

## 文字画像の修復による低解像度印刷漢字の認識

胡 彦\*・松山 繁史\*\*・松田 和生\*  
中村 千秋\*\*\*・宮原 末治\*Low-Resolution Kanji Printed Character Recognition  
by Restoration of Character Image

by

Yan HU\*, Shigefumi MATSUYAMA\*\*, Kazuo MATSUDA\*

Chiaki NAKAMURA\*\*\*, Sueharu MIYAHARA\*

A restoration method is presented for low-resolution printed kanji character recognition. First, character images are converted into binary character patterns after expanding and blurring. Next, the binary character patterns are modified by the system using ridge point and ravine one. The feature vectors of the modified character patterns which include the information of spatial structure are extracted by phrase feature distribution method and they are fed in an classifier. It is verified by experiments using the database ETL2 that proposed method improved recognition accuracy from 89.1% to 99.4%.

## 1. はじめに

デジタルカメラやデジタルビデオなどの普及に伴い、カメラ画像からの文字認識の要求が高まっている<sup>(1)(2)</sup>。しかしながら、現在実用化されているOCR (Optical Character Reader) は、スキャナから取り込んだ高解像度の文字画像を讀取対象としているため、既存の認識技術をそのまま適用することは困難と思われる。その主な原因は、より多くの情報を読み取ろうとして、カメラからの入力画像の視野を広く取ろうとするため、文字部分が低解像度の画像になることに起因するものと考えられる。低解像度の画像は、文字パターンの文字縁に段差が生じたり、細かな部分がつぶれたり、かすれたりするため、これらの画像をそのまま使って認識させると、高い認識精度を実現できないことが多い。

これらの問題に対して、これまでに文字パターンをわずかに移動させ、それらの差分をとることで文字線縁の傾きを精度よく検出する移動差分法<sup>(3)</sup>や、線形フィ

ルタを用いた画素間の補間と凸判定により品質のよい文字パターンを得る2値化法<sup>(4)</sup>などが提案されてきた。これらの手法は低解像度の画像を修復し、読みやすい文字パターンを得る方法として一定の効果を上げているが、汎用性や認識精度の点では十分であると言えない。

本稿では、デジタルカメラやデジタルビデオ画像からの文字認識を目的とし、低解像度文字画像の修復を多段に行うことにより文字画像の画質を改善し、それを用いて高精度に文字認識する方法について提案する。提案手法では、低解像度の入力文字画像に対し、文字パターンの拡大とぼかしの処理を行い、さらに2値化の際に尾根点・谷点の検出により文字パターンに補正を施し<sup>(5)</sup>、修復した文字パターンに対して位相特徴分布法<sup>(6)</sup>により文字線縁の局所的な傾斜特徴に位置情報を付与して特徴ベクトルを形成し、識別を行う。

実験データとして、産総研(旧電総研)が公開している多値の印刷漢字データベースETL2<sup>(7)</sup>を擬似的

平成14年4月19日受理

\*大学院生産科学研究科 (Graduate School of Science and Technology)

\*\*富士ソフトABC(株) (Fujisoft ABC Inc.)

\*\*\* 教育学部 (Faculty of Education)

に解像度を落として $16 \times 16$ 画素の低解像度文字データとして使用した。以下、2章では修復手法、3章では特徴抽出と識別、4章では評価実験と考察について報告する。

## 2. 修復手法

カメラなどで入力された画像は一般に濃淡画像であり、文字認識ではそれを2値化して用いる。2値化処理は、各画素の濃度値が予め定められたしきい値や、自動的に求められたしきい値との大小関係を比較して行われる。しかし、これらの方法を低解像度の文字画像に対して適用すると、つぶれやかすれ、ジャギーなどが発生する。そこで本稿では、2値化処理の前に文字の拡大とぼかし処理を行い、2値化の際に尾根点・谷点の検出と補正によって、2値化される文字パターンに修復を行う方法を提案する。本手法の処理の流れを図1に示し、画質改善のための修復の処理を破線の枠内に表す。

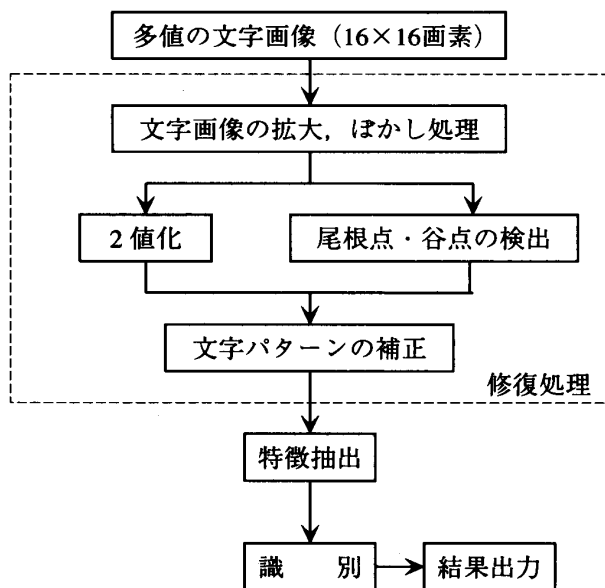


図1 文字認識処理における修復処理の手順

### 2.1 文字の拡大とぼかし処理による補完

ジャギーは、文字パターンを少ない画素で表現しようとしたために生じたものである。そこで文字パターンの拡大と、ぼかし処理とを同時に行い、見掛け上の解像度を上げることで輪郭部の凹凸を除去し、原画像の文字縁を滑らかにする。この処理により擬似的に高解像度文字画像を得ることができる。

処理では、まず入力された濃淡画像の各画素を $n \times n$ 画素に拡大し、次に $m \times m$ のフィルタでぼかし処理を行う。図2に例として、3倍に拡大して( $n=3$ )、

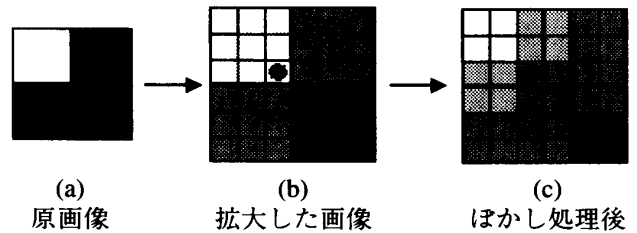


図2 文字パターンの拡大とぼかし処理

$3 \times 3$ のフィルタ( $m=3$ )をかけた場合を示す。すなわち $2 \times 2$ 画素の原画像(a)は、拡大の処理により $6 \times 6$ 画素の画像(b)になる。次に拡大した画像の全ての画素に $3 \times 3$ のマスクをかけ、そのマスク内で濃度値の平均をとって中央画素の濃度値を求める。処理結果を図2の(c)に示す。

以上の処理で原画像の濃度値を保存し、かつジャギーの除去と同時に原画像より大きさが縦横3倍になるぼかしパターンが生成される。入力された原画像が $16 \times 16$ 画素ならば、3倍に拡大してぼかしをかけた文字パターンは $48 \times 48$ 画素の大きさになる。原画像を大津の手法を用いて2値化した場合の一例を図3(a)に、文字パターンを拡大し( $n=3$ )、ぼかし処理( $m=3$ )を施して2値化した文字パターンを図3(b)に示す。

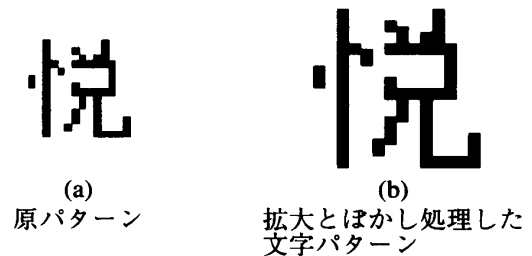


図3 拡大してぼかし処理を施した後2値化した文字パターン

### 2.2 尾根点・谷点による補正

かすれは文字線の濃度と同じ濃度になるはずの濃度が本来の値よりも低くなる事により生じ、つぶれは文字線の間隙部分でかすれと逆の現象が生じたために起きるものと考えることができる。このように文字線につぶれやかすれが生じた場合でも文字線部分では周囲と比べて濃度が高くなり、文字線の間隙部分では周囲より低くなることから、文字画像の濃度値の凹凸を調べることにより、文字線や文字線間隔部分をその周辺部分と区別することができる。そこで、本稿では大津の方法で求めたしきい値で2値化を行うと同時に、各画素の8近傍の濃度を調べ、濃度値の高い方を尾根点と、低い方を谷点として検出し、その検出結果に基づいて2値化の結果に対し補正を行う。

### 2.2.1 尾根点・谷点の検出

尾根点、谷点の抽出には図4に示すような $3 \times 3$ 画素のマスクを用いる。中心点 $X_0$ が尾根点か谷点かの判定は、 $X_0$ とその8近傍、すなわち $X_1 \sim X_8$ の濃淡値の分布の状態によって決定する。

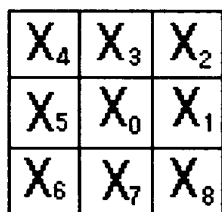


図4 8近傍

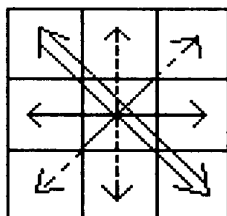


図5 4方向

図5のように、縦、横、左斜め、右斜めの4方向を考え、各々の方向で中央の $X_0$ の濃淡値が、両側の2点の濃淡値よりも大きい時、その方向で中央極大であるとする。また、 $X_0$ の濃淡値が両側の2点の濃淡値より小さい時、その方向で中央極小であるとする。

以上のような中央極大となるような方向が2つ以上ある点を尾根点、中央極小となる方向が2つ以上ある点を谷点とする。ただし、中央極大、中央極小となる方向がそれぞれ2つならば尾根点でも谷点でもないものとする。

さらに、文字背景の部分にノイズによる尾根点が生じることや、文字線上に、ノイズによる谷点が生じることがある。これらを防ぐため、明らかに文字背景の濃度値の時は、尾根点とはしないとする尾根点しきい値OTHを設ける。また明らかに文字線部分の濃度値のときは谷点としないとする谷点しきい値TTHを設け、このしきい値による処理を行い、ノイズの混入を防いでいる。

### 2.2.2 文字パターンの補正

大津の手法で求めたしきい値を用いて2値化した文字パターンに対し、尾根点が白になっているものを尾根点エラー、谷点が黒になっているものを谷点エラーとする。このエラーを検出して反転させ、尾根点を黒に、谷点を白に強制的に変更する。その一例を図6に示す。



図6 原画像と修復画像の比較

## 3. 特徴抽出と識別

本手法の評価のために、低解像度文字パターンを識別するための特徴としては、文献<sup>9)</sup>で文字線の傾き特徴が有効な特徴であることが示されているので、本稿でも同様な文字線縁の傾き特徴をマスク処理で容易に抽出できる方法として、印刷漢字の認識に実績のある位相特徴分布法<sup>10)</sup>を使用することにした。

### 3.1 特徴抽出

認識実験に使用する特徴としては、文字線縁の傾きに着目した傾斜特徴と黒点量を用いる。

#### 3.1.1 傾斜特徴の抽出

修復処理が施された文字パターンに対し、文字線の縁の部分に $3 \times 3$ のマスクを当てはめ、周囲の黒画素の状態を調べ、文字線縁の局所的な傾斜を示す特徴(記号として表したH, V, L, R点の中の一つの特徴)を付与する。この処理により、文字線縁のすべての黒点には水平(H)、垂直(V)、左斜(L)、右斜(R)の4種の特徴のいずれかが付与され傾斜特徴が形成される。識別ではこの4つの特徴に加え、黒点量(B)も求め、5つの特徴を用いる。

#### 3.1.2 スムージング

LまたはRの傾斜符号を有する黒点において、斜め前後にHとVの傾斜符号を有する黒点が存在する場合、その黒点にHとVの傾斜符号を付加し、一つの点に3種類の特徴を持たせる。この処理によって文字線の交差によって生じる不安定な特徴の影響が低減される。

#### 3.1.3 ブロック分割

文字線部に抽出された特徴に位置情報を付加するために、文字パターンの外接枠を基準にして、文字パターンを $r \times s$ の領域に分割して扱う。この分割により、各々の傾斜特徴の存在する位置が、文字パターン上の配置として表現されるようになり、特徴相互の関連付けとその分布の状態を知ることができる。ここで分割数を多くすると文字パターンの位置情報がより詳細に表現できるが、文字パターンが変形したり、劣化した場合、あるいは文字パターンの周辺部が欠けたりした場合にその影響を受けやすくなる。図7に示すように $r = s = 8$ の場合を例とすると、特徴抽出によって320次元( $8 \times 8 \times 5$ )の特徴ベクトルが得られる。得られた特徴は正規化のために、傾斜特徴量(H, V, L, R)は分割領域の周囲長で、黒点量(B)は分割領域の面積で正規化する。

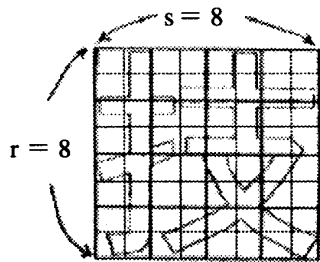


図7 ブロック分割

### 3.2 識別

実験では、ユークリッド距離を用い、距離値の近いものから順に答えとして出力した。

## 4. 評価実験と考察

16×16画素で表れられた低解像度の印刷文字と本手法によって修復された印刷漢字とに対し、認識実験の正解率によって、修復の改善効果を調べる方法を採用した。

### 4.1 実験データ

実験データとして、産総研印刷漢字データベース ETL 2 の中から明朝体の印刷漢字2136カテゴリーを使用した。評価実験では表1に示すように、カテゴリーあたり10パターンであり、前半6パターンを学習データとし、後半4パターンをテストデータとした。

低解像度画像の作成は、元の解像度が60×60画素の ETL 2 の文字画像に対し、中央の48×48画素を有効な部分として取り出し、この部分に3×3画素を1画素に変換する処理で16×16画素に落とした。以後この画像を原画像と呼ぶ。

表1の原画像をもとにして、評価実験のために表2に示す6種の画像を用意した。データ1は原画像を大津の手法で2値化した文字パターンである。データ2は原画像を2倍に拡大し、ぼかし処理を施した文字パターンである。データ3は原画像を3倍に拡大し、ぼかし処理を施した文字パターンである。データ4はデータ1に尾根点・谷点による補正を施した文字パターンである。データ5はデータ2に尾根点・谷点による補

表1 実験データ仕様

項目	内容
カテゴリー数	2136
パターン数/カテゴリー	6 (学習), 4 (テスト)
濃度	256階調
原画像の画素数 (大きさ)	16×16画素 (60×60画素の ETL 2 文字画像を縮小)

正を施した文字パターンである。データ6はデータ3に尾根点・谷点による補正を施した文字パターンである。表2の右端に文字パターンの大きさを画素数で表している。また、それと同じ箇所に認識結果と対応をとるために、学習データをA・Bとし、テストデータをC・Dと区別して、データ番号と実験状態との関係を示している。図8に16×16画素の文字画像の例を示す。比較のために、図8の16×16画素の文字画像を大津の手法で2値化した文字パターンの例を図9に示す。また、本手法で修復した48×48画素の文字パターンを縮小した例を図10に示す。

表2 実験データの種類の種類

データ番号	修復の内容	大きさ(画素) 学習/テスト
1	原画像に大津の手法で2値化した文字パターン	16×16 1A,1B/1C,1D
2	原画像を2倍に拡大し、ぼかし処理を施した文字パターン	32×32 2A/2C
3	原画像を3倍に拡大し、ぼかし処理を施した文字パターン	48×48 3A/3C
4	データ1に尾根点・谷点による補正を施した文字パターン	16×16 4B/4D
5	データ2に尾根点・谷点による補正を施した文字パターン	32×32 5B/5D
6	データ3に尾根点・谷点による補正を施した文字パターン	48×48 6B/6D

峽嵐嵩嶋循忌忍怒怠慈  
愚憂愁忙悦悔恨悟悼情

図8 多値の文字画像 (16×16画素)

峽嵐嵩嶋循忌忍怒怠慈  
愚憂愁忙悦悔恨悟悼情

データ1

図9 大津の手法を用いた文字パターン (16×16画素)

峽嵐嵩嶋循忌忍怒怠慈  
愚憂愁忙悦悔恨悟悼情

データ6

図10 本手法を用いた文字パターン (48×48画素)

### 4.2 実験結果

評価実験は表2に示した全6種類の実験データに対し、3章で述べた特徴抽出と識別処理を行った。特徴抽出では文字パターンに対し、3.1項に示したマスクを用いてすべての黒画素に傾斜符号を付与した後、文字パターンの外接枠を基準として8×8にブロック分割して特徴を計数した。その結果、識別に用いる特徴ベクトルは、320(8×8×5)次元となった。

識別結果を第1位正解率と候補5位までの累積正解率とに分けて、図11から図14に示す。縦軸は正解率を、横軸は文字パターンの修復の状態を示すデータ番号を表している。第1位の正解率を実線で、候補5位までの累積正解率を破線で示す。図11と図12は学習データに対する認識結果であり、図13と図14はテスト

データに対する認識結果を示している。両者を比較すると、正解率はテストデータの方が全体的に低いことが分かる。

テストデータの実験結果では、図13に示すように、本手法を用いることにより第1位の正解率が89.1%から99.1%へ、5位候補までの累積正解率が95.1%から99.8%へ改善している。また、図14に示すように第1位の正解率が89.1%から99.4%へ約10%、5位候補までの累積正解率が95.1%から99.9%へ、約5%向上している。

### 4.3 考察

本手法を低解像度の文字パターンに適用した場合、図9の文字パターンと図10の比較から視覚的にも画質

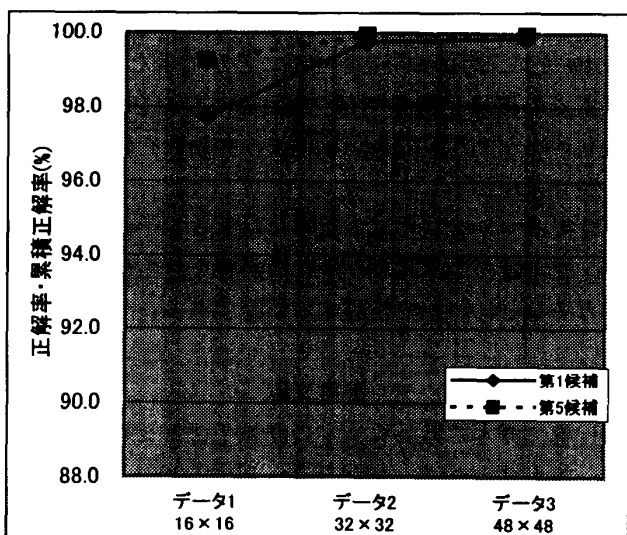


図11 学習データの認識結果 A (拡大, ほかし処理)

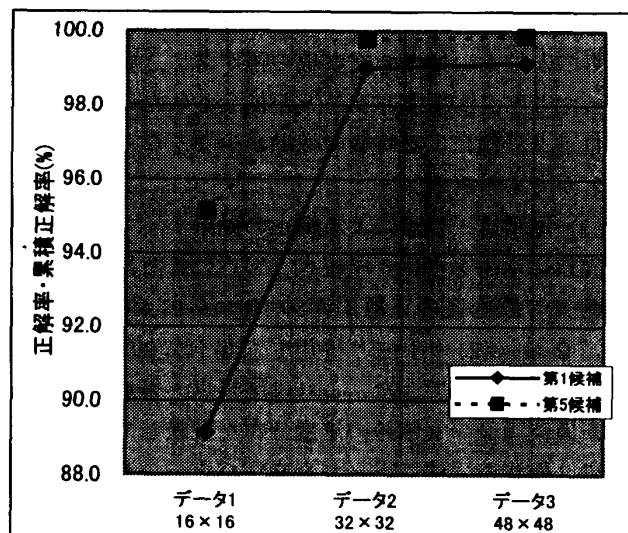


図13 テストデータの認識結果 C (拡大, ほかし処理)

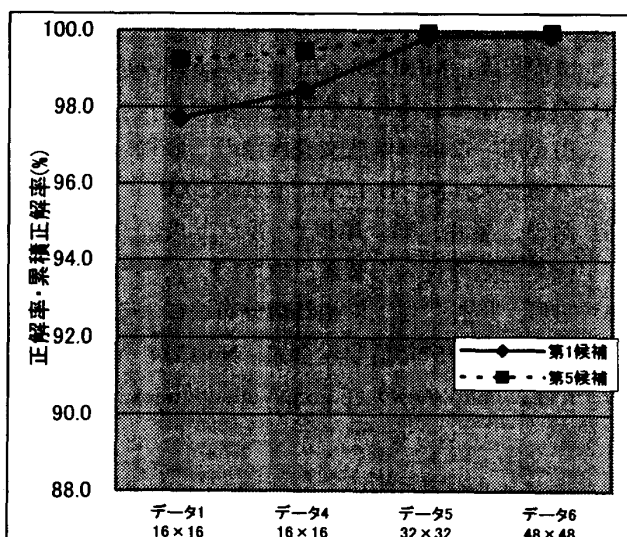


図12 学習データの認識結果 B (尾根点・谷点による補正)

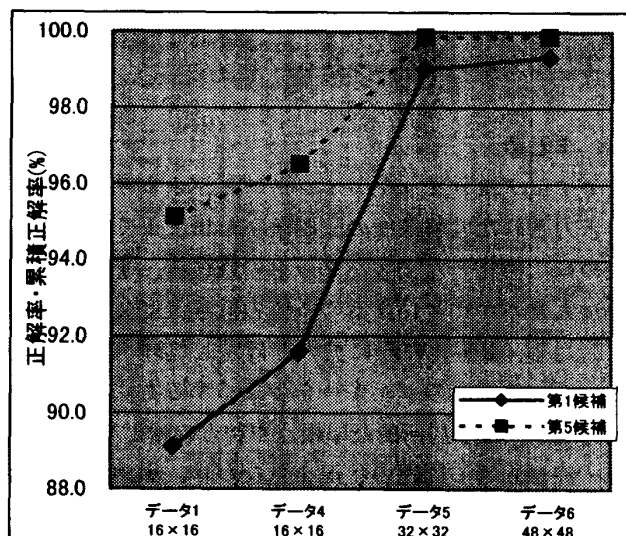


図14 テストデータの認識結果 D (尾根点・谷点による補正)

の改善効果が確認できる。定量的な評価のために行った文字認識実験の結果から、次のようなことが分かった。

#### (1) 認識精度の比較：

16×16画素の学習データとテストデータとの比較では、図11と図13、および図12と図14に見られるように、正解率に大きな差が生じた。これは学習データ数が少なく、入力データのバラツキを十分吸収することができなかったためと考えられる。

#### (2) 拡大とぼかし処理の効果：

図13に示すように、拡大とぼかし処理で第1位正解率は、89.1% (1C) から99.1% (3C) へ大幅に改善された。この原因は、①これまで不安定であった斜め成分の特徴 (LおよびRの特徴) が安定化されたこと、②16×16画素の文字パターンでは、文字線の幅が1画素の場合が多く生じ、文字線縁の特徴が不安定であったものが拡大の処理で文字線の幅が常に2画素以上となったこと、③文字パターンが拡大されたことによりブロック分割による特徴の変動が少なくなったこと、などが考えられる。

#### (3) 尾根点・谷点による補正の効果：

図14に示すように、尾根点・谷点による補正で16×16画素の原画像の正解率が89.1%から91.6%に向上した。なお、(2)の手法と併用した場合に99.4%まで向上させることができた。これは尾根点・谷点による補正が濃淡画像の文字線の濃度分布を反映しており、大津の2値化の手法で得られた文字パターンをさらに改善してくれたものと考えられる。

識別実験の結果に、候補5位までに出現せずに誤認識になった文字パターンの例を図15に示す。その主な原因は、つぶれやかすれが多く存在し、特徴が大きく変化したものかと考えられる。

## 5. まとめ

デジタルカメラやデジタルビデオカメラなどから入力された低解像度の印刷漢字を精度よく認識するために、劣化した文字パターンの修復法、すなわち

- (1) 大きさが16×16画素の多値の低解像度文字画像に対し、2値化の際に拡大とぼかし処理を施して文字パターンのジャギーを少なくする方法、
- (2) 文字画像の局所的な尾根点や谷点の検出を行って、注目画素の周囲の状況を見ながら、黒画素、白画素を補正する文字パターンの修復方法、

について検討した。

修復方法の評価では、産総研データETL2に対し、

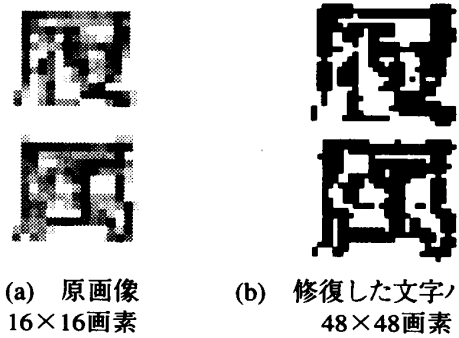


図15 誤認識の文字パターンの例

解像度を60×60画素から16×16画素に落として、擬似的な低解像度の印刷漢字を作成して使用した。認識実験では位相特徴分布法を用い、正解率による評価を行った。その結果、本稿で提案した文字パターンの修復法によって、テストデータで正解率が89.1%のものを99.4%まで、約10%向上させることができた。また第5位までの累積正解率では、95.1%のものを99.9%まで向上させることができ、本手法の有効性を確認することができた。

今後はマルチフォント文字に対する本手法の適用効果、文字線のおぼれやかすれの影響をさらに少なくする方法などについて検討する予定である。

## 参考文献

- (1) 森, 澤木: 低品質文字の認識手法とその応用に関するサーベイ, 信学技報, PRMU2001-275, p.25-40(2002).
- (2) N. Nakajima, N. Tanaka, K. Yamada: Document Reconstruction and Recognition from an Image Sequence, ICPR, p.922-925(1998).
- (3) 小佐井, 山本, 他: 低解像度における文字認識手法の提案, MIRU98-7, p.257-262(1998).
- (4) 鈴木, 窪田: 補間と凸判定に基づくストローク抽出を用いた低解像度文書画像の二値化, 2000年信学総合大, No. D-12-34, p.204(2000).
- (5) 岡田, 坂井, 他: 尾根点・谷点方式による文書画像の前処理付入力装置, EC82-5, p.45-54(1982).
- (6) 宮原, 川谷, 他: 位相特徴分布法による印刷漢字認識, 昭和57年信学総合大, No.1334, p.5(1982).
- (7) ETL2: <http://www.etl.go.jp/~etlcn/index.htm>