことで対応する [1), 2), 3)]。

細線化とは、二値画像を幅 1 ピクセルの線画像に変換する処理のことであり、文字認識やパターン処理を行う際の前処理などで行われる。 白いピクセル列を探して文字の切出し位置を探す場合、図 3 の様な文字列に対しては切出すラインが探せないが、文字線と文字線の間に白画素は存在する。そこで、背景領域に対して細線化処理を施すことで文字と文字の境界を探す。背景領域の細線化の過程を図 4 に示す。

図 3 背景の細線化の例

細線化手法に hilditch の細線化手法を用いる。この項では、細線化の手法について述べる。この処理は、2 値画像に対して行い、白画素の領域を細線化するものである。黒画素の濃度値を 1、白画素の濃度値を 0 とする。

2 値化された画像に対し、対象画像の画素についてラスタスキャンを行う。対象の画素の座標を P_0 とし、その座標の 8 近傍の画素をそれぞれ、図 5 のように右隣りから反時計回りに P_1 から P_8 と割り振る。

$B(P_k)$：P_k の濃度値(1 か 0)

$$N_4 = \{1,2,3,4\}$$

$$N_{odd} = \{1,3,5,7\}$$

$$N_8 = \{1,2,3,4,5,6,7,8\}$$

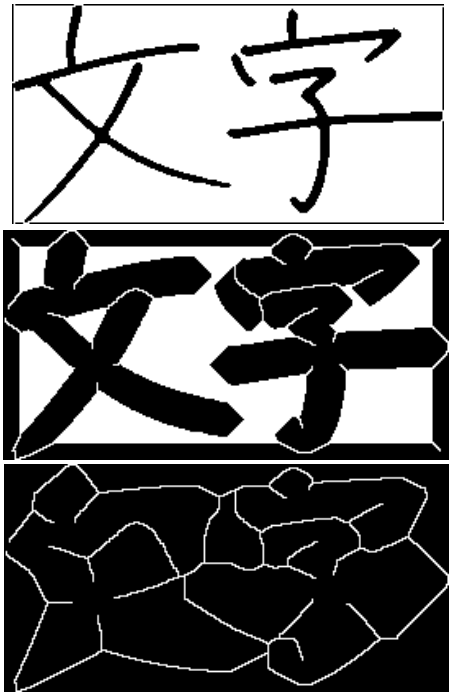

図 4 背景の細線化の過程。上：1 回目、中：10 回目、下：62 回目

$$C_k = \begin{cases} 1: B(Pk) = 1 \text{ のとき} \\ 0: \text{その他のとき} \end{cases}$$

$$F(P_k) = \begin{cases} 1: |B(Pk)| = 1 \text{ のとき} \\ 0: \text{その他のとき} \end{cases}$$

とし、以下の手順を行う。

P_4	P_3	P_2
P_5	P_0	P_1
P_6	P_7	P_8

図 5 8 近傍の画素

［手順 1］ 対象画像をラスタスキャンし、以下の 6 つの条件をすべて満たした時、$B(P_0)$に-1 を代入する。

(条件 1)

$B(P_0)$=1

P_0 が処理の対象である条件

(条件 2)

P_0 が境界画素である条件: $\sum_{k\in N_4}\{1 - |B(P_{2k-1})|\} \geq 2$

(条件 3)

端点を削除しないための条件：$\sum_{k\in N_8}|B(P_k)| \geq 2$

（条件 4）

孤立している点（周りに白画素がない点）を保存する条件：$\sum_{k\in N_8} C_k \geq 1$

（条件 5）

連結性を保存する条件（※この式の P_9 の値は P_1 の値を用いる）：

$$\sum_{k\in N_{odd}} \{F(P_k) - F(P_k)F(P_{k+1})F(P_{k+2})\} = 1$$

（条件 6）

幅が 2 の線分に対して、その片側のみを削除する条件[1]：$\forall i \in N_8$において、$B(P_k) \neq -1$ または$B(P_k) = 0$と仮定したときに

$$\sum_{k\in N_{odd}} \{F(P_k) - F(P_k)F(P_{k+1})F(P_{k+2})\} = 1$$

が成り立つ。

［手順 2］

ラスタスキャン終了後、$B(P_0)$=-1 となる点がある場合、$B(P_0)$=-1 の点の $B(P_0)$に 0 を代入する。その後、［手順 1］に戻る。$B(P_0)$=-1 となる点がない場合、処理を終了する。

以上の処理を行うことで、画像に対する細線化処理が完成する。

3. 接触文字への対応

背景領域に対して細線化をすることで、文字と文字の間にラインができ、そこで切出すことができるが、文字と文字が接触している場合には対応できない。そこで、端点同士の結合を行う。結合は 2 つの端点が以下の条件に当てはまる場合に行う（図 6）。

- 2 つの端点が、文字列画像の横方向の中心から±15%以内に存在する
- 2 つの端点の距離が、画像の縦の長さの 6 分の 1 以下になっている
- 2 つの端点を結んだ線分と、文字列画像に鉛直に引いた場合の線とのなす小さい方の角度が 45 度以下となる

[1] この式の P_9 の値は P_1 の値を用いる

図 6 端点の結合前と結合後

3.1 細線化された背景を用いた切出しラインの選定

例えば、図 3 において細線化した際、細線化された背景の中には「文」と「字」の間を通る経路が存在する。しかし、「字」の「うかんむり」の左側を分断する経路も存在し、どちらの経路で切出すかによって文字の形は変化するようになり、認識精度に影響を及ぼすことになる（図 7）。

そこで、切出しラインの候補を複数用意し、その中から最善のラインを探す手法を提案する。以降は候補数を 5 とした場合の例を示す。 切出しラインの候補は以下の手順で探す。

- 背景領域を細線化してできた線上の交差点の中から、文字列画像の中心に水平方向でみた際の距離が近い点を 5 つ選ぶ。
- 始点から上方向と下方向に対して、線上をたどる。その方向に線がない場合は、画像の中心に向かうように左右に移動する。
- 画像を水平方向に 4 分割した際の、一番上と一番下の領域の交差点にたどり着くまで行う。

こうして得られた 5 つの切出しラインの候補に対して、画像の中心からの横方向の距離を用いる手法と文字認識の際に利用される特徴の 1 つである P-LM 特徴を用いた手法の 2 つを考える。

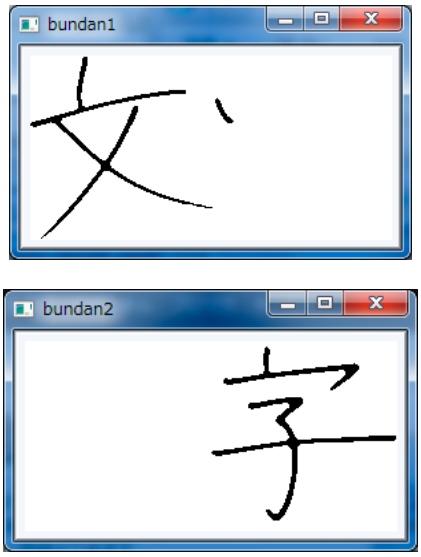

図 7 切出しの失敗例

3.2 中心からの水平方向の距離を用いる手法

2 文字で構成されている文字列画像の場合、文字と文字の境目は画像の中心に近いと考えられる。そこで、切出しラインの候補の内、画像の中心に近いラインを切出すラインとする。

方法としては、まず中心を通る鉛直方向の線をとり、縦に 1 ピクセルごとずらしながら切出しラインとの距離を調べ、その距離の 2 乗の平均をとる。これを 5 つの切出しラインの候補すべてに行い、値が一番小さいものを切出しラインとする。なお、図 8 の上から 2 列目のような場合は、それらのピクセルの中心との距離を測る。

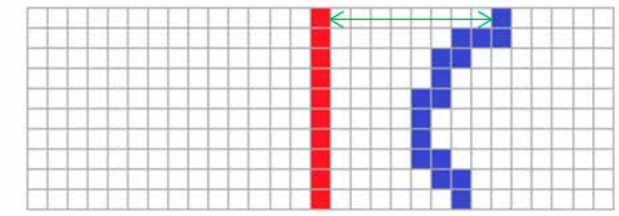

図 8 切出しラインと中心との距離の求め方

3.3 P-LM 特徴を用いる手法

文字の切出しが正確に行われていなかった場合、文字認識の際に悪影響を及ぼす。このことに注目し、各切出しラインで切出しを行った時の画像に対し文字認識を行う際に用いられる特徴量を求め、切出しラインの候補から切出しラインを選ぶ。

本研究では、オープンソースの日本語 OCR ソフトウェア、NHOCR[4]などで用いられる外郭局所的モーメント(peripheral local moment)特徴[5]（以下、P-LM 特徴）を文字特徴量として用いる。P-LM 特徴は輪郭線に着目した特徴抽出法で、局所的に観測した文字線の方向情報を分散、共分散の値で表現する。P-LM 特徴は文字のつぶれ、太さ変動、位置変動の変形文字の影響を受けにくいという特徴がある。

P-LM 特徴は、観測点において文字線を中心にしてある大きさのウィンドウを開き、ウィンドウ内の画素をX,Y 方向に射影する。そして、ウィンドウ内の画素の総和をn、第i番目の画素が存在する各座標を$(x_i,\ y_i)$として、式(1)のように 2 次モーメント（分散、共分散）を定義する。ただし、 $\bar{x},\bar{y}$は、射影の重心座標である。

$$S_x=\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^2,S_y=\frac{1}{n}\sum_{i=1}^{n}(y_i-\bar{y})^2,$$

$$S_{xy}=\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})\,(y_i-\bar{y}) \qquad (1)$$

そこで、5 つの切出しラインの候補で切り出した際の左右の画像をそれぞれ認識させ、その時用いた特徴量と辞書データでの文字との特徴量の分散をとる。その左右の値を合計させ、一番値の小さくなる切出しラインで切出すこととする。

4. 実験

実験データとして、200dpi の解像度、jpg 形式で取り込んだ、ひらがな、カタカナ、漢字で書かれた横書き文字列の画像データを用いた。あらかじめノイズ除去、2 値化を行い、接触、もしくは文字間に縦方向の白画素列のない（入り込んでいる）2 文字の画像を用いて実験を行った。

背景領域の細線化によって 2 つに分割した画像にお互いの文字の線が入っていなければ○、片方の画像にのみ入っていれば△、左右の画像にお互いの文字の線が入っていれば×とした。結果を以下に表す。

表 1 実験結果

	手法 1			手法 2		
	○	△	×	○	△	×
接触文字	1	4	4	3	2	4
入り込み文字	2	3	0	3	2	0

実験データが少ないものの、接触文字入り込み文字、共に P-LM 特徴量を用いた手法の方が、切出しの成功率は高い。

どちらの手法でも結果の悪かった接触文字は図 9 や図 10 のようなもので、接触している線が比較的滑らかにつながっていたことが原因と考えられる。背景領域を細線化したものを見てみると、線が文字との間を通るものはなく、端点もなかったため、切出しラインが作れなかったものと思われる。図に至っては、″あ″の右下の線と″な″の 2 画目の線がつながる形で切出されてしまっている。これは″あ″の右下のくぼみに線が入り込んだことと、″な″の 2 画目の線の傾きが似ていたことの 2 つが原因と思われる。このような文字との接触部位にあまり角度が現れないときの切出し方が今後の課題と思われる。

5. まとめ

背景領域の細線化から横書きの文字列から 1 文字ずつ切出す実験を行った。この実験から、文字認識の特徴から、切出しラインの選定に役立つことが分かった。特徴量を用いた文字認識技術だけではなく、他の文字認識技術を用いることでより正確なラインの決定が行えることが考えられる。

他にも、文字の接触部位の形によっては文字の間に切出しラインが得られないことが分かり、今後このような接触に対しての解決が課題となると考えられる。

参考文献

1) 中山 英久、藤原 勇太、加藤 寧、”背景領域細線化を用いた手書き文字切出しの改良手法”情報科学技術フォーラム講演論文集 8(3), 365-370, 2009.

2) 梅田 三千雄、橋本 智広、背景領域の細線化に基づく古文書の文字切り出しと認識、情報処理学会論文誌 45(4), 1188-1197, 2004.

3) Z. Liang, and P. Shi, A metasynthetic approach forsegmenting handwritten Chinese character strings, Pattern Recognition, vol.26, no.26, pp.1498–1511, 2005.

4) 後藤英昭、NHOCR, http://code.google.com/p/nhocr/

5) 堀桂太郎、根本孝一、伊藤彰義、“文字の輪郭線に着目した手書き漢字の特徴抽出法:外郭局所的輪郭線特徴と外郭局所的モーメント特徴”、電子情報通信学会論文誌、J82-D-II(2), 188-195, 1999.

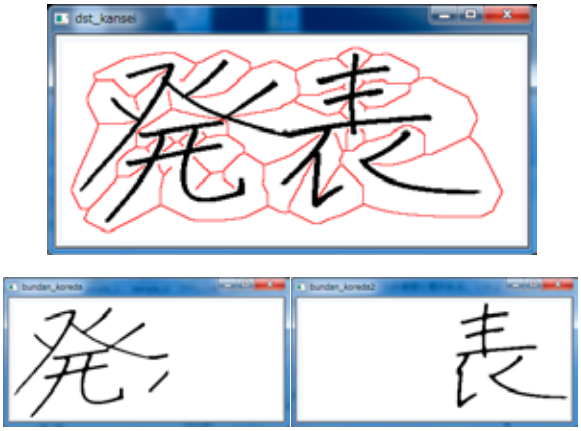

図 9 切り出しの失敗例 (1)

図 10 切り出しの失敗例 (2)