



HOKKAIDO UNIVERSITY

Title	漢字パターンの細線化と特徴抽出の一方法
Author(s)	仲谷, 元; Nakantani, Hajime; 青木, 由直 他
Citation	北海道大學工學部研究報告, 80, 45-56
Issue Date	1976-06-30
Doc URL	https://hdl.handle.net/2115/41368
Type	departmental bulletin paper
File Information	80_45-56.pdf



漢字パターンの細線化と特徴抽出の一方法

仲谷 元* 青木由直*

(昭和 50 年 12 月 27 日受理)

A Method for Extracting Line Figures and Features from Chinese-Characters

Hajime NAKATANI Yoshinao AOKI

(Received December 27, 1975)

Abstract

In this paper, we propose a method for extracting line figures and features from Chinese-characters. This method is less influenced by noises on the edges of a line than a conventional one. Line figures are obtained by extracting maximum points after blurring of original patterns. On the other hand strokes are classified according to their lengths and directions. Feature points, such as edge points, branch points, and crossing points are extracted by counting the number of connection for each "1" element of line figures. Lastly, we classify Chinese-characters according to three steps by those features.

1. ま え が き

文字の自動認識を行なう場合、対象は記録された文字であり、それらが印刷文字であるか、手書文字であるかによらず、一般には種々の雑音を伴ったものである。そのため、前処理によって対象に含まれる雑音を除去、軽減することが以後の認識処理において特徴の抽出を安定にし、識別率を向上させる意味で重要である。前処理として具体的にどのような処理を行なうかは、認識の対象となる文字の種類や認識の方法によって異なるが、ストロークアナリシ的な幾可学的特徴抽出法では、文字線とそれらの相互の連結状態を確実に抽出することが認識に必須の条件となることから、文字線の細線化処理を行なって心線図形を得ておくことは前処理として重要な意味をもつものである。

細線化処理については、対象を文字とする場合に限らず多くの研究例が報告されている^{1)~3)}。一般には、文字などの被細線化パターンにおいて輪廓部分を構成する要素を反復除去し、最後に残った要素で構成されるパターンを心線図形とすることで細線化を行なっているが、輪廓要素が除去可能である要素とは限らず、除去することで文字線が消失・切断される場合も考えられる。また、その原理から、得られる心線図形は線縁に分布する雑音に強く影響されたパターンとなる。特に記録された文字パターンは線縁の雑音が多と考えられることから、これらの方法は必ずしも文字線の細線化に適した方法であるとは言えない。そこで本報告では、線縁雑音に対して

* 電子工学科

強く比較的 に文字線の細線化に適した方法として、ぼかし処理による細線化の方法について印刷漢字を対象として論じ、さらに漢字パターンからの特徴の抽出方法とその表現方法を幾可学的特徴抽出法の観点から論じている。

2. 漢字パターンの細線化

2.1 ぼかし処理による細線化の方法

(1) 原理 ここで提案する方法は、“0”と“1”に標本化された二値図形に対して、文字線の心線図形を構成する要素が高い値をもつように重み付けを行ない、その後極大点を抽出して心線図形とする方法である。すなわち、周囲に値が“1”の要素(以後“1”要素と呼ぶ)を多くもつ点ほど文字線の心線図形を形成している度合いが強いと考へ、重み付けを行なう要素の周囲の“1”要素に対して適当な重みを乗じて和をとる操作により重み付けを行なう。図1に1次元の場合を例にとって、細線化の原理を示す。被細線化パターン(a)から細線化パターン(b)を得

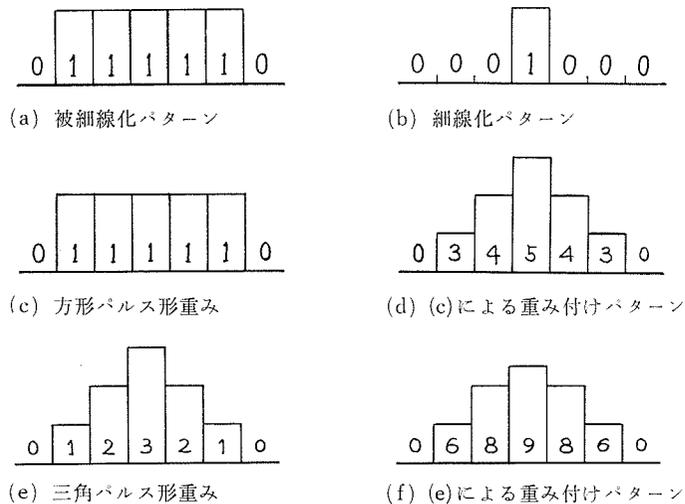


図1 細線化の原理

るためには、(d)または(f)に示すような重み付けされたパターンの極大点を抽出すればよいことがわかる。(d)に示す重み付けパターンは(c)に示す方形パルス形重みで、つまり被細線化パターンの重み付けを行なう要素を中心とする適当な範囲に含まれる“1”要素に等しい重みを乗じて和をとることで得られる。しかし、方形パルス形重みによる重み付けの方法では、和をとる範囲を被細線化パターンの幅に等しくとらねば(d)のような極大値をもつ重み付けパターンは得られない。このことは、文字が線縁雑音などを含み一定の線幅をもったものではないことを考えると、文字線の細線化には不適当な重みであると言えよう。これに対して、(e)に示す三角パルス形重みのような、重み付けを行なう要素に近いほど高く、遠いほど低い重みを選ぶと、和をとる範囲を被細線化パターンの幅に等しくとらなくとも、より広くとっておくことで常に(f)に示すような極大値をもつ重み付けパターンが得られる。したがって、文字線の細線化を目的とする重み付けには三角パルス形やガウスパルス形のような重みを選ぶことが大切である。

さて、このような重み付けは、数学的には被細線化パターンと重み関数との相関をとるものと言える。すなわち、被細線化パターンを f 、重み関数を w とすれば、重み付けパターン g を式(1)で求めることである。

$$g(x, y) = \iint_S f(x', y') w(x-x', y-y') dx' dy' \quad (1)$$

ただし、積分範囲 S は被細線化パターンが存在する領域である。また、重み係数は偶関数にとることを考えれば式(2)となる。

$$g(x, y) = \iint_S f(x', y') w(x'-x, y'-y) dx' dy' \quad (2)$$

式(2)は f と w のコンボリューション演算となっている。したがって、ここに示した重み付けの方法は被細線化パターンをぼけの関数を w としてぼかすことによるものであると考えられ、この方法をぼかし処理による細線化の方法と呼ぶことにする。

(2) 重み関数の選択 式(1)または式(2)を計算することで重み付けがなされるが、その際に重み関数 w を適当に定めることが必要である。以下に1次元の場合の重み関数を示す。

a. 方形パルス形

$$w(x) = \begin{cases} 1/(2\alpha) : |x| \leq \alpha \\ 0 : |x| > \alpha \end{cases} \quad (3)$$

b. 三角パルス形

$$w(x) = \begin{cases} (1/\alpha)(1-|x|/\alpha) : |x| \leq \alpha \\ 0 : |x| > \alpha \end{cases} \quad (4)$$

c. 余弦パルス形

$$w(x) = \begin{cases} \{1 + \cos(\pi x/\alpha)\}/(2\alpha) : |x| \leq \alpha \\ 0 : |x| > \alpha \end{cases} \quad (5)$$

d. ガウスパルス形

$$w(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{x^2}{2\sigma^2}\right) \quad (6)$$

a. b. c. の重み関数については正定数 α により和をとる範囲を定め、d. については正定数 σ によって定める。 α または σ の定め方は細線化の目的に合った重み付けをするうえで非常に重要である。いずれの場合も α または σ を大きく定めると和をとる範囲が広くなる、すなわちより広い範囲の情報により重み付けを行なうことになる。この場合、極度に広い範囲から重み付けを行なうことは、隣接するほかの文字線分による影響が強くなるので好ましくなく、逆に狭い範囲からであると心線図形の情報を見失うことになる。前項で方形パルス形の重み関数は文字線の細線化を目的とする場合には不適當であると述べたが、ほかの場合は比較的に適した重み関数である。したがって、b. c. の場合はこの点に注意して、 α を最大文字線幅 T_{\max} に対して式(7)の範囲に定める。

$$T_{\max}/2 \leq \alpha \leq T_{\max} \quad (7)$$

式(7)は少なくとも文字線幅の範囲から、文字線幅の2倍の範囲までの情報を得て重み付けを行なうことを示している。また、ガウスパルス形の重み関数は式(6)から $\pm 4\sigma$ までの間に成分の99.99%が含まれることがわかるから、線縁雑音を考慮しても最大文字線幅の2倍の範囲と一致するように式(8)で σ を定めておくことが適當である。

$$\sigma = T_{\max}/4 \quad (8)$$

ところで、実際に細線化を行なう文字パターンは2次元パターンとなるから、重み関数も2次元に拡張して考えることが必要である。ここでは、ほかの重み関数と比較して σ に関する条件

がゆるやかであると考えられるガウスパルス形の重み関数を例にとって考察する。この時、重み関数は式(9)で示される。

$$w(x, y) = \frac{1}{2\pi\sigma_x\sigma_y} \exp\left\{-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right)\right\} \quad (9)$$

正定数 σ_x, σ_y は図2に示されるようにそれぞれ横方向、縦方向のどれくらいの範囲から情報を得るかに対応し、1次元の場合における考察から、最大縦線幅を T_x 、最大横線幅を T_y とすると式(10)で定めることが適当である。

$$\sigma_x = T_x/4, \quad \sigma_y = T_y/4 \quad (10)$$

このことは、文字線が線雑音などにより変動する場合でも線幅の2倍程度の変動ならばほとんど影響されないことを示し、印刷漢字を細線化の対象に選んだ場合には活字の種類によっておおむね文字線幅が定まることを考えると、1種類の重み関数ですべての漢字パターンに対して重み付けが可能となるわけで、2次元ガウス分布形の重み関数は極めて都合の良い重み関数であると言える。

(3) **重み付けの方法** 原理で述べたように、ここでの重み付けは式(1)または式(2)に示される被細線化パターンと重み関数との相関ないしコンボリューション演算を行なうことでなされる。以下にこのための方法を示す。

a. 空間回路網による並列処理の方法 この方法は図3に示すような加算係数器を用いたハードウェアによる並列処理での重み付けの方法である。一般に2次元パターンを扱う場合には、格子状の点で標本化することから、式(1)の積分形式は式(11)のように総和形式となる。

$$g_i = \sum_{j=1}^N w_{j-i} x_j \quad (11)$$

ただし、 N は標本点の数である。したがって、式(11)は式(12)で与えられる図3(b)に示す演算増幅器を用いた加算係数器の動作に一致していることがわかる。

$$y = -\sum_{j=1}^N \frac{R}{r_j} x_j \quad (12)$$

結局、 R と r_j の比を適当に選ぶことで重み係数が定まり、図3(a)のように重み付けする各点に対して、図3(b)の加算係数器を用意することで瞬時に重み付けが可能となる。

b. 光学的空間フィルタリングによる

方法 この方法は、被細線化パターンと重み関数とのコンボリューション演算が周波数領域における乗算で求まることを利用したものである。すなわち、式(2)は式(13)と等しい。

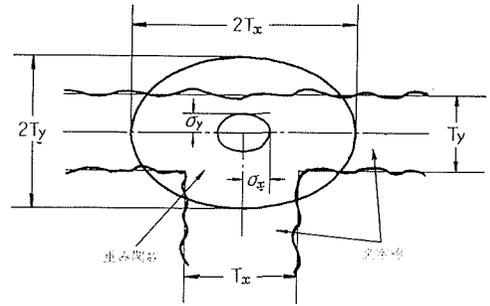
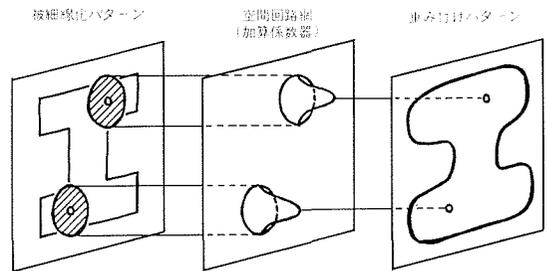
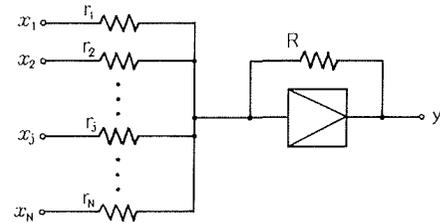


図2 2次元ガウス分布形重みと文字線幅の関係



(a) 空間回路網による重み付け



(b) 加算係数器

図3 空間回路網による並列処理

$$g(x, y) = F^{-1} \left[F \{ f(x, y) \} \times R(\xi, \eta) \right] \tag{13}$$

ただし、 F, F^{-1} はそれぞれフーリエ変換、逆変換オペレータをあらわし、 R は重み関数のフーリエ変換であり、式(9)で与えられる2次元ガウス分布形の重み関数を選べば式(14)のように解析的に求められる。

$$R(\xi, \eta) = \exp \left\{ -2\pi^2 (\sigma_x^2 \xi^2 + \sigma_y^2 \eta^2) \right\} \tag{14}$$

したがって、レンズのもつフーリエ変換作用を利用して式(13)の演算が可能である。結局、図4に示す光学系において、入力面に被細線化パターンのトランスベアレンシーを、周波数面に式(14)で与えられる重み関数のフーリエ変換のトランスベアレンシーを配置することで、出力面に重み付けされたパターンを得ることができる。この方法は、まさに出力として入力のぼけたパターンを得るぼかし処理であると言えよう。

c. ソフトウェアによる方法

先に示した二つの方法は、2次元パターンを空間的に同時に処理できるということで、文字認識における時間的な制約を満たすうえでも非常に都合の良い方法である。しかし、このような並列処理装置は特殊なものであり、必ずしも一般に使用できるものではない。そこで通常のコピュータを使ったソフトウェアによる重み付けの方法について述べる。それには、式(1)または式(2)に示される演算を行なうわけであるが、この時に

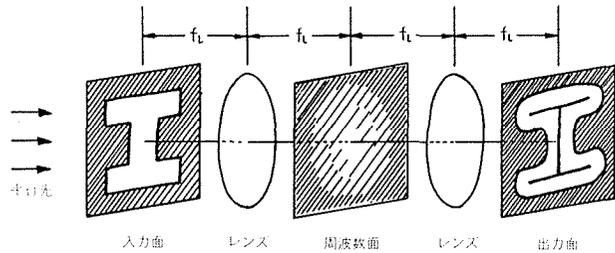


図4 光学的空間フィルタリングによる重み付け

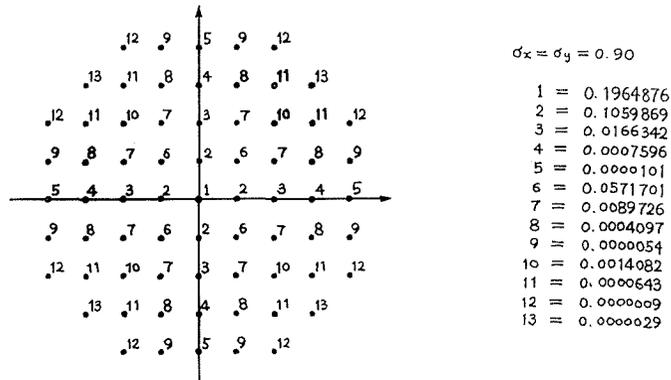


図5 2次元ガウス分布形重みの対称性

要する演算時間をいかに短くするかが問題であり、このためには乗算回数を少なくすることが必要である。被細線化パターンは格子状の標本点で二値に標本化されているが、一般には英数字記号の場合では 16×16 程度の標本点が必要であり、漢字の場合では 48×48 程度は必要となる。式(1)をまともに計算すると、 $M \times M$ に標本化した場合には M^4 回の乗算回数が必要となる。しかし、前に述べたように重み関数はせいぜい線幅の2倍の範囲でのみ値をもちそれ以外では零である。しかも、点対称であることを考えると、乗算回数を大幅に減らすことができる。図5として2次元ガウス分布形の重み関数の対称性を $\sigma_x = \sigma_y = 0.9$ の場合について示す。例えば線幅が $M/10$ とすると、一点の重み付けには約 $(2 \times M/10)^2 \times (1/8) = M^2/200$ 回の乗算回数が必要であり、すべての点に対しては約 $M^4/200$ 回の乗算回数で済むことになる。また、重み付けを“1”要素についてのみ行なえばさらに演算回数は減り、二けたから三けたの演算時間の短縮になるとと思われる。北大大型計算機センターのFACOM 230-75を使用した計算では $M=48$ の場合、FORTRAN言語を用いたCPU-TIMEで約0.6秒であった。

(4) **細線化の例** 先に述べたような重み付けの方法にしたがって重み付けされたパターンをもとに、その極大点すなわち尾根線を形成する要素を抽出することで細線化がなされ、心線図形を得ることができる。図6と図7にその例を示す。被細線化パターンは各図(a)に示すような15 mm 角のゴシック体を45×45の標本点に標本化したものを使用した。図6は四角すい形重みを使った例であり、図7は2次元ガウス分布形の重み関数を使った例である。

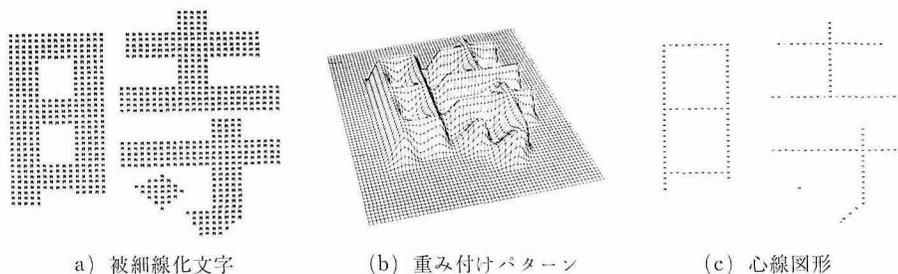


図6 四角すい形重みを使った細線化の例

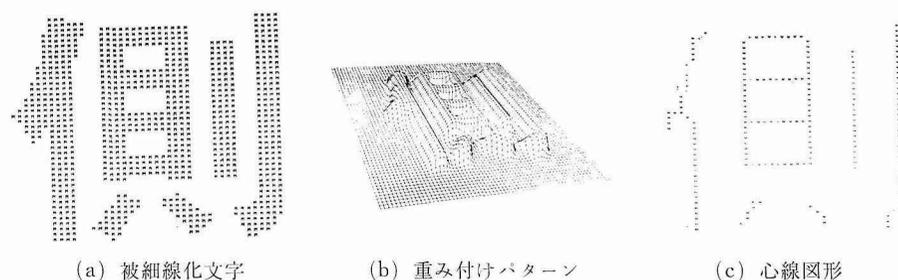


図7 2次元ガウス分布形重みを使った細線化の例

2.2 線縁雑音による影響

(1) **線縁雑音の発生法** 文字認識の対称となる記録された文字は実に濃淡変化の激しいパターンとして観測され、こうしたパターンを適当なしきい値を選んで二値化した場合は文字線の縁が細かい凹凸となる。このような文字線縁の細かい凹凸を線縁雑音と呼ぶ。したがって、文字認識装置を設計する場合は線縁雑音による影響を十分考慮しなければならない。以下にこのためのシミュレーションに使用できる、線縁雑音をパターンによらずに定量的に発生する方法について述べる。

この原理を簡単に述べると、雑音は文字線の輪廓部分すなわち白黒の境界部分に多く存在し、しかも周囲に黒い部分が多い所ほど多く存在すると考える。雑音を付加して“1”要素とするか、そのまま“0”要素とするかは、その周囲の8個の要素(以後近傍と呼ぶ)に含まれる“1”要素の数を計算し、その数の大小に応じた確率で判定すればよい。つまり、近傍に含まれる“1”要素の数が7のものは2のものよりも雑音となる頻度を高くとることにする。このときの確率を全体的に調節することで雑音付加の程度を変えることができる。確率現象は、一様乱数を発生させて使用する。一様乱数は、区間(0.1)で一様分布する乱数 x であり、その確率密度関数 P は式(15)である。

$$P(x) = \begin{cases} 1: 0 \leq x \leq 1 \\ 0: x > 1 \end{cases} \quad (15)$$

このとき、乱数 x と規準値 CR とを比較し式(16)が満足されるときに、“0”要素を“1”要素に

変えることで雑音を確率的に発生させることができる。

$$x \leq CR \quad (16)$$

図8に雑音レベルを4段階 ($L_1 \sim L_4$) に選んだ時の規準値と近傍に含まれる“1”要素の数 N との関係グラフとして示す。ここでは、規準値をレベルに応じて N^4 , N^3 , N^2 , N に比例させて設定している。さらに図9として各レベルによる線縁雑音付加パターンの例を示す。

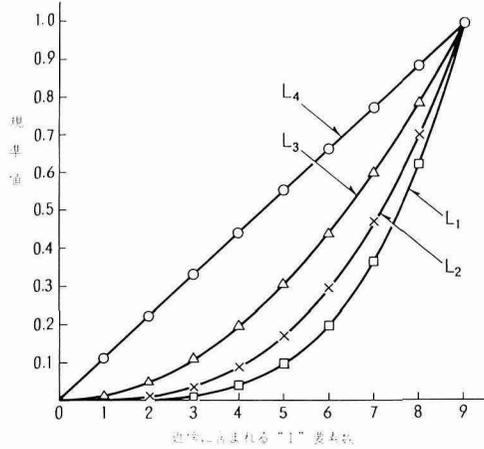


図8 線縁雑音発生の規準値 (4レベル)

(2) 線縁雑音による影響とその除去

文字線の細線化を行なう場合、従来の輪廓要素除去する方法では必然的に文字線縁に存在する雑音により影響を受け、雑音部分がひげとなって残ったり、直線部分のとび出しが生じたりすることになる。この点に対しては、雑音による影響を軽減するためにあらかじめ文字線縁のスムージングを行なっている。しかし、この処理のためには線縁のとび出しを検出するための多くのマスクが必要となり、決して簡便な雑音による影響の軽減化の方法とは言えない。これに対して、ここに示す方法は標準パターンの細線化の処理方法をそのまま用いたものである。しかし、このとき得られる細線化パターンは、線縁雑音の影響が本来得られるべき心線図形に対して1 bit (ここでは1標本点であらわされるパターンを1 bit のパターンと呼ぶ) ないし2 bits のひげとなって現れたものである。この理由は、心線図形を得るための重み付けが線幅の2倍程度の範囲を考慮して行なわれているため、心線図形を構成する要素が線縁雑音の有無にかかわらずに依然として高い値に重み付けされているからである。図10(a)に、図9

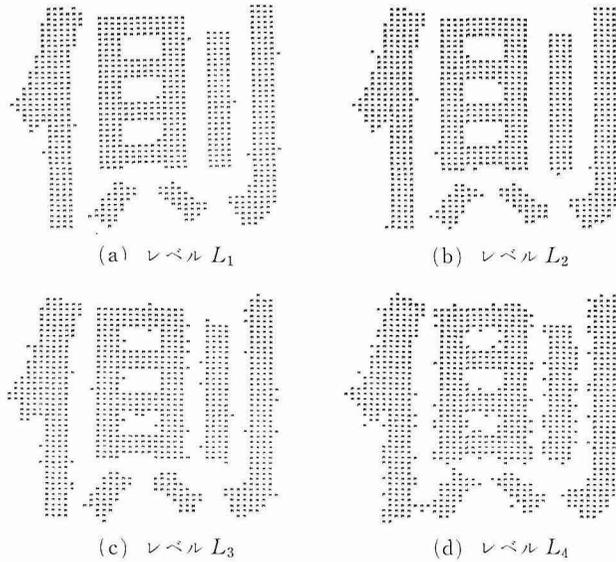


図9 線縁雑音付加パターンの例

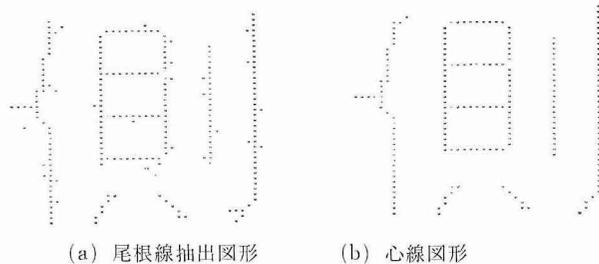


図10 線縁雑音付加パターンからの細線化の例

図10(a)に、図9

(c) に示したレベル L_3 の線縁雑音付加パターンに対する尾根線抽出図形を示す。したがって、この 2 bits 以下のとび出し部分を除去することで線縁雑音による影響を軽減した心線図形を得ることができる。このための具体的な方法は 3.3 節に関連して後述する。図 10 (b) に 同図 (a) から 2 bits 以下のとび出しを除去したパターンを示す。図 7 (c) に示した標準パターンから得られた心線図形と比較して大差ない結果が得られている。

3. 漢字パターンの特徴の抽出方法とその表現方法

3.1 認識に必要な特徴

文字認識をどのような方法で行なうかによって、認識に必要な情報すなわち文字の特徴を何に求めるかは異なってくる。パターンマッチング的な方法⁹⁾では、標準パターンとの相関の度合いをあらわす類似度という量が特徴となる。また、文字パターンの黒部と白部との関連から文字の凹凸によってできると考えられる場を特徴とする場合⁹⁾もある。ここでは印刷漢字（文字線の連結状態が定まっていれば手書漢字でもよい）を認識の対象とし、ストロークアナリシス的な方法で認識を行なう場合について考察する。

このような漢字を認識の対象としてみた場合、比較的直線的な文字線で構成されているパターンが非常に多いことに注目できる。つまり、漢字は直線線分がある点で相互に連結されて構成されていると考えることができる。このとき、直線線分はその方向から縦線分、横線分、斜め線分に分類でき、また連結の中心となっている点は特徴点と呼ぶべきもので、端点、屈折点、分岐点、交差点に分類される。したがって、文字線に関する特徴を長さや方向から文字線分として抽出し、さらに文字線分相互の連結状態を特徴点として抽出することで漢字を認識するうえに必要な情報を得ることができる。

3.2 文字線分の抽出方法

ここでは文字線に関する特徴の抽出方法について述べる。すなわち、文字線分を図 11 に示す 4 方向に量子化し、その長さと共に抽出する。方向の量子化には、図 12 に示すようなマスクを用いて各条件を満たす“1”要素をそれぞれ分類する方法があるが、こうした方法はパターンの局部

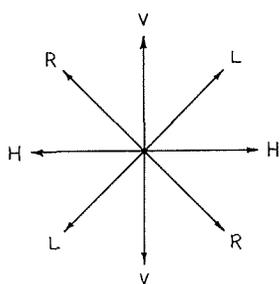


図 11 文字線の方向の量子化

にとらわれた方法であり、文字線の方向はかなり全体的なものであることを考えれば、必ずしも適切な方法とは言えない。これに対してより全体的な面から文字線分を抽出する方法として、縦横の二方向の走査による方法を示す。このときに走査の対象となるパターンは 2.1 節で述べた重み付けされたパターンである。このパターンは文字の心線図形となる要素に対して高い値をもつようなパターンであるから、図 13 (a) に示すような縦方向走査によって極大点を抽出することで得られるパターンは文字の心線図形のうち横線分となる成分であり、これを X パターンと呼ぶ。例えば 図 14 (a) のように重

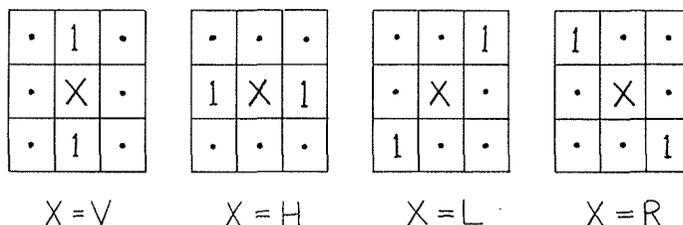


図 12 マスクによる文字線構成要素の分類

み付けされたパターンに対しては、 X パターンは同図(b)となる。同様に図13(b)に示すような横方向走査によって極大点を抽出することで縦線分となる成分が得られ、これを Y パターンと呼ぶ。図14(c)が Y パターンの例である。さて、ここで注目できることは X パターンと Y パターンに共通して含まれる成分があることである。この共通に含まれる成分によるパターンを S パターンと呼ぶ。このような S パターンはもとの文字パターンにおいて斜め線分を構成する成分であり、また屈折点、分岐点、交差点の特徴点の情報を含んだパターンである。先の例では図14(d)に示すパターンとなる。したがって、大まかには X パターンから S パターンを差引いたものは横線分であり、 Y パターンから S パターンを差引いたものは縦線分であり、 S パターンは斜め線分と特徴点であるということになる。さて、このような文字線分の抽出方法は第2章で述べた細線化処理を含んでいることがわかるであろう。すなわち、 X パターンと Y パターンの論理和をとることで文字の心線図形が得られるわけで、このように文字線の細線化と特徴抽出が同一の処理で実行できるということは認識方法として非常に有利なものであると言えよう。

さて、つぎに文字線の長さの測定を行なう。これは、 X 、 Y 、 S の各パターンにおいて連続する“1”要素を数えることで可能である。図15として図14に示した例に対応した測定結果を示す。こうして測定した長さをもとに表1に示す基準で各線分を SS 、 S 、 M 、 L の4種類の線分に分

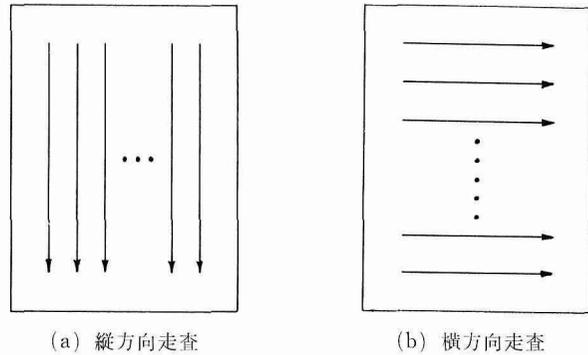


図13 文字線抽出のための走査方向

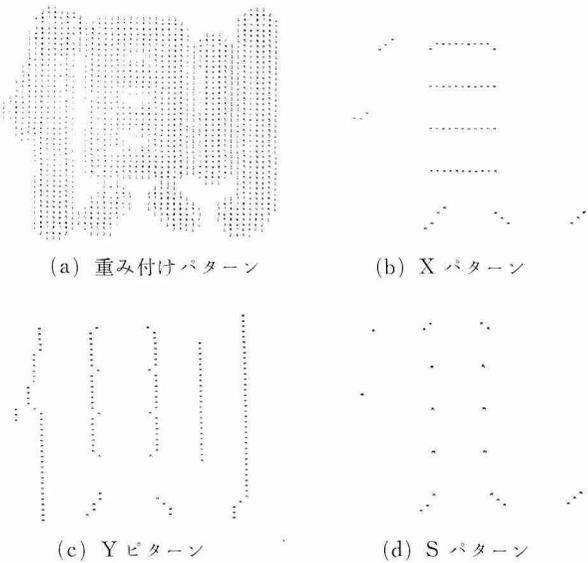


図14 縦、横方向走査による文字線の抽出

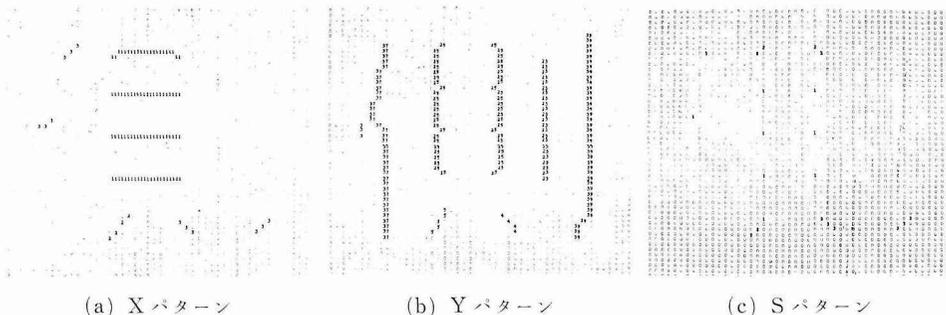


図15 X 、 Y 、 S パターンの線分長の測定

表 1 線分長による文字線の分類

分 類	SS 線 分	S 線 分	M 線 分	L 線 分
文字幅に対する割合	15% 未満	15% 以上 45% 未満	45% 以上 70% 未満	70% 以上
文字幅 45 bit における bit 数	1 ~ 7	8 ~ 20	21 ~ 32	33 ~ 45

類する。そしてさらにこの線分長による分類をもとに、 X, Y, S のパターンから縦、横、斜めの各線分を抽出する。ここで各線分には、 S, M, L 線分と分類されたものに対して方向を与えるものとし、 SS 線分に対してはほかの線分に含ませるか、またはそれ自体方向を与えずに処理することにする。 SS 線分に対してこのような処理を行なったわけは、線分長が短いために誤った方向の量子化がなされることを防ぐためであり、このことは誤識別およびリジェクトを減少させることになる。一般に S パターンにはこのような SS 線分が多く含まれることになる。具体的には、 SS 線分を含む X, Y パターンにおける線分を X, Y とすれば以下に示す処理を行なうことになる。

- (1) $X = SS, Y > SS$ …… SS 線分を縦線分に含ませる。
- (2) $X > SS, Y = SS$ …… SS 線分を横線分に含ませる。
- (3) $X = Y = SS$ …… SS 線分として方向を与えずに分類する。
- (4) $X > SS, Y > SS$ …… SS 線分を屈折点、分岐点、交差点とする。

(4)では特徴点として処理しているが、そのうち分岐点、交差点については次節で述べるように容易に抽出できるため、ここでは後述する方法では抽出できない屈折点が抽出できることが重要である。 S パターンに含まれるほかの S, M, L 線分はその方向により斜め線分として R, L 線分と分類する。また、 X, Y の各パターンから S パターンを除いたものをそれぞれ横線分、縦線分として H, V 線分と分類する。結局、各線分は $SS, H(S, M, L), V(S, M, L), R(S, M, L), L(S, M, L)$ というように 13 種類に分類して抽出することになる。

3.3 特徴点の抽出方法

ここでは、特徴点として端点、分岐点、交差点の各点を抽出する方法について述べる。そこで、まず細線化処理によって得られる心線図形を構成する“1”要素の分類方法について述べる。心線図形は幅が1の図形であるから、“1”要素がいくつかの方向に連続しているかを知ることはパターンの特徴をとらえるうえで有効である。そのために、任意の“1”要素 x_0 とその近傍の要素を図 16 のようにあらわす。標本化された二値図形に対して連結数³⁾なる指数が提案されているが、ここでは心線図形を取り扱うのに適当な量として、式 (17) で計算される N を用いる。

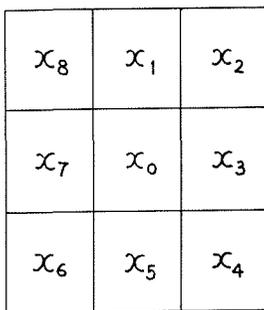
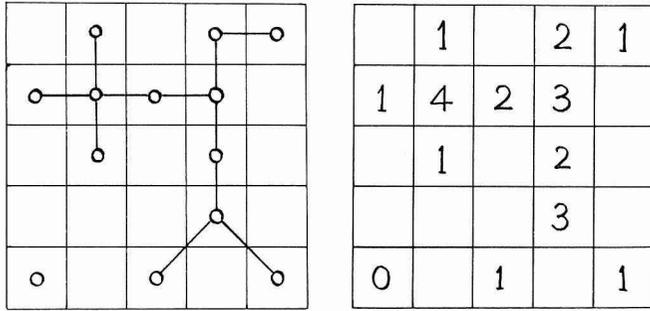


図 16 要素 x_0 とその近傍

$$N = \sum_{k=1,3,5,7} x_k + \sum_{k=1,3,5,7} \bar{x}_k x_{k+1} \bar{x}_{k+2} \tag{17}$$

ただし、 x_k は図 16 に示す x_0 の近傍の要素であり、同時に各要素の値をあらわす。また \bar{x}_k は $(1-x_k)$ をあらわす。結局、 N により各“1”要素の連続する方向の数がわかり、これによって“1”要素を以下のように分類することができる。 N を用いた分類例を図 17 に示す。これにより、 N による分類が実によく我々の直観と一致することがわかるであろう。結局、前節での結果とあわせて図 18 に示すような特徴の抽出ができることになる。



(a) 心線図形 (b) “1”要素のNによる分類
 図17 心線図形を構成する“1”要素の分類

- N = 0 孤立点
- N = 1 端点
- N = 2 連結点 (屈折点を含む)
- N = 3 分岐点
- N = 4 交差点

2.2節で線縁雑音による影響とその除去について述べたが、先に述べたNにより雑音により生じたひげを除去することができる。つまり、このようなひげの先端はN=1として端点に分類されるから、こうした“1”要素に注目し、その近傍にN=3またはN=4となる“1”要素があれば、この端点は除去すべきものであることがわかる。同様に2 bitsのひげを除くことができる。

3.4 漢字の表現方法

漢字パターンの特徴として文字線分が長さや方向により分類され、各特徴点が抽出された。したがって、ここではこれらの特徴により漢字を表現する方法について示す。そこで、まず任意の漢字パターンKをその漢字が含むV, H, R, L, SSの各線分の数要素とする5次元ベクトルであらわす。さらに各線分を長さによる分類にしたがってS, M, L線分の数要素とする3次元ベクトルであらわす。そして最後にS, M, L線分に含まれる特徴点を組にしてあらわす。こうした表現を図18に示した例について示せば以下のようになる。

$$\begin{aligned} \langle K \rangle &= \langle V, H, R, L, SS \rangle = \langle 5, 4, 0, 0, 3 \rangle \\ \langle V \rangle &= \langle S, M, L \rangle = \langle 0, 3, 2 \rangle \\ \langle M \rangle &= \langle (1, 1), 2(2, 3, 3, 2) \rangle \\ \langle L \rangle &= \langle (1, 1), (1, 3, 1) \rangle \\ \langle H \rangle &= \langle S, M, L \rangle = \langle 4, 0, 0 \rangle \\ \langle S \rangle &= \langle 2(2, 2), 2(3, 3) \rangle \\ \langle SS \rangle &= \langle (3, 1), 2(1, 1) \rangle \end{aligned}$$

ところで、こうした表現は各線分や特徴点についての位置情報を積極的に利用していないため、必ずしも1対1に対応して表現できない漢字がある。しかし、このような例はごく少数の漢字に対してであると予想されることから、これに対して救済手段を特別に用いても問題とはならないであろう。こうした問題よりも、漢字パターンを3層に分けて表現することによる利点の方が大きいと考えられる。すなわち、漢字それに含まれる文字線分をもとに第1層で表現したことで、文字数が極めて多い漢字に対して考慮すべきものをわずかな範囲に限定して処理できるわけであり、こうしたことから認識能率が良くなることが期待できるのである。

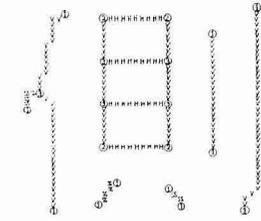


図18 漢字パターンからの特徴抽出

4. あとがき

漢字をストロークアナリシスの幾何学的特徴抽出法により認識するための前処理として、文字線のぼかし処理による細線化の方法を提案し、文字線を長さや方向により分類し、分岐点、交差点などの特徴点を抽出し、これらにより漢字を三つの層に分けて表現する方法について論じた。本論文で述べたような方向での認識では、文字線分をいかに安定に抽出、分類し、特徴点を抽出できるかに問題がある。したがって、対象となる漢字は文字線の連結が明確な印刷漢字ないし丁寧に手書された漢字が望ましい。ここで述べた方法は文字線の細線化と同時に文字線分に関する特徴が抽出される点に特徴がある。漢字の表現方法については、実際の認識の問題を考えた場合、各層においてどの程度対象となる漢字を限定できるかが問題であり、この点に関しては具体的に検討しなければならないであろう。

参 考 文 献

- 1) 鹿野, 鳥脇, 福村: “濃淡図形を線図形に変換する方法” 信学論(D), 55-D, 10, p. 668 (昭47-10).
- 2) 小川, 手塚: “漢字パターンとその表現および処理系について” 信学会オートマトンと言語-パターン認識と学習研資, AL 72-128, PRL -129 (昭48-01).
- 3) 横井, 鳥脇, 福村: “標本化された二値図形のトポロジカルな性質について” 信学論(D), 56-D, 11, p. 662 (昭48-11).
- 4) 中野, 安田, 中田: “周波数領域での階層的パターン整合法による漢字認識” 信学論(D), 58-D, 2, p. 94 (昭50-02).
- 5) 森見徳, 森俊二, 山本: “場の効果法による特徴抽出” 信学論(D), 57-D, 5, p. 308 (昭49-05).