

# 情景画像中文字認識のための変形を許容する高速な テンプレートマッチング

根岸 和也<sup>†</sup> 岩村 雅一<sup>††</sup> 大町真一郎<sup>†</sup> 阿曾 弘具<sup>†</sup>

<sup>†</sup> 東北大学大学院工学研究科 〒 980-8579 仙台市青葉区荒巻字青葉 6-6-05

<sup>††</sup> 大阪府立大学大学院工学研究科 〒 599-8531 大阪府堺市学園町 1-1

E-mail: <sup>†</sup>{negishi,machi,aso}@aso.ecei.tohoku.ac.jp, <sup>††</sup>masa@cs.osakafu-u.ac.jp

あらまし 情景中の文字を認識するためには、一般に前処理として文字切り出しが必要であるが、既存の手法では文字列を成さない「孤立文字」や隣接した2文字が接触した接触文字などを切り出せないという問題がある。また、情景画像中の文字画像は射影変換などの変形を受けることも知られている。そこで本論文では、射影変換や字形の違いに対する頑健さを持ち、孤立文字や接触文字など、既存の手法では切り出せない文字の切り出しが可能でかつ高速な手法を提案する。提案手法を実際の画像に適用したところ、射影変換や字形の違いに対して頑健であり、孤立文字である時計の文字盤等や、数式中の接触文字を良好に、また単純なテンプレートマッチング手法より高速に切り出せることを確認した。

キーワード 孤立文字, 接触文字, 情景画像, 文字変形, テンプレートマッチング

## Fast Deformable Template Matching for Character Recognition in Scene Images

Kazuya NEGISHI<sup>†</sup>, Masakazu IWAMURA<sup>††</sup>, Shinichiro OMACHI<sup>†</sup>, and Hirotomo ASO<sup>†</sup>

<sup>†</sup> Graduate School of Engineering, Tohoku University  
6-6-05 Aoba, Aramaki, Aoba, Sendai, 980-8579 Japan

<sup>††</sup> Graduate School of Engineering, Osaka Prefecture University  
1-1 Gakuencho, Sakai, Osaka, 599-8531 Japan

E-mail: <sup>†</sup>{negishi,machi,aso}@aso.ecei.tohoku.ac.jp, <sup>††</sup>masa@cs.osakafu-u.ac.jp

**Abstract** To recognize characters in scene images, a character segmentation method is generally required for pre-processing. However, there are problems that traditional methods are not able to segment “isolated character” which does not constitute a character string and “touching character” which is connected to other characters. Moreover, it is known that character images in scene images suffer from perspective distortion. In this paper, we propose a new fast character segmentation method of characters including the isolated and the touching ones with robustness for perspective distortion and character deformations. By applying the proposed method to scene images, it is confirmed that the method is robust against perspective distortion and character deformations, and the method is superior in computational time against a simple template matching method.

**Key words** isolated character, touching character, scene images, character deformation, template matching

### 1. ま え が き

情景中に存在する文字は有益な情報源であり、非常に利用価値が高い。しかし、文字情報を利用するために必要な文字認識技術にはまだまだ課題が多い[1]。例えば、情景中の文字は必ずしも真正面から撮影されるとは限らないため、射影変換の形で

文字画像が歪むことが知られている。また、情景中には様々な字形の文字があるため、複数の字形が認識できなければならない。しかし、このような課題の中で最大のものは文字領域を的確に抽出することの難しさであると思われる。

例えば、文献[2]では画像中の局所的な複雑度を利用して文字領域の候補を算出し、一度文字列を抽出した後に文字を抽出

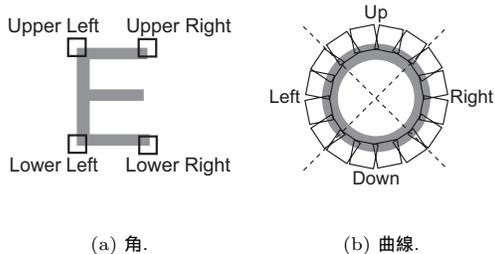


図 1 用いる特徴.

する.しかし,一度文字列を抽出する手法では文字列を成さない「孤立文字」や,複雑なレイアウトの数式中の文字を切り出すことができない.また,文字切り出しの際に連結成分を用いる手法もあるが,連結成分に着目した手法では,隣接した2文字が接触した接触文字を切り出すことができない[3].

このような問題は変形を許容するテンプレートマッチングであれば解決できると考えられる.画像全体に対するテンプレートマッチングを高速に行う手法としてはアクティブ探索法[4]が知られている.しかし,これは一致検索であり,前述した射影変換や字形の違いなどで生じる文字画像の変形を許容する方法ではない.

このため,我々は文字の持つ特徴点を利用して,孤立文字や接触文字に対して有効な文字切り出し手法[5]を提案したが,射影変換に対する頑健性や処理時間に課題があった.そこで本論文では射影変換に対して頑健な特徴を用い,特徴点から代表点を選び用いることで,高速な処理を実現するテンプレートマッチング法を提案する.提案手法は変形を許容するため,射影変換だけでなく字形の違いなどで生じる文字の変形を許容する能力がある.また,テンプレート数の増加に対する計算量の増加量が少ない.そのため,テンプレート数が一定以上であれば,提案手法の計算量は変形を許容しない単純なテンプレートマッチングを下回るという優れた特徴を持つ.さらに提案手法は従来の文字切り出し手法とは異なり,個別文字認識の手法ほどではないが,一定の文字認識能力がある.そのため,提案手法は文字の切り出しと認識の大分類を一度に行うことができる.

## 2. 特徴の探索

提案手法は特徴に基づくテンプレートマッチングを行う.本節では提案手法の特徴探索方法について述べる.

### 2.1 入力画像と参照画像

本論文では,切り出したい文字を含む画像を入力画像,辞書にあたる個々の文字の画像を参照画像と呼ぶ.参照画像としては,認識対象の文字画像をあらかじめ用意しておく.

### 2.2 特徴

本論文では局所的画像の特徴として,文字の判別に有効であり,探索にも適していると考えられる角と曲線を用いる.角は図 1(a)に示す4種類を用い,曲線は図 1(b)に示す16種類の特徴を抽出する.特徴を抽出する局所的画像(以下では領域と呼ぶ)の大きさを  $p \times p$  とし,次節で述べる特徴量を用いて抽

出する.本論文では  $p = 7$  とした.

### 2.3 特徴量: エッジの方向を利用したヒストグラム

ある領域  $R (p \times p)$  の特徴量として,エッジの方向を利用した正規化ヒストグラムを考える.まず,画像中の各画素におけるエッジの方向と強度を Sobel Filter で算出しておく.その際,雑音の影響を排除するため,エッジの強度が一定値以下の画素に「エッジを持たない画素」というラベルを付けておく.

領域  $R$  の中で,エッジの方向を幅  $\theta = \frac{2\pi}{n}$  ラジアンに  $n$  分割し,同じ区間に割り当てられたエッジを持つ画素の数をその区間の度数とするヒストグラムを作成する.ただし,あらかじめ「エッジを持たない画素」とラベル付けされた画素はヒストグラムの作成に利用しない.そして,ヒストグラムの度数の総和が1になるように正規化したものを領域  $R$  を特徴付ける  $n$  次元特徴量とする.この特徴量を以下「エッジヒストグラム」と呼ぶ.

角と曲線のエッジヒストグラムの各区間は

$$\left[ -\frac{\pi}{n} + \frac{2\pi}{n}t, \frac{\pi}{n} + \frac{2\pi}{n}t \right), \quad 0 \leq t \leq n-1 \quad (1)$$

で与えられる.範囲を  $\frac{\pi}{n}$  ずつずらしているのは垂直なエッジと水平なエッジを安定に検出するためである. $n$  について考えると,フォントの異なる文字の特徴点も柔軟に検出するためには  $n$  を小さくしたほうがよい.一方,複雑な構造を持つ特徴を検出するためには  $n$  を大きくする必要がある.そのため,特徴の複雑さに応じて  $n$  を変え,角は  $n = 4$ , 曲線は  $n = 16$  としてエッジヒストグラムを構築する.

エッジヒストグラムの類似性判定のための尺度として,Swainら[7]が提案した類似度(対応する要素の最小値の和)を用いる.類似度の最大値は1,最小値は0となる.各特徴を表すエッジヒストグラム  $M$  と領域  $R$  のエッジヒストグラムの類似度を求め,類似度が閾値  $T_1$  以上のとき領域  $R$  の左上の画素を特徴  $M$  の検出箇所(特徴点)とする.ここで,  $T_1$  は  $0 \leq T_1 \leq 1$  の範囲で定める.曲線の特徴については,求めた16種類を図 1(b)に示すように4種類に分類して用いる.これは射影変換の影響を受けると角度が変化してしまうため,その変化を吸収するために細かく分類した後に4つに再分類している.

## 3. 変形を許容するテンプレートマッチングによる文字切り出し

### 3.1 参照画像に対する処理

以下の処理では,角の特徴4種類,曲線の特徴4種類を用いる.

参照画像の特徴を登録するため,参照画像から全ての特徴点を抽出し,特徴点の種類と位置(後述する「投票ベクトル」)を記録する.画像中から抽出される特徴点は多量に検出されるため,全ての特徴点を対象にすると莫大な計算時間がかかる.そこで,図 2 に示すように検出された特徴点のあるかたまりに代表点を定める.全ての  $(2q+1) \times (2q+1)$  ピクセルの領域中の特徴に対して,中心の画素の類似度が最大であれば,その画素を代表点とする.また代表点には,代表点の近傍に存在した特徴点の数を投票する際の重み  $W$  (次段落で詳しく述べる)と

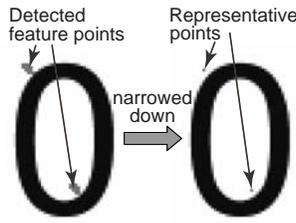


図 2 特徴点の代表点の計算 .

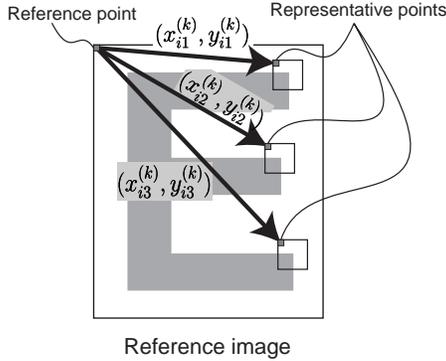


図 3 参照画像における投票ベクトルの算出 .

して割り当てる . これにより , 代表点を用いて投票を行っても , 全ての特徴点を用いて投票する時と遜色ない結果を出力することが可能となる .

参照画像の特徴点について , 図 3 に示すように参照画像の左上の画素を基準点とし , 代表点の位置を基準点から代表点へのベクトルで表す . このベクトルを「投票ベクトル」と呼ぶ . 図 3 は右上の角の特徴の代表点が 3 箇所検出され , それぞれの代表点に対応する投票ベクトルを表している . 以下 , 投票ベクトルを表す記号を次のように導入する . まず準備として , それぞれ 4 種類ある角と曲線の特徴に通し番号  $i$  を振り , 「 $i$  番目の特徴」として参照する .  $1 \leq i \leq 8$  である . また , 字種  $k$  の参照画像から検出された  $i$  番目の特徴の代表点の個数を  $N_i^{(k)}$  とおき , 特徴  $i$  の代表点に 1 から  $N_i^{(k)}$  までの番号  $j$  を振る . 基準点を原点としたときの  $j$  番目の代表点の位置を投票ベクトル  $(x_{ij}^{(k)}, y_{ij}^{(k)})$  で表す ( $1 \leq j \leq N_i^{(k)}$ ) . この  $j$  番目の代表点に割り当てられた投票重みを代表点が代表する特徴点数  $W_{ij}^{(k)}$  で表す . 投票ベクトルと投票重みの組の集合が字種  $k$  を表す特徴になる .

### 3.2 入力画像に対する処理

文字切り出しの処理は投票ベクトルに依存する処理としない処理に分けられる . 投票ベクトルに依存しない処理は入力画像 1 枚につき 1 度行うだけでよいが , 投票ベクトルに依存する処理は参照画像の枚数分だけ行う必要がある . 以下 , それぞれの処理の詳細について述べる .

#### 3.2.1 投票ベクトル非依存の処理

特徴検出位置の変化に頑健なテンプレートマッチングを行うため , 特徴検出位置の近傍も特徴検出位置とみなす . 入力画像から全ての特徴点を検出し , 特徴点のかたまりを代表点で表す . 入力画像内の各点  $p = (x, y)$  に特徴  $i$  が存在するかどうかを表

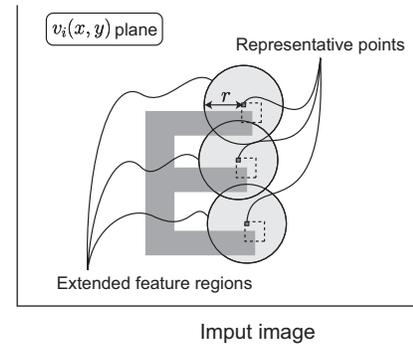


図 4 入力画像における特徴点の抽出 .

すテーブル  $\{v_i(x, y)\}$  を用意し , 図 4 に示すように , 各代表点から半径  $r$  画素の円内の点に 0 から 1 の値を , それ以外の点には 0 を与える . この円を特徴点領域と呼ぶ . 0 から 1 の値は次のように定める . 特徴  $i$  の  $j$  番目の代表点  $P_{ij} = (X_{ij}, Y_{ij})$  に対して ,  $y_j(x, y)$  を

$$y_j(x, y) = \begin{cases} \frac{\exp(-((x-X_{ij})^2+(y-Y_{ij})^2)/2)}{\text{Var}}, & \text{if } \|P_{ij} - (x, y)\| \leq r \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

と定め ,

$$v_i(x, y) = \max_j y_j(x, y) \quad (3)$$

とする . ここで  $\text{var}$  は , 代表点から距離  $r$  の点の投票度数が 0.85 になるように ,  $\text{Var} = 3.07r^2$  とした . 式 (3) の  $v_i$  は 3.2.2 で投票を行う際に用いられ , 異なる  $j$  に対する投票にはそれらの投票度数の最大値を用いることを意味する .

#### 3.2.2 投票ベクトル依存の処理

座標  $(x, y)$  が字種  $k$  の文字画像の基準点である可能性の高さを表す  $V^{(k)}(x, y)$  を次式で定める .

$$V^{(k)}(x, y) = \sum_i \sum_j W_{ij}^{(k)} v_i(x - x_{ij}^{(k)}, y - y_{ij}^{(k)}) \quad (4)$$

式 (4) は図 5 のように図 4 の  $v_i(x, y)$  面を平行移動し , 用いた投票ベクトルの投票重みを掛け合わせて投票することを意味する .  $v_i(x, y)$  の最大値は 1 であるので ,  $V^{(k)}(x, y)$  の最大値  $V_{\max}^{(k)}$  は

$$V_{\max}^{(k)} = \sum_{i=1}^8 \sum_{j=1}^{N_i^{(k)}} W_{ij}^{(k)} \quad (5)$$

となる . そこで , 参照画像と入力画像中の文字画像の類似度を  $V^{(k)}(x, y)/V_{\max}^{(k)}$  と定める . 文字画像間の類似度の閾値  $T_2 (0 \leq T_2 \leq 1)$  を定め ,

$$V^{(k)}(x, y) \geq V_{\max}^{(k)} T_2 \quad (6)$$

を満たす  $(x, y)$  が字種  $k$  の基準点であると判定する .

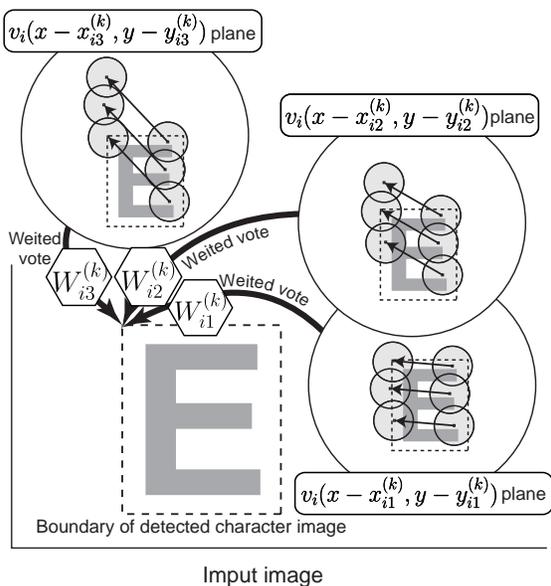


図 5 切り出し位置を特定するための投票 .

### 3.3 提案手法の計算量

大きさが  $m \times m$  の参照画像が  $C$  枚あり、入力画像の大きさが  $n \times n$  であるとする .  $p \times p$  画素の特徴を  $s$  種類用いると、入力画像からの特徴点抽出は単純なテンプレートマッチングであるので、必要な計算量は  $O(sp^2n^2)$  である . また、入力画像から代表点が  $T_{in}$  個検出されたとすると、半径  $r$  ピクセルの近傍に投票するために必要な計算量は  $O(r^2T_{in})$  である . 字種  $k$  の参照画像から代表点が  $T_{re}^{(k)}$  点検出されたとすると、類似度の計算に必要な計算量、参照画像 1 枚当たり  $O(n^2T_{re}^{(k)})$  である . したがって、 $C$  字種を切り出すときの提案手法の計算量は  $O(sp^2n^2 + r^2T_{in} + Cn^2T_{re})$  である . ここで  $T_{re}$  は  $T_{re}^{(k)}$  の平均値である .

ところで、入力画像の代表点の個数 ( $T_{in}$ ) は一般に入力画像の大きさ ( $n^2$ ) と比べるとはるかに少ない . したがって、類似度計算の際は代表点とその近傍点 ( $r^2T_{in}$ ) のみに行えば、提案手法の計算量は  $O(sp^2n^2 + r^2T_{in} + Cr^2T_{in}T_{re})$  に削減できる . 本論文の実験ではこの方法を用いる .

## 4. 従来手法との比較

### 4.1 単純なテンプレートマッチングとの比較

提案手法と単純なテンプレートマッチングとの性能面での違いは変形を許容するかどうかである . しかし、提案手法はテンプレート数  $C$  の増加に対する計算量の増加量が少ないため、テンプレート数が一定以上であれば、変形を許容しない単純なテンプレートマッチングより少ない計算量で、変形を許容するマッチングを実現できる .

単純なテンプレートマッチングの計算量は、参照画像の大きさを  $m \times m$ 、入力画像の大きさを  $n \times n$  とすると、1 テンプレート当たり  $O(m^2n^2)$  で、 $C$  個のテンプレートのマッチングに必要な計算量は  $O(Cm^2n^2)$  である . また、クラスタリングと差分計算を行う場合には、クラスタ数を  $c$  とすると、1 文字当たりの計算量は  $O(cmn^2)$  となり、 $C$  文字では  $O(Ccmn^2)$  と

なる . これは一般に提案手法の計算量よりも大きくなると考えられ、5.5 の実験でこのことを確認する .

### 4.2 一般化 Hough 変換や MAP 整合法との比較

基本的に変形を許容しない図形切り出し方法としては、一般化 Hough 変換 [8] や MAP 整合法 [9] が知られており、MAP 整合法を文字認識に使用した例もある [10] .

提案手法は一般化 Hough 変換に以下のような変更を加え、変形を許容するようにした手法であると解釈できる .

(1) 一般化 Hough 変換ではエッジ上の各点の角度に応じて投票するのに対し、提案手法では検出された特徴の種類に応じて投票を行う .

(2) 一般化 Hough 変換は基準点にのみ投票する<sup>(注1)</sup>が、提案手法は変形を許容するため、基準点の近傍にも投票する .

(3) 一般化 Hough 変換では投票度数の最大値は不定であるが、提案手法では最大値が式(5)で与えられる . そのため、参照画像と入力画像中の文字画像の類似度 (式(4)) を定義することができる .

これらの処理のうち (2) の処理には時間がかかる . しかし、この処理はテンプレート数に依らず 1 度実行するだけで済むため、本論文ではこのことに着目し、変形を許容しない手法に比べて 1 文字当たりの処理時間を大幅に減少させた .

## 5. 実験

提案手法の有効性を確認するため、(1) 射影変換を受けた文字、(2) 参照画像とはフォントが異なる文字、(3) 孤立文字、(4) 接触文字、の 4 種類の文字画像を対象に認識実験を行った . 参照画像には Arial を用い、入力画像には Arial の他に太字の Arial Black、斜字体の Franklin、角が丸い丸ゴシックを用いた . なお、参照画像と認識対象の文字が全く同じ場合の認識率は 100%であった .

低解像度の画像に対しても良好な検出結果を得るために、特徴の検出を行う前にあらかじめ平滑化フィルタを参照画像、入力画像の両方に適用した . 特徴検索での類似値は閾値  $T_1$  以上になった箇所を特徴点の検出箇所とした . 投票の際は、得票割合が閾値  $T_2$  以上の領域を文字の切り出し箇所とした . また、予備実験により特徴を絞り込む際の範囲を  $q = 5$  とした .

提案手法では、方向成分を持たない「エッジを持たない画素」を除いた領域で各方向成分の比を特徴としているため、探索領域中の大部分がエッジを持たない画素の場合にはわずかなノイズでも特徴が大きく変化してしまい、信頼性に欠ける . そのため、探索領域がエッジを持たない画素が 7 割以上含む場合は、その箇所での特徴検出は行わない .

### 5.1 射影変換を受けた文字画像に対する認識性能

参照画像を  $150 \times 150$  ピクセルの画像の中心に配置し、4 隅の  $x, y$  座標をそれぞれ 0 または  $\delta$  ( $\delta = 5, 10, \dots, 50$ ) だけ変位させた射影変換を施し、各  $\delta$  について  $2^{4 \times 2}$  個のテストパター

(注1): 一般化 Hough 変換でも離散化誤差を許容するために近傍にも投票することがあるが、変形を許容するためではないため、近傍の範囲は大きくないと考えられる .

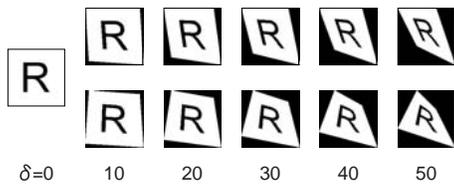


図 6 射影変換を受けた画像 .

$\delta$	認識率 (%)	$\delta$	認識率 (%)
5	100	30	100
10	100	35	100
15	100	40	99.9
20	100	45	99.9
25	100	50	99.7

表 1 射影変換に対する頑健性

フォント	Correct	Match	Miss
Arial Black	3 (5%)	59 (95%)	0 (0%)
Franklin	1 (2%)	61 (98%)	0 (0%)
Maru Gothic	1 (2%)	61 (98%)	0 (0%)

表 2 フォントの異なる文字に対する認識結果

ンを生成した (図 6 参照) . 生成したパターンを入力画像とし、変換前の (同じ字種の) 参照画像を用いたときの認識率を表 1 に示す . パラメータは  $T_1 = 0.75$ ,  $T_2 = 0.7$ ,  $r = 4\delta$  とした . その結果,  $\delta = 35$  までの射影変換に対して認識率 100% を達成することができ,  $\delta = 50$  の時でも 99% 以上の認識精度を示していることから,  $r$  を適切に設定する限りは提案手法は射影変換の文字に対して非常に頑健であるといえる .

## 5.2 フォントの異なる文字に対する認識性能

Arial の文字を参照画像とし, 太文字である Arial Black, 斜体字である Franklin, 丸みを帯びている丸ゴシック体の文字を認識対象として認識実験を行った . 入力画像には各フォントの英数字 62 文字が並んでいる画像を用いた . それぞれの画像には, 英数字 62 文字が含まれている . パラメータは  $T_1 = 0.75$ ,  $T_2 = 0.8$ ,  $r = 15$  とした .

実験結果を Correct, Match, Miss の 3 種類に分類して評価した . Correct は探索している文字と同じ文字のみを切り出した場合で, 文字の切り出しと認識が同時に行われたことに相当する . 入力画像中に探索する文字が複数ある場合は, それらの文字を全て切り出し, それ以外の文字を一切切り出さなかった場合のみ Correct とする . Match は探索する文字を切り出したが, 他の箇所も切り出している場合である . この場合, 切り出した全ての箇所に既存の個別文字認識手法を適用することで正しい認識結果を導くことができ, 切り出し手法としての役割は果たしていると考えられる . Miss は探索している文字を切り出すことが出来なかった場合である .

認識結果を表 2 にまとめた . Correct の数は 1 ~ 3 個程度であるが, 参照画像と同じ文字を切り出すことが出来た割合 (Correct と Match の和) は全てのフォントに対して 100% であり, フォントが異なるにも関わらず, 高い精度で切り出しに成功している . これにより, 提案手法はフォントが異なる場合で



図 7 時計の文字盤 10 字種に対する認識結果 : 参照画像に「2」を使用 .



図 8 道路標識に対する認識結果 : 参照画像に「4」を使用 .

も一定の切り出し性能を持つことが確認できた .

## 5.3 孤立文字に対する認識性能

孤立文字を含む時計の文字盤と道路標識の画像に対して認識実験を行った . 画像はグレースケールに変換してから用いた . パラメータは, 時計の文字盤には  $T_1 = 0.75$ ,  $T_2 = 0.87$ ,  $r = 15$  を用い, 道路標識には  $T_1 = 0.75$ ,  $T_2 = 0.93$ ,  $r = 15$  を用いた . 一般化 Hough 変換において平行移動, 拡大縮小, 回転に同時に対応するとき, 投票を段階的に行うことで計算時間を削減する方法が知られている [11] . 提案手法においても同様の手法で拡大縮小や回転に比較的短い計算時間で対応できると考えられるが, 現在は未実装であるため, 今回の実験では参照画像の大きさを認識対象に合わせることで対応した . 認識結果の一例を図 7 と図 8 に示す . 時計の文字盤に対しては, 参照画像に「2」を用いた時に, 対象となる文字を全て切り出し, 他の文字を一切切り出さなかったため Correct とし, 他の全ての数字で Match となった . ここで多くの文字が Match となった理由は, 文字盤の数字の字形が, 参照画像である Arial とは大きく異なっているため, 正しい文字の類似度が低くなってしまったためであると考えられる . 道路標識に対しては, 参照画像に「4」を用いた時に Match になった . その他の数字も, 切り出し可能であることを確認しているが, 対象文字以外の部分も切り出していた . この理由としては, 背景中の雑音が特徴として検出されたために, 投票度数が高められ, 結果として対象となる文字以外での箇所で検出されたためと考えられる .

これらの結果は, 提案手法が孤立文字を含む実画像に対しても一定の文字切り出し能力を有していることを示しているが, 一方で参照画像と字形が異なる文字に対する限界も示している . この場合, 複数の字形を用意しておくという解決策があるが, 参照画像の増加による計算時間の増加が問題になるため, これについては 5.5 で議論する .

## 5.4 接触文字に対する認識性能

文献 [3] から接触文字を含む画像を借用して認識実験を行っ

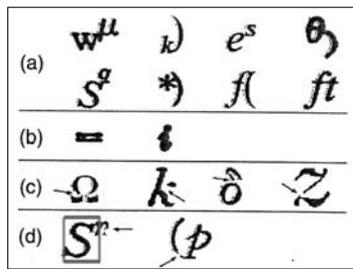


図9 接触文字に対する認識結果：参照画像に「S」を使用。

画像	画像の大きさ	総計算時間 (括弧内は参照画像が 1枚増えたときの平均増加量)	
		TM	提案手法
Arial Black	1234 × 927	309.3s (4.97s)	86.8s (1.33s)
Franklin	951 × 904	212.6s (3.41s)	65.2s (1.00s)
丸ゴシック	1204 × 781	245.8s (3.94s)	78.1s (1.20s)
時計の文字盤	655 × 606	92.2s (1.48s)	31.12s (0.48s)
道路標識	1280 × 1024	355.2s (5.70s)	108.87s (1.67s)
数式	749 × 558	98.4s (1.58s)	33.79s (0.52s)

表3 用いた画像の大きさと計算時間。TMはテンプレートマッチングを表す。表中の「道路標識」は参照画像の大きさを「4」に合わせたときの計算時間であり、表中の「数式」は参照画像の大きさを「S」に合わせたときの計算時間である。

た。実験で使用した参照画像は Arial の「S」である。パラメータは、 $T_1 = 0.75$ ,  $T_2 = 0.8$ ,  $r = 15$  を用いた。5.3 と同様に、参照画像の大きさを認識対象に合わせている。参照画像に「S」を用いたときの結果は Correct となり、その時の画像を図9に示す。実験の結果より、従来法では切り出しが困難であった接触文字に対しても提案手法は精度の高い切り出しを行うことができる。図9には大きさが異なる「S」が3文字含まれている。今回の実験では、最も大きい「S」に参照画像の大きさを合わせたために、他の「S」は切り出せなかった。そのため、今後は文字の拡大・縮小にも対応する必要があると考えられる。

### 5.5 計算時間の比較

4.1 で述べたように、提案手法の計算時間は単純なテンプレートマッチング手法よりも短いと考えられ、このことを実験により確認する。比較に用いるテンプレートマッチング手法は、2.3のエッジヒストグラムを特徴量とし、Swainら[7]の類似度を用いる。類似度計算の際には、既に計算した領域の類似度と新たに計算する領域の差分のみを計算する高速化を施してある。実験にはCPUがXeon 2.4GHzであるPCを用いた。

5.2から5.4で用いた入力画像に対する英数字62文字の計算時間を表3に示す。ここで参照画像1文字あたりの計算時間とは、提案手法の場合は、全体の処理時間から3.2.1で述べた特徴ベクトル非依存の処理時間を除いて参照画像の枚数で割った値である。テンプレートマッチングの場合は、同様に2.3で述べたエッジヒストグラムの計算時間を除いて参照画像の枚数で割った値である。どちらの手法も参照画像に対する処理は事前に済ませてあり、計算時間に含めていない。

両者の計算時間を比較すると、提案手法の全体の計算時間は

テンプレートマッチングのおよそ1/3であった。また、参照画像1枚当たりの計算時間はおよそ1/3.5であった。これらは、提案手法は変形を許容するにもかかわらず、変形を許容しない単純なテンプレートマッチング手法よりも高速に計算できることを示している。

## 6. まとめ

情景画像中の文字認識には様々な課題があるが、本論文ではそのうち、射影変換や字形の違いに対する頑健さを持ち、孤立文字や接触文字など、既存の切り出し手法では切り出せない文字を高速に切り出し、さらに文字認識の大分類も同時に行う手法を提案した。提案手法は、変形を許容するテンプレートマッチングの一種であるが、変形を許容しない単純なテンプレートマッチングよりも高速で、特に参照画像の増加に対する計算時間の増加が僅かである。

実験では、提案手法の射影変換、フォントの違いに対する頑健さと、孤立文字や接触文字などに対する切り出し能力、さらに変形を許容しない単純なテンプレートマッチングに対する計算時間における優位性を確認した。

文字画像の回転や拡大・縮小への対応、閾値をフォントや画像状態に合わせて動的に変更する手法の開発、さらに本論文で行った文字切り出し処理と連携する認識処理の開発は今後の課題である。

謝辞 貴重な助言を頂いた大阪府立大学の黄瀬浩一教授に感謝致します。本研究の一部は、科学研究費補助金 基盤研究(C) 16500096、ならびに若手研究(B) 17700205の補助を受けている。

## 文 献

- [1] 黄瀬浩一, 大町真一郎, 内田誠一, 岩村雅一, カメラを用いた文字認識・文書画像解析の現状と課題, 信学技報 PRMU2004-246, Mar., 2005.
- [2] 松尾賢一, 上田勝彦, 梅田三雄, “適応しきい値法を用いた情景画像からの看板文字列領域の抽出,” 信学論 (D-II), vol.J80-D-II, no.6, pp.1617-1626, 1997.
- [3] 内田誠一, 野村明弘, 鈴木昌和, 数学文書データベースの解析, 信学技報 PRMU2003-234, 2004.
- [4] 村瀬洋, V. Vinod, “局所色情報を用いた高速物体探索—アクティブ探索法—,” 信学論 (D-II), vol.J81-D-II, no.9, pp.2035-2042, 1998.
- [5] 岩村雅一, 根岸和也, 大町真一郎, 阿曾弘具, “特徴点探索による孤立文字認識” 画像の認識・理解シンポジウム (MIRU2005) 論文集, pp.559-566, 2005.
- [6] 森健一 (編), パターン認識, 電子情報通信学会, 1988.
- [7] M. Swain and D. Ballard, “Color indexing,” International Journal of Computer Vision, vol.7, pp.11-32, 1991.
- [8] D. Ballard, “Generalizing the Hough transform to detect arbitrary shapes,” Pattern Recognition, vol.13, no.2, pp.111-122, 1981.
- [9] 山田博三, “並列演算による方向パターン弾性整合法の提案—MAP 整合法—,” 信学論 (D-II), vol.J73-D-II, no.4, pp.553-561, Apr., 1990.
- [10] 澤宗昭, 山本和彦, 加藤邦人, Map マッチングによる複雑背景からの文字探索手法, 2002年信学総大, p.248, Mar., 2002.
- [11] 木村彰勇, 渡辺孝志, “図形検出力を向上させた高速一般化ハフ変換,” 信学論 (D-II), vol.J83-D-II, no.5, pp.1256-1265, May, 2003.