

孤立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[†] 岩村 雅一^{††a)} 大町真一郎[†] 阿曾 弘具[†]

Fast Deformable Template Matching for Isolated Character Recognition

Kazuya NEGISHI[†], Masakazu IWAMURA^{††a)}, Shinichiro OMACHI[†],
and Hiroto ASO[†]

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出しが必要であるが、既存の手法では文字列をなさない「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是射影変換の影響で文字画像がひずむことが知られている。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もち、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を切り出すための文字切出し手法を提案する。提案手法は変形を許容するテンプレートマッチング法の一つであるが、一定の条件のもとでは変形を許容しない単純なテンプレートマッチング手法の計算量を下回るという優れた特徴をもつ。また、テンプレートマッチングであるがゆえに文字切出しのみでなく、文字認識の大分類も同時に行うことができる。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切り出せることが確認できた。

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

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、射影変換等の影響を受け、文字画像がひず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。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。

情景画像中の文字を認識する一般的なアプローチは、文字画像領域を情景画像中から切り出し、切り出された文字画像に対して個別文字認識の手法を適用する2段階の処理である。そのため、文字認識性能は切出し性能に大きく依存する。ところが、文字認識の前提と

なる文字切出しには制約が多い。

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、1度文字列を抽出した後に文字を抽出する。しかし、1度文字列を抽出する手法では文字列をなさない「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。

そこで本論文では、情景画像中の文字切出しを頑健かつ高速に行うため、変形を許容する高速なテンプレートマッチング手法である特徴点投票法(FPV法、Feature Point Voting Method)を提案する。提案手法は局所的な特徴点を検出し、テンプレートから決まる投票ベクトルに従って投票してマッチング箇所を検出するもので、射影変換や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、提案手法はテ

[†] 東北大学大学院工学研究科, 仙台市
Graduate School of Engineering, Tohoku University, Sendai-shi, 980-8579 Japan

^{††} 大阪府立大学大学院工学研究科, 堺市
Graduate School of Engineering, Osaka Prefecture University, Sakai-shi, 599-8531 Japan

a) E-mail: masa@cs.osakafu-u.ac.jp

ンプレート数の増加に対する計算量の増加が少ない．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もつ．更に提案手法は従来の文字切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．本章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，情景画像等，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

文字認識で用いられる特徴としては，端点・分岐点・交点・屈折点等が検討されている [5]．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) に示すように 16 種類の特徴を抽出する．特徴は大きさ $p \times p$ の部分画像（領域と呼ぶ）に対して，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 $p \times p$ の領域 R の特徴量は，エッジの方向に関する正規化ヒストグラムとして作成する．

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 を用いて算出する．エッジの強度が一定値以下の画素に「エッジなし画素」のラベルをつけ，以後の処理ではノイズの影響を避けるために方向をも

たないとして扱う．エッジの方向は， $0 \sim 2\pi$ ラジアン の角度を n 分割した区間に分類する．各区間は，次で定める．

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

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．領域 R において，同じ区間に分類された「エッジなし画素」以外の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．この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 を特徴づける n 次元特徴量とする．これを「エッジヒストグラム」と呼ぶ．分割数 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 を小さくした方がよい．一方，複雑な構造をも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，検出したい特徴の複雑さに応じて n を変え，角検出には $n = 4$ ，曲線検出には $n = 16$ として 2 種類のエッジヒストグラムを作成する．

4 種類の角の特徴及び 16 種類の曲線の特徴を表すエッジヒストグラムは次のように定めた．右上角の特徴は 0 ラジアンと $\frac{\pi}{2}$ ラジアン方向のエッジからなるので， $[0.5, 0.5, 0, 0]$ というエッジヒストグラムで表される．一般的には，4 種類の角の特徴を表すエッジヒストグラムは式 (1) に示す 4 区間のうち， $t, (t+1 \bmod 4)$ 番目の区間の値を 0.5 に，他の区間の値を 0 にした 4 次元特徴量 ($t = 0, \dots, 3$ の 4 種類) である．同様に，16 種類の曲線の特徴を表すエッジヒストグラムは，16 区間のうち， $t, (t+1 \bmod 16)$ 番目の区間の値を 0.5 に，他の区間の値を 0 にした 16 次元特徴量 ($t = 0, \dots, 15$ の 16 種類) である．

2.4 エッジヒストグラム間の類似度

特徴であるかどうかを検証する領域を探索領域と呼ぶ．特徴と探索領域のエッジヒストグラム同士が類似しているとき，その探索領域の左上の画素を特徴点とする．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 [6] が提案した類似度を用いる． H を入力画像または参照画像中の探索領域のエッジヒストグラム， M をある特徴のエッジヒストグラム， n をエッジヒストグラムの次元数（角は 4，曲線は 16）とすると， H と M の類似度 S_{HM} は次式で求められる．

$$S_{HM} = \sum_{t=0}^{n-1} \min(H_t, M_t) \quad (2)$$

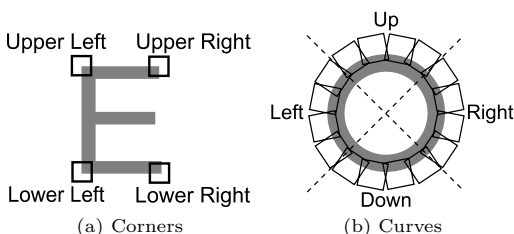


図 1 特徴
Fig.1 Features.

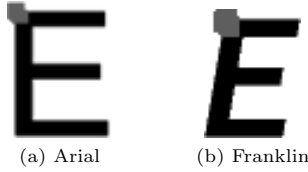


図 2 傾いたフォントに対する特徴抽出
Fig. 2 Extraction of a feature for a slant font.

ただし、 H_t と M_t はそれぞれ H と M の t 番目の区間の値である。エッジヒストグラムは和が 1 になるように正規化されているので、類似度の最大値は 1、最小値は 0 である。類似度が T_1 以上のとき探索領域 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、しきい値 T_1 を $0 \leq T_1 \leq 1$ の範囲で定める。

図 2 は左上の角の特徴の検出位置を示したものである。図 1 のような単純な特徴を用いることで、傾いたフォントに対する頑健な特徴抽出が可能になる。

曲線の特徴については、角の特徴に比べて、文字の傾きやフォントの違いに敏感であるため、同一文字の同一部分で異なる特徴が検出されることがある。この差異を吸収するため、特徴に対する投票処理（次節で述べる）では、求まった 16 種類を図 1 (b) に示すように、4 種類に分類して用いる。これにより、文字の傾きやフォントの違いなどに頑健になる^(注1)。

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出し

提案手法のアルゴリズムを Algorithm 1 に示し、以下に詳細を記す。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。

3.1 参照画像に対する処理

参照画像の特徴を登録するため、参照画像からすべ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」で表す）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、すべての特徴点をそのまま対象にすると膨大な計算時間がかかる。そこで、図 3 に示すように検出された特徴点のかたまりに代表点を定める。すべての $(2q+1) \times (2q+1)$ ピクセルの領域中の特徴点に対して、中心の画素の類似度（2.4 で定義）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、すべての特徴点を用いて投票する

Algorithm 1: 変形を許容する高速なテンプレートマッチング.

```

/* 参照画像に対する処理 */
1 すべての参照画像について投票ベクトル  $\{(x_{ij}^{(k)}, y_{ij}^{(k)})\}$  と、この投票ベクトルにおける投票重み  $\{W_{ij}^{(k)}\}$  を計算する。ここで  $k$  は参照画像の番号（字種番号）である。
/* 入力画像に対する、投票ベクトル非依存の処理 */
2 入力画像から 8 種類の特徴を探索し、検出された特徴点を代表点で表す。
3 特徴  $i$  が入力画像の座標  $(x, y)$  で検出されたかどうかを表す投票テーブル  $v_i(x, y)$  を用意し、代表点とその近傍（半径  $r$  の円内）には 0~1 の値を、それ以外には 0 を記録する。
/* 入力画像に対する、投票ベクトル依存の処理 */
4 for  $k \in \{\text{全字種 (テンプレート)}\}$  do
5   入力画像の座標  $(x, y)$  を基準点とする領域が字種  $k$  である可能性（類似度） $V^{(k)}(x, y)$  を投票ベクトル  $\{(x_{ij}^{(k)}, y_{ij}^{(k)})\}$ 、投票重み  $\{W_{ij}^{(k)}\}$  と、投票テーブル  $\{v_i(x, y)\}$  から計算し、しきい値以上であれば座標  $(x, y)$  に字種  $k$  の基準点があると判定する。
6 end
    
```

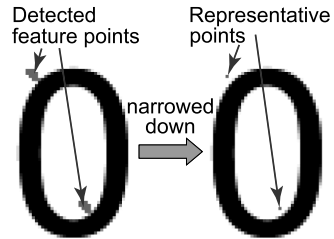


図 3 特徴点の代表点の計算
Fig. 3 Detected feature points of the upper left corner feature (left) are narrowed down to their representative points (right).

ときと遜色ない結果を出力することが可能となる。全参照画像の特徴点について、図 4 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 4 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。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

(注1): 角の特徴のヒストグラムは 4 分割するため、左右に $\frac{\pi}{4}$ ラジアン以内の文字の傾きであれば、同じ種類の特徴をほぼ同じ位置で検出することができる。それに対して曲線の特徴は 16 分割するため、文字の傾きが $\frac{\pi}{16}$ ラジアンを超えると別の特徴になってしまう。そこで 4 種類の特徴を同一視することで、角の特徴と同程度の頑健性を実現している。

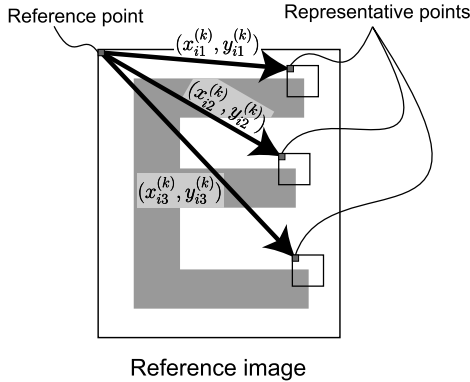


図 4 参照画像における投票ベクトルの算出
Fig. 4 Calculation of voting vectors in the reference image.

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 (代表点が代表する特徴点数) を $W_{ij}^{(k)}$ 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 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に、特徴点の位置にぼかしを加え、特徴点の近傍も特徴点であるかのようにみなす．すなわち、まず 3.1 で参照画像に行ったように、入力画像の特徴点を代表点で表す．そして、入力画像内の各点 (x, y) に特徴 i が存在するかどうかを表す得点を与え、テーブル $\{v_i(x, y)\}$ に格納する．この得点は、図 5 に示すように、各代表点の近傍、すなわち、半径 r 画素の円内の点に 0 でない値を与え、その外では 0 を与える．円内の各点には、代表点で 1、円周上の点では a ($0 < a < 1$) となる端の切れた正規分布の値を与える．ある点が複数の代表点の近傍に属するときはそれぞれから定まる得点の最大値とする．このように決められた得点を投票度数と呼ぶ．

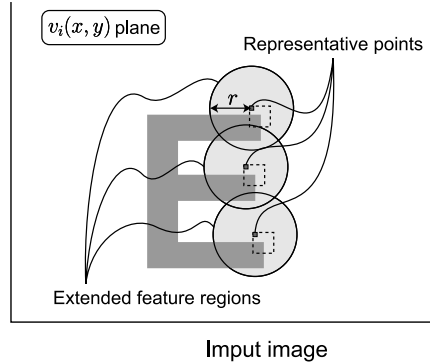


図 5 入力画像における特徴点の抽出
Fig. 5 Feature point extraction from the input image. Three representative points are found.

具体的には、各点の投票度数 $v_i(x, y)$ 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

$$y_j(x, y) = \begin{cases} \exp\left(-\frac{((x-X_{ij})^2+(y-Y_{ij})^2)}{2\text{Var}}\right), & \text{if } \|P_{ij} - (x, y)\| \leq r \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

と定め、

$$v_i(x, y) = \max_j y_j(x, y) \quad (4)$$

とする．ここで、半径 r の円周上の点で値を a とするため、 $\text{Var} = r^2 / (-2 \log a)$ と定めた．この a の値は予備実験により、 $a = 0.85$ とした．したがって、 $\text{Var} = 3.07r^2$ である．

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．

$$V^{(k)}(x, y) = \sum_i \sum_j W_{ij}^{(k)} v_i(x - x_{ij}^{(k)}, y - y_{ij}^{(k)}) \quad (5)$$

式 (5) は図 6 のように図 5 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

$$V_{\max}^{(k)} = \sum_{i=1}^8 \sum_{j=1}^{N_i^{(k)}} W_{ij}^{(k)} \quad (6)$$

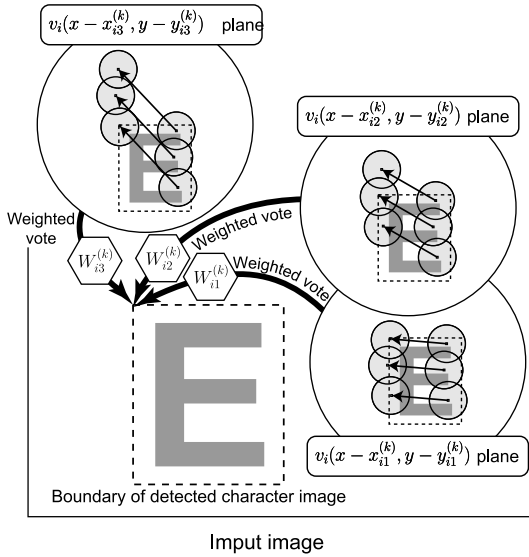


図 6 切り出し位置を特定するための投票
Fig. 6 Voting to determine the position of the character image.

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しきい値 T_2 ($0 \leq T_2 \leq 1$) を定め，

$$V^{(k)}(x, y) \geq V_{\max}^{(k)} T_2 \quad (7)$$

を満たす (x, y) が字種 k の基準点であると判定する．

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は参照画像 1 枚当り $O(n^2T_{re}^{(k)})$ 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． T_{re} は n, m には依存せず，参照文字に固有に決まる定数である．

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．これは入力画像のほとんどの点は特徴点にならないことと，3.1 で述べた代表点抽出の相乗効果による．したがって， r があまり大きくない場合は，類似度計算

を代表点とその近傍点 (r^2T_{in} 個) にのみ行えば，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，別のテンプレートマッチング手法として，カラー画像において色ヒストグラムを用いるテンプレートマッチング手法 [6] がある．この手法では， $m \times m$ の領域のヒストグラムを求めるとき，既に計算した領域のヒストグラムと新たに計算する領域のそれとの差分のみを実際に計算する．したがって，最初の色ヒストグラムの作成に $m \times m$ ステップかかるが，それ以降は $2m$ ステップで済むため，色空間を c 個のクラスターに分けた場合，類似度計算は c ステップで，1 文字当りの計算量は $O(cmn^2)$ 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小さくなると考えられる．5.6 の実験では，色の代わりにエッジの方向を用いたテンプレートマッチング法と提案手法を比較することにより，このことを確認する．

4.2 一般化ハフ変換との比較

基本的に変形を許容しない図形切り出し方法としては，一般化ハフ変換 [7] ~ [9] や，それと類似した処理を並列化した MAP 整合法 [10], [11] 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 [12]．

提案手法は一般化ハフ変換 [7] ~ [9] 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) 一般化ハフ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) 一般化ハフ変換は基準点にのみ投票するが，

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) 一般化ハフ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(6)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(5))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よらず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文字当りの処理時間の増加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。すべての実験で、参照画像には Arial フォントの英数字 62 文字を用いた。入力画像には Arial のほかに太字の Arial Black、斜字体の Franklin、角が丸い丸ゴシックを用いた。用いた各フォントの「A」の大きさは表1のとおりである。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しきい値 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、式(5)で定義される類似度がしきい値 T_2 以上の領域を文字の切出し箇所とした。

提案手法では、方向成分をもたない「エッジなし画素」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なし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、探索領域がエッジなし画素を7割以上含む場合は、その箇所での特徴検出は行

わないこととし、特徴なしとして扱う。

以下の実験のパラメータについて、投票をする際の半径は、特に断らない限り、 $r = 15$ を用いた。また、パラメータ $T_1 = 0.75$ とした。 T_2 は大きくすると変形を許容できなくなり、小さくすると極端な変形あるいは他の文字にもマッチしてしまうので、適切な値を選ぶ必要がある。以下の実験では、様々な値で計算し最も性能がよかった値を示している。いかに適切な値に決めるかは今後の課題である。

5.1 予備実験

3.3で示したように、提案手法の計算時間は投票ベクトル数(参照画像から得られた代表点の数と同じ)に大きく依存する。そのため3.1で述べた q を大きくし、代表点数を減らすことで計算時間の短縮が見込める。その反面、 q を大きくしすぎると、特徴検出位置に誤差を生じ、切出し精度の低下を招くことが考えられる。そのため予備実験として、絞込みをする範囲を変化させたときの計算時間を調べた。

変化させる範囲は $q = 0$ (絞り込まない) から $q = 10$ までとした。入力画像として Arial Black (図9(a))を用い、しきい値は $T_2 = 0.85$ を用いた。英数字 62 文字を通して認識したときの計算時間を図7に示す。横軸が q の値、縦軸が計算時間(秒)を表す。結果を見ると、範囲を拡大させると絞込みをしないときと比較して、1/3程度まで短縮できているが、 $q = 5$ 以上に範囲を大きくしても、計算時間の短縮があまり見込めなくなることが分かった。そこで、本論文では $q = 5$ を使用することにする。

5.2 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 150×150 ピクセルの画像の中心に配置

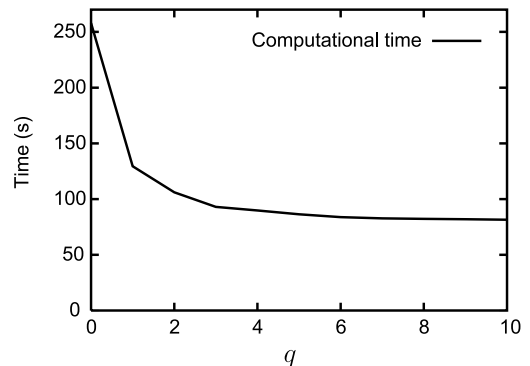


図7 絞込みをする範囲 q を変化させたときの計算時間
Fig. 7 The computational times by change of q .

表1 「A」の文字画像の大きさ
Table 1 Sizes of character images of "A"s.

フォント	文字画像の大きさ (ピクセル)
Arial	72 × 78
Arial Black	85 × 78
Franklin	64 × 73
丸ゴシック	76 × 78



図9 評価基準
Fig.9 Evaluation criteria.

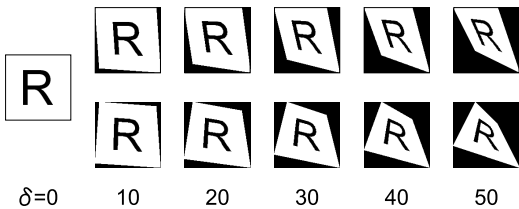


図8 射影変換を受けた画像
Fig.8 Distorted images.

表2 射影変換に対する頑健性
Table 2 Robustness against projective distortion.

δ	認識率 (%)	δ	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

し、4 隅の x, y 座標をそれぞれ 0 または δ ($\delta = 5, 10, \dots, 50$) だけ変位させた射影変換を施し、各 δ について $2^{4 \times 2}$ 個のテストパターンを生成した (図 8 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 2 に示す。パラメータは $T_2 = 0.7, r = 4\delta$ 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、 $\delta = 50$ のときでも 99% 以上の認識率を示している。このことから、 r を適切に設定する限りは提案手法は射影変換に対して非常に頑健であるといえる。

5.3 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う。入力画像には各フォントの英数字 62 文字が図 9 のように並んでいる画像を用いた。それぞれの画像には、英

表3 フォントの異なる文字に対する認識結果
Table 3 Recognition results for different fonts.

(a) The numbers of occurrences and their ratios

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%)

(b) Recall & Precision

フォント	Recall	Precision
Arial Black	100%	9.27%
Franklin	100%	6.16%
Maru Gothic	100%	5.71%

数字 62 文字が含まれている。パラメータは $T_2 = 0.8$ とした。

本論文では、実験結果を Correct, Match, Miss の 3 種類に分類して評価した。Correct は図 9 (a) のように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、それらの文字をすべて切り出し、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 は図 9 (b) のように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、切り出した箇所の中に探索したい文字が含まれていることから、切出し手法としての役割は果たしていると考えられる。Miss は図 9 (c) のように探索している文字を切り出すことができなかった場合である。

認識結果を表 3 にまとめた。まず表 3 (a) は Correct, Match, Miss になった文字数と割合を表している。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できた割合 (Correct と Match の和) はすべてのフォントに対して 100% であり、フォントが異なるにもかかわらず、高い精度で切

出しに成功している．すなわち，提案手法はフォントが異なる場合でも一定の切出し性能をもつことが確認できた．

ちなみに，Match に分類された入力画像では，フォントによって違いはあるが，10～20 箇所の切出し箇所となっており，そのうち正解は 1 箇所であるから，文字種に関する適合率は 10% 以下であった．その詳細を表 3 (b) に示す．しかし，Miss に分類されたものがないことから再現率は 100% である．提案手法では，角・曲線の特徴の存在だけをチェックしているため個々の文字種を区別するのは困難な場合が多く，図 9 (b) に示されるように参照画像以外の文字も多く切り出される．したがって適合率は低い値となる．この適合率が低いことは参照文字を正確に切り出すには更に文字認識が必須であることを示している．切出し箇所が 10～20 箇所であったことは，1000 × 1000 のサイズの画像からわずか 10～20 箇所の位置を特定したことであり，切出し手法としては十分な役割を果たしているといえる．

5.4 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 $T_2 = 0.87$ を用い，道路標識には $T_2 = 0.93$ を用いた．一般化ハフ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 [13]．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少な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 10 と図 11 に示す．時計の文字盤に対しては，参照画像に「2」を用いたときに，対象となる文字をすべて切り出し，他の文字を一切切り出さなかったため Correct とし，他のすべての数字で 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」を用いたときに Match になった．「2」と「4」以外の数字に対しても，切出しが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．このような誤検出の原因は，提案手法が「特定の位置に特定の特徴があ



図 10 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

Fig. 10 A recognition result against a clock-face.



図 11 道路標識に対する認識結果：参照画像に「4」を使用

Fig. 11 A recognition result against a road sign.

る」ということを重点的に検出し，「特定の位置に特定の特徴がない」ということを軽視したものになっているためである．したがって，参照文字の特徴を含んでいればそれ以外のものが過剰にあってもマッチすることになる．このことは，文字切出しにもれがないという頑健さを与える理由でもある．

本節の実験で，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出し能力を有していることが確認できたが，一方で参照画像と字形が異なる文字画像に対する限界も示している．字形の変動に対しては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．提案手法がこの問題点の時間増加を少し軽減していることを 5.6 で議論する．

5.5 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た．実験で使用した参照画像は Arial の「S」と「e」である．パラメータは，「S」の場合は $T_2 = 0.8$ を，「e」の場合は $T_2 = 0.89$ を用いた．5.4 と同様に，

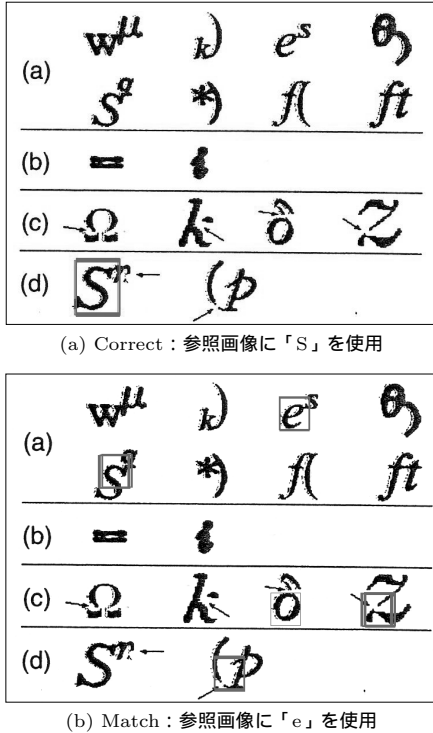


図 12 接触文字に対する認識結果

Fig. 12 Recognition results against touching character.

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を図 12 (a) に，参照画像に「e」を用いたときの結果を図 12 (b) に示す．参照画像に「S」を用いた場合は Correct で，参照画像に「e」を用いた場合は Match であった．実験の結果より，従来法では切出しが困難であった接触文字に対しても提案手法は切出しを行うことができる．

図 12 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．

5.6 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，マッチングする領域内の「エッジなし画素」ではない画素を 16 方向に分割して作成したエッジヒストグラムを特徴量とし，2.4 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

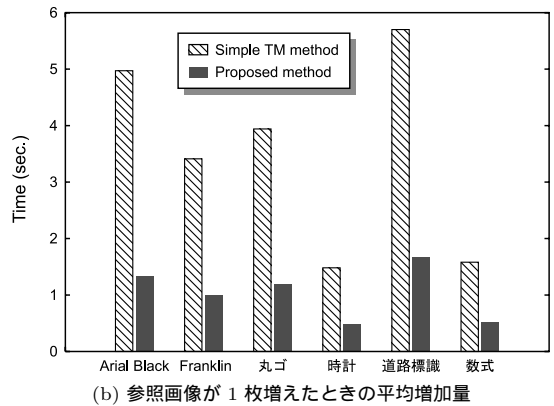
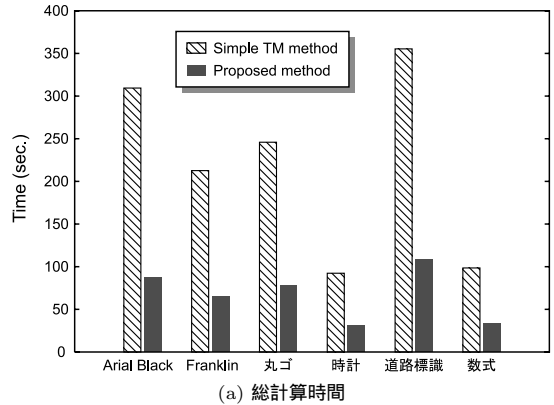


図 13 用いた画像の大きさや計算時間

Fig. 13 The sizes of used input images and computation times.

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 2.4 GHz である PC を用いた．

5.3 から 5.5 で用いた入力画像に対する英数字 62 文字の計算時間を図 13 に示す．図中の TM はテンプレートマッチングを表す．「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，「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．また，参照画像が 1 枚増えたときの平均増加量とは，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. む す び

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もち、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、更に文字認識の大分類も同時に行う手法 (FPV 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少ない。

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、更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、しきい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、更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、並びに若手研究 (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, March 2005.
- [2] 松尾賢一, 上田勝彦, 梅田三千雄, “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, June 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, Sept. 1998.
- [5] 森 健一 (編), パターン認識, 電子情報通信学会, 1988.
- [6] M. Swain and D. Ballard, “Color indexing,” Int. J. Comput. Vis., vol.7, pp.11-32, 1991.
- [7] D. Ballard, “Generalizing the Hough transform to detect arbitrary shapes,” Pattern Recognit., vol.13,

no.2, pp.111-122, 1981.

- [8] D.H. Ballard and C.M. Brown (著), 福村晃夫ほか (訳), コンピュータ・ビジョン, 日本コンピュータ協会, 東京, 1987.
- [9] 松山隆司, 久野義徳, 井宮 淳 (編), コンピュータビジョン: 技術評論と将来展望, 新技術コミュニケーションズ, 東京, 1998.
- [10] 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, April 1990.
- [11] 山田博三, 山本和彦, 細川勝美, 中村匡伸, “MAP (multi-angled parallel) 演算法と MAP 整合法による地形図の特徴抽出と認識” 信学論 (D-II), vol.J74-D-II, no.6, pp.718-726, June 1991.
- [12] 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 信学総大, D-12-72, 2002.
- [13] 木村彰男, 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.

(平成 18 年 5 月 2 日受付, 8 月 30 日再受付)



根岸 和也

平 16 東北大・工・通信卒。平 18 同大大学院修士課程了。同年東芝入社。在学中はパターン認識の研究に従事。



岩村 雅一 (正員)

平 10 東北大・工・通信卒。平 15 同大大学院博士課程了。同年同大学院工学研究科助手。平 16 阪府大大学院工学研究科助手。現在に至る。博士 (工学)。パターン認識, コンピュータビジョンに関する研究に従事。IEEE, 情報処理学会, Pattern Recognition Society 各会員。



大町真一郎 (正員)

昭 63 東北大・工・情報卒。平 5 同大大学院博士課程了。同年同大情報処理教育センター助手。平 8 同大工学部助手。平 11 同大大学院工学研究科助教授。現在に至る。博士 (工学)。その間, 平 12~13 米国ブラウン大学客員助教授。パターン認識, コンピュータビジョン, 並列処理, 文字認識システムの開発などの研究に従事。IEEE, 情報処理学会, 人工知能学会, Pattern Recognition Society 各会員。



阿曾 弘具 (正員:フェロー)

昭 43 東北大・工・電気卒。昭 49 同大大学院博士課程了。昭 48 東北大・工・助手，昭 54 名大・工・講師。昭 57 同助教授，昭 61 東北大・工・助教授を経て，平 3 同教授。現在，同大大学院工学研究科教授。工博。その間，学習オートマトン，セル構造オートマトン，並行処理理論，シストリックアルゴリズム設計論，文字認識，音声認識，ニューラルネットワークなどの研究に従事。平 3 年度本会業績賞受賞。IEEE，ACM，EATCS，情報処理学会，人工知能学会，LA 各会員。