

地図からの文字の切り出し

志 久 修[†] 姉 川 正 紀^{††}
中 村 千 秋^{††} 中 村 彰^{††}

地図など、複雑な背景の中に埋もれて散在する任意の定形文字を切り出すには、文字位置の予測、背景との区別、文字の欠損や背景との接触等解決しなければならない難問が多く、有効な一般的手法はほとんど報告されていない。本文では、任意の線図形を対象とした従来の Merilin-Farber-Hough変換法を改善し、上記の諸問題を克服する方法を提案した。すなわち、(1)テンプレートとして「文字枠図形」を採用し、従来文字ごとに多数必要であったテンプレート数を、同一ポイント数の文字についてはただ一つだけで、すべての文字が切り出せるようにして高速切り出しを可能にするとともに、(2)投票方法を画素単位から画素集合単位を行うように改善して投票処理を高速化し、さらに、(3)投票結果から文字中心点を抽出する方法を考案して文字候補数を絞り込んだ。本方法ではテンプレートと原画像との適合度(黒画素密度)を抽出基準とする点は従来のMF法を踏襲しているので、背景との接触等にも強いという特徴は保存されている。本方法を用いて、14枚の地図画像に含まれる310個の文字の切り出し実験を行った結果、文字候補数を全画素数の1%弱に絞り込んだとき、抽出成功率92%を得、本方法の有効性を確かめた。なお、文字候補の数と抽出成功率のバランスをとることが必要であり、そのための適切な抽出パラメータ値の選定の目安も明らかにした。

Character Extraction from Map Image

OSAMU SHIKU,[†] MASANORI ANEGAWA,^{††} CHIAKI NAKAMURA^{††} and AKIRA NAKAMURA^{††}

This paper describes a method for extracting regular-form-characters with complicated background in maps. The method is based on Merilin-Farber-Hough transformation method fundamentally but with significations modifications. Those are, firstly, so called "a character frame block" is used as a template, which contributes to reducing the number of (hundreds or thousands) templates to only one, and hence a processing time is reduced remarkably. Secondly, the voting scheme is improved from in each-pixel-base to in pixels-block-base, which also contributes to reduction of processing time. Thirdly, a criterion for finding the center of a character from the voting results is introduced, which contributes to reduction of total number of character candidates extracted. The method applied to character extraction of 14 maps (310 characters). As the results, the number of candidates is reduced to 1% of total pixels, and 92% of characters are found correctly in character candidates.

1. はじめに

本文は、地図画像のような様々な図形が混在している画像の中に含まれる文字を切り出す方法について述べる。従来、このような画像からの文字の切り出しは、文字の位置が予測できない上に、文字と周辺図形との区別が容易につけにくい、さらには文字が雑音等によって欠損したり周辺の図形と接触したりする等難しい問題が多く、キーとなる特定文字を指定する方法¹⁾等があるだけで、有効な一般性のある手法は殆ど

報告されていない。複雑な背景に埋もれて画像のどこに存在しているかわからない任意の文字を抽出するには、通常のテンプレートマッチング法は、SSDA法²⁾のような高速化アルゴリズムを用いたとしても適当ではない。なぜなら、すべての文字(数百ないし数千)のテンプレートを用意し、その各々に対して全画素数と同じ回数だけマッチング操作を繰り返す必要がある、多大の処理時間がかかってしまうからである。

一般に地図中の文字は数種類の大きさ(ポイント数)の印刷文字(斜体を含む活字)が用いられるが、一連の情報は同一ポイント数の印刷文字で印字されている。したがって、地図中の文字を抽出するには、指定のポイント数の文字(これを定形文字という)がすべて切り出せればよい。しかし、地図の場合、様々な

[†] 佐世保工業高等専門学校電子制御工学科
Faculty of Engineering, Sasebo College

^{††} 長崎大学工学部電気情報工学科
Department of Electrical Engineering and Computer Science, Faculty of Engineering, Nagasaki University

図形が文字とともに混在しており、文字だけを的確に切り出すことは困難である。

そこで、次善の策として、文字らしい部分をすべて文字の“候補”（その中には文字でないものも含まれている）として切り出し、その候補の中の文字の特定は別の文字認識処理に任せるという2段階の方法をとる。本論文は、その前半の“文字候補の切り出し”に関するものである。

文字候補の切り出しに要求されることは、抽出すべき文字が正しく含まれていること、および候補の数が少ないことである。

一般に特定ポイント数の定形文字では、文字に外接する四辺形の大きさが字種によらず一定であるという特徴を持つ。この四辺形の枠の内部がすべて黒画素で満たされ、枠外はすべて白画素であるような図形を特に文字枠図形と呼ぶ。本文では、「定形文字は、文字枠図形の中から文字部分のみを残してそれ以外の部分を侵食（白画素化）した図形」としてとらえる。すなわち、図1に示すように、黒画素で満たされた文字枠図形（図a）が侵食され（図bの斜線部）、たとえば“あ”などの定形文字になると考える。ここで、 L_x 、 L_y は、それぞれ定形文字の外接四辺形の水平、垂直方向の長さである。

これより、定形文字の切り出しは、定形文字（侵食された文字枠図形）から元の文字枠図形を抽出することにより行うことができる。本文では、従来線図形の抽出に用いられている Merilin-Farber-Hough 変換³⁾（以下 MF 法）を、文字枠図形のような面積を持つ図形の抽出に拡張し、元の文字枠図形の抽出、すなわち定形文字の切り出しを行う。

さらにこの際、2.4 および 2.5、2.6 節で詳述するように、次のような工夫を行った。

- ①MF 法で最も処理時間がかかる投票処理の高速化
- ②多数検出される投票値のピーク点から、真の文字中

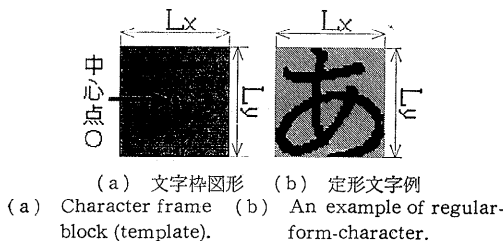


図1 文字枠図形と定形文字
Fig. 1 Character frame block and regular-form-character.

心点の検出方法

以下本文では、最初に従来の MF 法の概要と特徴を紹介した後、本論文の定形文字切り出し方法と処理手順を述べ、さらに、実際の地図中の文字の切り出しに適用した結果について報告する。

2. 定形文字の切り出し法

2.1 MF 法の概要

MF 法は、原画像と同じ大きさの配列（初期値 0）および抽出したい線図形のテンプレートを用意し、つぎのように行う³⁾（図2参照）。

- (1) テンプレート上の 1 点に基準点 O を設定する。図(a)
- (2) テンプレートを 180 度回転させる。これを転置テンプレートと呼ぶ。図(b)
- (3) 原画像中のすべての黒画素について転置テンプレートの基準点 O を一致させながら順次移動させ、転置テンプレートの軌跡（図(c)の実線部分）となる配列要素に 1 を累積加算（投票という）する。このとき、図(c)からわかるように、転置テンプレートの基準点 O が、原画像中の抽出したい線図形（点線図形）上にある場合は、必ず転置テンプレートの軌跡の 1 点 O' を通る。この結果、配列の投票累積値は基準点 O' において最大値をとる。
- (4) 配列の投票累積値が最大値をとる点を見つける。
- (5) この点がテンプレートの基準点と最も一致する点となる。

この手順により、任意形状の線図形が抽出できる。

この方法で、投票配列の値はその位置にテンプレートの基準点があるときのテンプレートと原画像との適合度を表している。そのため、投票値が高いほど原画

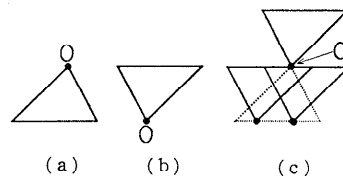


図2 MF 法の原理図

Fig. 2 Illustration of MF method.

- (a) Original template, (b) Rotated template, (c) Voting method.
line: rotated template.
dot line: original sample pattern.
(The rotated template moves along the dot line, accompanying with voting.)

像とテンプレートが一致していると考えられる。MF法では、この投票値をもとに図形の抽出を行っているため、接触・欠損等の雑音に対しては頑強な方法となっている。また、原画像中の黒画素のみについて処理を行うので、通常のテンプレートマッチングのように画像全面について照合を行う必要がなく、高速に行える特徴がある。ただし、この方法では、複数の図形（または文字）を抽出したい場合には、その数だけテンプレートを用意し、上記の処理を繰り返し行う必要がある。

2.2 MF法の拡張

本論文で提案する定形文字の切り出し方法では、MF法のテンプレートを線図形から面図形に拡張している。すなわち、侵食雑音を受けた文字枠図形から元の文字枠図形を抽出するため、テンプレートとして図1(a)に示す文字枠図形を用いる。また、基準点の位置は、MF法の手順(2)の処理を省くためテンプレートの中心点Oに設定する。これより、テンプレートと原画像の適合度を表す投票値は、文字枠領域内における原画像の黒画素数すなわち黒画素密度を表し、その値は定形文字の中心点で最大となる。この場合にも、面積をもつ文字枠図形のテンプレートと画像の適合度を抽出基準とするため、雑音に強いという性質は失われない。また、黒画素についてのみ処理を行うので高速に行うことができる。そのため、図3のような線接触雑音を受けた定形文字も、高速に抽出できる。本方法によれば、文字の種類数に関せずテンプレートの数は、1種類ですむことになる。

2.3 処理手順

定形文字切り出し法は、前述のテンプレートを用いて以下の手順で行う。

- ①原画像からの投票配列作成
- ②投票配列中の極大点抽出
- ③文字中心点候補抽出
- ④文字候補領域の切り出し



図3 接触雑音を受けた定形文字
Fig. 3 Regular-form-characters with complicated background.

以下、各手順について述べる。

2.4 原画像からの投票配列の作成

MF法では、前述したように原画像中の黒画素に沿ってテンプレートの基準点を一致させながら順次移動させ投票を行っていく。つまり投票を画素単位で行っているため、画像中の黒画素数に比例して投票回数が多くなってしまふ。そこで、投票単位を画素単位から画素集合単位にすることにより投票回数の削減を行う。この例として、線分を投票単位とする方法が提案されている⁴⁾。しかし、今回対象とする文字枠図形は面図形であるため、線分単位の場合、線幅分の画素についての投票を行うことができない。そのため、ここでは投票単位として、一括投票処理の容易さも考慮し、ランレングス、4進木表現した黒画素ブロック（以下4進木ブロックと呼ぶ）の2種類の画素集合を用いる。これらは原画像中のすべての黒画素情報を忠実に保存しているので、MF法での投票結果と全く同じ結果が得られる。

これらを投票単位とした場合と画素単位の場合の投票回数を比較する。

まず、MF法での投票回数 T_0 は、画像中の総黒画素数を M 、テンプレートとする文字枠図形の大きさを $L_x \times L_y$ とすると

$$T_0 = M \times (L_x \times L_y) \quad (1)$$

となる。

次に投票単位をランレングス単位とした場合の投票回数 T_1 は、ランレングスが p ($p=1, 2, \dots$) になる回数を t_p とすると、

$$T_1 = L_y \times \sum t_p \{L_x + (p-1)\} \quad (2)$$

となる。ただし $\sum t_p \times p = M$ である。

同様に、投票単位を4進木ブロック単位とした場合の投票回数 T_2 は、4進木の最下位ノードの黒画素ブロックが $2^q \times 2^q$ ($q=0, 1, \dots$) になる回数を t_q とすると、

$$T_2 = \sum t_q \{(L_x + (2^q - 1)) \times (L_y + (2^q - 1))\} \quad (3)$$

となる。ただし $\sum t_q \times (2^q \times 2^q) = M$ である。

以上の投票回数 T_1 、 T_2 を MF 法での投票回数 T_0 と比べると、それぞれ次のようになる。

$$T_1/T_0 = \{ \{L_x - 1\}/L_x \} \cdot (\sum t_p/M) + 1/L_x \quad (4)$$

$$T_2/T_0 = \{ (L_x + L_y - 2)/(L_x L_y) \} \sum t_q 2^q/M \\ + \{ L_x L_y - (L_x + L_y + 1) \} / (L_x L_y) \} \sum t_q/M \\ + 1/(L_x L_y) \quad (5)$$

上の二つの式より(4)式では $\sum t_p/M$ が、(5)式では、 $\sum t_q/M$ が小さくなるほど、投票回数は少なくて済

むことがわかる. この $\Sigma t_p/M$ は画像中の平均ランレングスの逆数, また $\Sigma t_q/M$ は4進木ブロックの平均サイズの逆数を表している. よって, 投票回数は, 画像中の平均ランレングスが長いほど, あるいは4進木ブロックの平均サイズが大きいほど少なくなる. なお, 両場合で最も多くの投票処理を必要とするのは, それぞれ平均ランレングスが1, 平均4進木ブロックサイズが1×1の時, その投票回数はMF法での投票回数 T_0 と同じである.

2.5 投票配列中の極大点抽出

投票値である黒画素密度は, 原画像中の定形文字の中心点と文字枠図形の基準点が一致したときに極大値をとる. 一般的にその形状は, 定形文字の大きさにより図4ようになる. (1)は文字外接四角形と文字枠図形の大きさが一致している場合の極大値の形状を表している. ほとんどの定形文字ではこのような形状になる. また(2)は文字外接四角形の水平あるいは垂直方向の大きさが文字枠図形より小さい場合である. また, (3), (4)は, (1), (2)の場合の定形文字が, かすれなどによりその外接四角形が小さくなった場合である. これらの極大値の抽出は, 図5に示す5×5画素のマスクを用いて行う. つまりこのマスクを投票配列全面について移動させ, マスクの中心点とその近傍での投票値が, 式(6)から式(9)のいずれかの条件(極大値形状条件)を満足する点を求める.

$$d_0 > \sum_{i=1}^8 \frac{d_i}{8} \text{ かつ } d_0 > d_j \quad (j=1, 3, 5, 7) \quad (6)$$

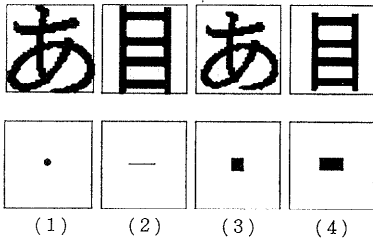


図4 定形文字の極大値形状

Fig. 4 Shape of black picture cell density's peak.

12		11		10
	4	3	2	
13	5	0	1	9
	6	7	8	
14		15		16

図5 極大点抽出用マスク

Fig. 5 Mask for extracting peak.

$$d_0 > \sum_{i=1}^8 \frac{d_i}{8} \text{ かつ } d_0 > d_j \quad (j=2, 4, 6, 8) \quad (7)$$

$$d_0 > \sum_{i=9}^{16} \frac{d_i}{8} \text{ かつ } d_0 > d_j \quad (j=9, 11, 13, 15) \quad (8)$$

$$d_0 > \sum_{i=9}^{16} \frac{d_i}{8} \text{ かつ } d_0 > d_j \quad (j=10, 12, 14, 16) \quad (9)$$

なお, d_x は投票配列上の任意の点にマスクの中心点があるときの, x (図5のマスク参照) の位置の黒画素密度を表している.

この処理で黒画素密度が極大となる点が特徴点として多数抽出される. これらは文字枠図形の特徴のみを用いて取り出した点なので, 文字枠中心点群と呼ぶ.

2.6 文字中心点候補抽出

文字枠図形の特徴のみで抽出した文字枠中心点群には, 文字でない点が非常に多く含まれている. そこで文字が持つ一般的な特徴を用いて, 文字枠中心点群から真の文字中心点候補を絞り込む. その特徴として投票値(黒画素密度)を用いた次の二つの基準を用いる.

(1) 文字を構成する黒画素密度

文字を構成する黒画素密度は, ある範囲に限定できるので, 次の条件を用いる.

$$\alpha_{\min} < d < \alpha_{\max} \quad (10)$$

ただし, d は注目している点の黒画素密度, $\alpha_{\min}$, $\alpha_{\max}$ は, それぞれ文字を構成する黒画素密度の最小値, 最大値とする.

(2) 黒画素密度の変化率

黒画素密度の変化率は, 図6に示すように文字枠図形 ($L_x \times L_y$) 内での黒画素密度 d と, 同じ中心点を持つそれより大きい文字枠図形 ($L_x' \times L_y'$; これを拡大文字枠図形という) 内での黒画素密度 d' を用いて次式で定義する.

$$k = (d - d')/d \quad (11)$$

この値 k は, d に比べ d' が小さいほど大きな値を持つ. 一般的に注目している点が文字の中心点である場合, d' は d より十分小さくなるため, k は大きな値を持つ. そこで, しきい値 β を設定して, k が次の条件を満たす場合, その点を文字中心点候補とする.

$$k \geq \beta \quad (12)$$

ここで, 文字枠図形内に含まれる黒画素数を N , 拡大文字枠図形内に含まれる黒画素数を N' , 二つの文字枠図形の差の部分に含まれる黒画素数 (つまり接触している黒画素数) を n とすると, 黒画素密度はそれぞれ次式で表される.

$$d = N/(L_x \times L_y) \quad (13)$$

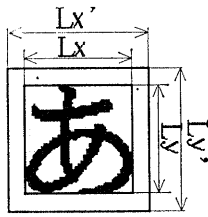


図 6 黒画素密度の変化率

Fig. 6 Difference of black picture cell density.

$$\begin{aligned} d' &= N' / (L_x' \times L_y') \\ &= (N + n) / (L_x' \times L_y') \end{aligned} \quad (14)$$

これらの式を, (11), (12) 式に代入すると, 次式のように, 文字枠図形内の黒画素数 N に応じた接触許容画素数が決定される.

$$n \leq \left\{ \frac{L_x' L_y'}{L_x L_y} (1 - \beta) - 1 \right\} N \quad (15)$$

これより, 黒画素数 N が増えるに従い, 接触許容画素数も直線的に増える. また, 黒画素数 N が同じ場合でも, しきい値 β が小さくなるに従い, 接触許容画素数が増える. 例えば, $L_x = L_y = 21$ 画素, $L_x' = L_y' = 23$ 画素とした場合の接触許容画素数は, $N = 100$ 画素のとき, $\beta = 0.075$ で 10 画素, $\beta = 0.11$ で 6 画素である.

2.7 文字候補領域の抽出

文字候補領域の抽出は, 前節で求めた文字中心点候補に対し, その点を中心とする文字枠図形領域を, 原画像から強制的に切り出すことにより行う.

なお, この結果切り出された文字候補領域には, 文字以外の図形 (模様) も含まれており, 本当の文字かどうかの最終的な判定は, 別の認識処理で行うことになる. 本文では文字の切り出しのみを対象とする.

3. 地図中の文字抽出実験

今回提案した方法を用いて地図中の文字の切り出しを行う.

3.1 地図画像と文字

今回の実験では, 国土地理院発行の二万五千分の一地形図を処理対象地図とする. この地形図では, 居住地を表すのに 4 種類の大きさの文字 (1.5, 2.0, 2.5, 3.0 mm 四方) が用いられている⁵⁾. このなかの最小文字である 1.5 mm 四方の文字は, 市街地等の名称を示すのに用いられており, 図形的には複雑な図形が密に表現されている部分に記入してある. そのため, 多くの接触雑音を受け, 4 種類の文字の中で, 最も切



図 7 原画像

Fig. 7 An original map image.

り出しが困難な文字である. そこで, 今回はこの 1.5 mm 四方の大きさの文字を切り出し対象文字として実験を行う.

地形図の市街地領域をイメージスキャナにより解像度 320 dpi で読み込んだ 2 値画像 (512 画素 $\times$ 512 画素) を原画像とする. 実験では原画像として 14 枚の画像を用意する. この中には, 310 文字の切り出し対象文字が含まれている. 原画像の一例を図 7 に示す.

3.2 実験定数の設定

(1) テンプレートの大きさ

テンプレートの大きさは, 文字の大きさが 21 画素 $\times$ 21 画素であったので, $L_x = L_y = 21$ 画素とする.

(2) 黒画素密度の範囲

予備実験として, 原画像以外の文字について黒画素密度を測定した結果, すべての文字の黒画素密度が, 0.1 から 0.7 の範囲になったため, 実験では, $\alpha_{\min} = 0.1$, $\alpha_{\max} = 0.7$ に固定する.

(3) 黒画素密度の変化率のしきい値 β

この値は, 比較に用いる拡大文字枠図形の大きさや, 切り出そうとしている文字にどれくらいの接触雑音まで許容するかを考慮して設定する. 実験では, 拡大文字枠図形の大きさは, 文字枠図形を 1 画素だけ太らせた大きさ ($L_x' = L_y' = 23$ 画素) とする. これは, 文字枠を大きくとりすぎると, 近傍の図形まで含んでしまうので, 真に文字枠図形に接触した図形だけを扱うためである. また, しきい値 β は, 文字枠図形と拡大文字枠図形の二つの黒画素密度が等しくなる $\beta = 0$ から, 接触画素数が 0 となる $\beta = 0.166$ までの範囲と

し、候補数と次節で定義する抽出成功率とを調べる。

3.3 文字抽出成功率

今回の実験では、文字抽出成功率を次式で定義し、切り出しがどの程度うまくいったかの評価基準とする。

文字抽出成功率 = (正しく切り出した文字数 / 全切り出し対象文字数)

ここで、文字が正しく切り出されたかどうかの判定は、抽出された文字中心点候補群中の点が、真の文字中心点（人手により作成）の近傍領域に存在するかどうかで行う。近傍領域は、注目点からユークリッド距離で2画素以内の範囲とする。これは、この程度のズレならば文字認識処理で十分許容できると思われるからである。

4. 実験結果と考察

今回提案した定形文字切り出し法を14枚の原画像について適用し、その有効性を調べた。

4.1 投票回数の比較

MF法での投票回数と、2.4節で提案したランレングス単位、4進木ブロック単位での投票回数の比較結果を表1に示す。これより、画素単位の投票（MF法）、4進木ブロック単位、ランレングス単位の順で、投票回数が少ないことがわかる。これは、原画像が主に線図形で構成されているため、表からもわかるようにランレングスの場合が最もよく原画像中の黒画素をまとめた形となるためである。この投票回数は処理速度に比例するので、投票回数が少ないほど、高速に投票配列が作成できることを意味する。よって、ランレングス単位の投票が最も高速であるといえる。

4.2 文字抽出実験結果

文字抽出実験を、

方法A；2.5節の方法のみで抽出を行った場合。

方法B；Aに2.6節(1)の方法を併用した場合。

方法C；Bに2.6節(2)の方法 ($\beta=0.075$) を併用した場合。

の三つの場合について行った。図7の原画像を含みランダムに選んだ14枚の原画像を対象としたときの結果を表2に示す。

表1 投票回数の比較（処理速度の改善効果）
Table 1 Comparison of voting time.

MF法	ランレングス/MF法 (平均ランレングス)	4進木/MF法 (平均ブロック)
1.0	0.32 3.54画素	0.65 1.56×1.56画素

表2 実験結果
Table 2 Results of experiment.

方 法		A	B	C
抽出された 文字候補数	総 数 (原画像 1 枚 当たりの平均 値)	17455	16694	2500
	絞り込み率 (対全画素数) 512×512画素	6.67%	6.36%	0.95%
候補として 抽出できな かった文字	14原画像 310 文字中の 全抽出失敗文 字数	5*1	5*1+4*2 =9	5*1+4*2 +16*3 =25
	抽出成功率	98.4%	97.1%	92.0%

*1: 方法Aによる結果 (図8)

*2: 方法Bの併用による増分 (図9)

*3: 方法Cの併用による増分 (図10に一例を示す)



図8 未抽出文字（文字枠図形の特徴のみ）
Fig. 8 Unextracted characters.
(the case of using character frame block feature)

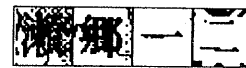


図9 未抽出文字の増分（黒画素密度の特徴）
Fig. 9 Increased unextracted characters.
(the case of using character frame block feature and black picture cell density feature)



図10 未抽出文字の増分の例 ($\beta=0.075$ のとき)
Fig. 10 Examples of increased unextracted characters (parameter $\beta=0.075$).

なお、抽出失敗例の原因を分析すると、図8では文字枠図形周辺に存在している雑音の影響で黒画素密度が極大値形状条件（式(6)～(9)）を満たさなかった場合であった。図9では入力時に生じたかすれやつぶれ、さらに接触雑音により黒画素密度が条件（式(10)）を満たさなかった場合であった。図10では接触画素数が、式(15)によって決定される許容数を越えてしまった場合であった。

表2より、A→B→Cと文字抽出基準を厳しくするにつれて抽出される文字候補の数は減少するが、候補の中に抽出漏れとなった文字の数は微増することがわ

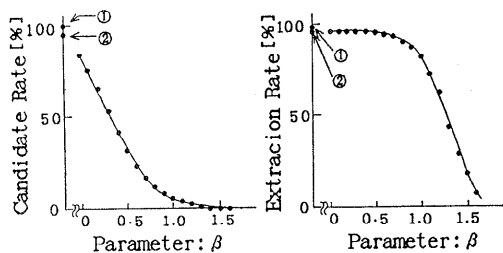


図 11 1画面当たりの平均文字候補数
Fig. 11 Number of character candidates per a map image.

図 12 文字抽出成功率
Fig. 12 Extraction rate of characters.

かる。

ちなみに β の値に対する抽出される文字候補の原画像 1 枚当たりの平均数、および 14 画面の全抽出成功率の変化を図示するとそれぞれ図 11、図 12 のとおりである。図 11 において文字候補数は、方法 A での候補の数（文字枠中心点数）に対する比率（候補数比率）で表している。両図において、①は、2.5 節のみで求めた文字枠中心点数（方法 A）の結果を示し、②は、2.6 節(1)で述べた黒画素密度を併用した場合（方法 B）の結果を示している。

これらの図から抽出される文字候補の数と抽出成功率の関係は相反関係にあり、両者のバランスを考慮して最適な β の値を決める必要がある。実際には、候補数比率が 14.3%、抽出成功率が 92.0% となる $\beta=0.075$ 近傍が適当な値と思われる。

4.3 定形文字の切り出し例

$\beta=0.075$ とした場合の、図 7 の画像から切り出した文字候補領域の一部を図 13 に示す。これにより文字領域が良好に抽出されていることがわかる。なお、この図では文字部分の切り出し結果のみを表示しているが、この他にも 3960 点の候補を抽出している。

また、原画像から文字中心点候補を抽出するまでにかかった処理時間の比率（原画像 14 枚での平均値）を表 3 に示す。これより、処理時間の大半が投票処理で費やされていることがわかる。なお、この投票方法

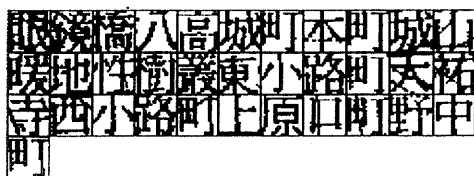


図 13 図 7 からの切り出し例
Fig. 13 Examples of extracted character from Fig. 7.

表 3 処理時間の比率
Table 3 Ratio of processing time.

処 理 名	処理時間比率 [%]
投票処理（ランレングス単位）	88.0
文字枠中心点抽出処理	7.4
文字中心点抽出処理	4.6

は、最も高速に行えるランレングス単位の投票方法を用いた。

4.4 実験のまとめ

実験結果より本方法を用いることにより、雑音を受けた文字も良好に切り出せることがわかった。また、次のような欠点も判明した。

- ①文字候補を多数抽出してしまう。
- ②抽出できない文字がある。

まず①は、切り出しに黒画素密度を基本とした文字図形共通の低レベルの特徴しか用いていないため、原画像中にこれらの特徴を満たす部分が多数存在してしまうためである。しかし、これらの中には、ほとんどの文字が含まれているので、認識処理により文字だけを取り出すことは可能である。この場合、画像全体について認識処理を行う場合に比べ、1%弱の手間で実現できる。さらに、新しい文字特徴を設定することにより、候補数の削減が可能と考えられる。

また②は、本方法が文字枠図形をもとにして文字切り出しを行っていることに起因する。つまり、文字枠付近の雑音がひどいと、定形文字としての特徴がなくなり切り出しができなくなる。これは、文字枠図形と定形文字の形状が著しく違う場合（例えば漢数字の“一”など）に顕著になる。

これらの欠点のある程度解決する方法としては、文字以外の黒画素（道路、家屋記号等の図形）をあらかじめ前処理により除去することが考えられる。また、これは黒画素数が減ることになるので、処理速度も向上すると考えられる。

さらに本方法の本質的な限界として、本方法では定形文字の切り出しにしか利用できないことがあげられる。すなわち、手書き文字のような大きさが揃っていない文字の切り出しには利用できない。しかし、今回対象とした地形図をはじめ印刷図面中の文字を認識したいというニーズは大きく、それらを対象とした場合の方法としては十分利用できると考えている。

5. む す び

本文では、拡張した MF 法とそれを応用した定形文字切り出し方法を提案した。

本方法の主な特徴は次のとおりである。

(1) 「文字枠図形」をテンプレートとして用いるよう従来の MF 法を拡張し、文字ごとに多数必要であったテンプレート数を、ただ一つだけで、すべての同一ポイント数の定形文字が切り出せるよう改善し、処理の高速化を図った。

(2) 文字枠図形(テンプレート)と原画像の適合度(黒画素密度)を抽出基準として文字切り出しを行うことにより、複雑な背景をもつ画像(たとえば地図画像など)に含まれる定形文字を良好に抽出できる。

(3) MF 法で最も時間のかかる投票処理を、ランレングス、または4進木ブロックの画素集合単位の投票方法を用いることにより高速化した。

(4) 投票結果(黒画素密度の特徴)を用いた文字中心点の抽出方法を考案し、文字候補数を絞り込んだ。

本方法の性能を評価するため、14枚の地図画像に含まれる310個の文字の切り出し実験を行い、次の結果を得た。

(1) 抽出成功率: 92.0%

(2) 候補数の削減効果:

抽出候補の数=全画素数の約1%弱に減少

(3) 投票回数(処理時間の主要因子)の削減効果:

ランレングス単位/画素単位 $\approx$ 0.32倍

4進木ブロック単位/画素単位=0.65倍

なお、抽出不成功の事例を分析すると、強度の接触雑音やかすれ、つぶれ等により、定形文字(文字枠図形)としての特徴を失った場合であった。

抽出成功率と候補数とは、トレードオフの関係にあり、どの程度接触雑音を許すかに関して適切な抽出基準値を選ぶ必要がある。

今後は、切り出した文字の効率的な認識方法を検討する予定である。

参 考 文 献

- 1) 長尾智晴, 安居院猛, 中島正之: 地図画像に対する文字抽出と修復処理, 1989 信学秋季全大, SD-

9-5 (1989).

- 2) 高木幹雄, 下田陽久(監修): 画像解析ハンドブック, 東京大学出版会 (1991).
- 3) Merilin, P.M. et al.: A Parallel Mechanism for Detecting Curves in Pictures, *IEEE Trans. Comput.*, Vol. C-24, No. 1, pp. 96-98 (1975).
- 4) 渡辺孝志, 石戸橋真: 線分近似による一般化ハフ変換の高速化と任意図形検出, 信学論(D-II), Vol. J 74-D-II, No. 8, pp. 995-1003 (1991).
- 5) 大竹一彦: 二万五千分の一地図, 古今書院 (1990).

(平成3年10月28日受付)

(平成4年11月12日採録)



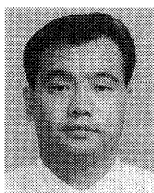
志久 修 (正会員)

昭和41年生。平成元年長崎大学工学部電子工学科卒業。平成3年長崎大学工学部修士課程修了。現在、佐世保工業高等専門学校助手。画像処理およびパターン認識の研究に従事。電子情報通信学会会員。



姉川 正紀

昭和42年生。平成2年長崎大学工学部電子工学科卒業。平成4年同大大学院修士課程修了。現在、同大大学院海洋生産科学研究科在学中。ニューロコンピュータおよび画像処理の研究に従事。電子情報通信学会会員。



中村 千秋 (正会員)

昭和39年生。平成元年九州大学大学院修士課程修了。現在、長崎大学工学部助手。画像処理および計算機ネットワーク上のアプリケーションの研究に従事。



中村 彰 (正会員)

昭和6年生。昭和33年九州大学工学部通信工学科卒業。日本電信電話公社電気通信研究所を経て、現在、長崎大学工学部電気情報工学科教授。画像処理およびパターン認識等の研究に従事。工学博士。電子情報通信学会会員。