

様々な方向に傾いた印刷数字の認識アルゴリズム

濱本 高志・川島 広誉*・大倉 充*・塩野 充*

岡山理科大学大学院工学研究科修士課程電子工学専攻

*岡山理科大学工学部情報工学科

(1996年10月7日 受理)

1. ま え が き

近年、自動車のナンバープレートの抽出・認識¹⁾等の情景画像中の文字列を対象とした研究が盛んになりつつある。情景画像から文字列を抽出した際には、多くの場合、抽出された文字列が傾いていることが考えられる。そのような文字列を認識するためには、一般的に前処理として文字列の回転補正が必要となる。しかし、視点と対象物体が位置する角度が定まらなければ回転補正を行うことは不可能である。この点を考慮し、本研究では回転補正を行わずに文字認識を行うことを試みた。現実世界では様々な文字が存在するが、本研究では単一フォントの印刷数字のみを認識対象とし、コンピュータ上で基本となる数字画像を回転させ実験用サンプル画像を作成し認識実験を行った。

2. 実験で使 用 した 画 像 デ ー タ

実験で使 用 した フォントはサイズが80ポイントの中ゴシック BBB で白黒2値である(図1参照)。このフォントでは数字の面に対して垂直な軸で回転処理を施すと、数字の6は180度回転させた9と同じ形状をなすことがわかる。このため本研究では、数字の9を省き、カテゴリー数を数字の0から8までの9カテゴリーとした。

実験用サンプル画像は、それらの数字フォントを100×100画素の白領域のほぼ中心に配置し、図2に示す回転軸を定め、回転処理を行った後に X-Y 平面に正投影法²⁾で平行投影して作成した。数字画像の回転は、座標軸の正の向きに対して右回りの回転を正の方向の回転とし、その時の回転角を θ と定め、Z軸→X軸→Y軸の順番で10度刻みにアフィン変換²⁾を用いることで行った。X軸、Y軸の回転処理に関しては次に示す理由により回転処理を行う角度 θ に制限を設けた。第一に奥行きに対する歪というものを考慮していないため、 $-180^\circ \leq \theta \leq 180^\circ$ の範囲で考えると 0° を境にして同じ画像が対称的に存在する理由から負の範囲を除いた。更に鏡像も考慮していないので $90^\circ < \theta$ の範囲も除いた。また、本研究で用いた画像サイズでは $\theta_x + \theta_y > 70^\circ$ の範囲では、目視した際に認識不可能な画像が多く含まれてくるため、この範囲も除いた。上述したことより、本研究では回転角度を以下に示す範囲とした。

Z軸： $0^\circ \leq \theta_z < 360^\circ$

X軸： $0^\circ \leq \theta_x \leq 70^\circ$

Y軸： $0^\circ \leq \theta_y \leq 70^\circ$

ただし、 $\theta_x + \theta_y \leq 70^\circ$

したがって、1 カテゴリーあたりのサンプル数は1296となる。

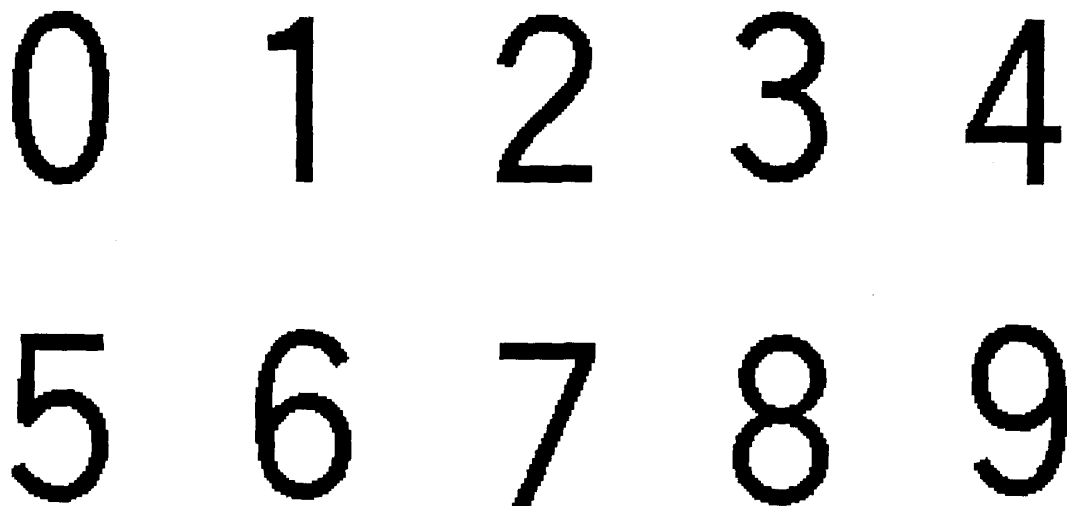


図1 中ゴシック BBB フォント

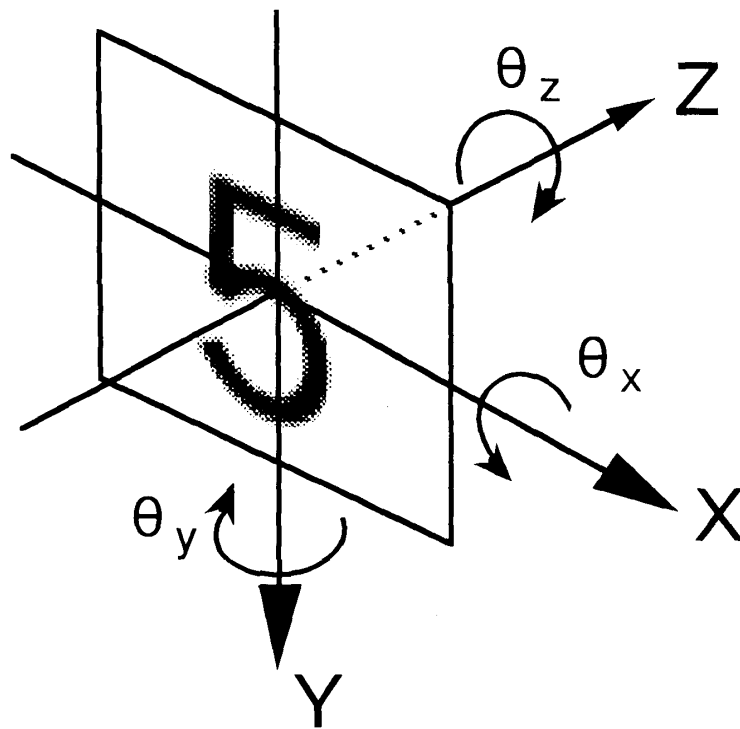


図2 回転軸

3. 認識処理

まず前処理として、数字領域に対してラベリング処理を行い、最大画素数となるラベルを持つ領域以外は背景領域とする。更に背景領域に対しても同様にラベリング処理を行い、あるしきい値以下の画素数を持つラベル領域は数字領域の欠け等のノイズと見なし数字領域とする。以後、本論文中的図に関して、基本的に背景領域は“白画素部分”を指し、数字領域は“黒画素部分”を指すこととする。

全体の認識過程を図3に示す。以下、各認識処理について詳細に述べる。

まず背景領域に対してラベリング処理を行う(図4参照)。そして、ラベル数が3の場合は8に対する認識処理へ進み、ラベル数が2の場合は0, 4, 6のグループに、ラベル数が1の場合は1, 2, 3, 5, 7のグループに分類する。また、それ以外のラベル数となった場合は棄却する。8に対する認識処理では数字のストロークで囲まれた2つの背景領域の重心を求め、各領域の重心を結んだ直線で数字を構成する画素がほぼ同数に分けることがで

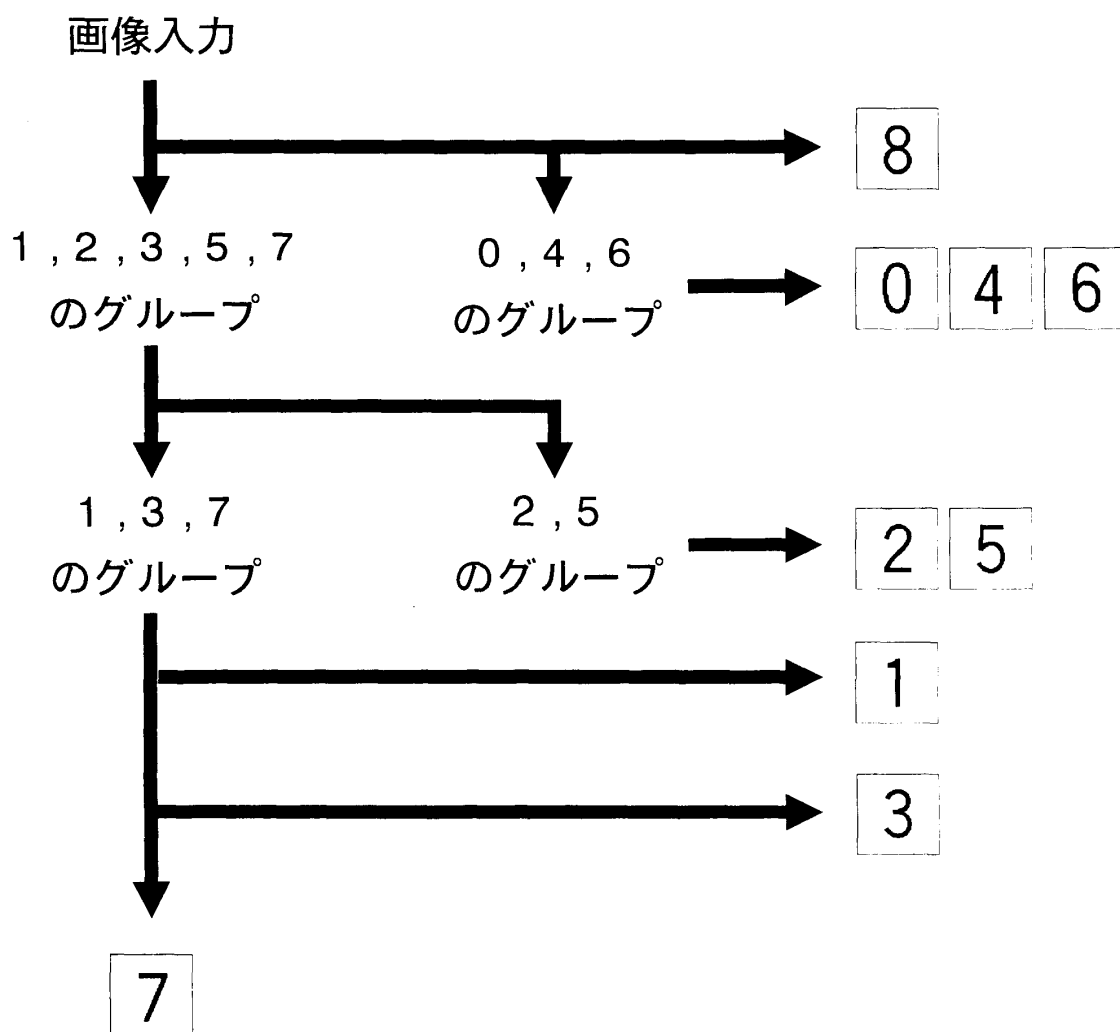


図3 認識処理過程

きる場合には8と認識し、それ以外は棄却する。

0, 4, 6のグループに属すると判定された入力画像に対しては、以下に述べる3つの認識処理を施し、どれか1つの処理で認識された場合はその数字と認識し、複数の処理で認識された場合や全く認識されなかった場合には棄却する。まず、第一の認識処理は0を認識するための認識処理である。数字のストロークで囲まれた背景領域の重心と、入力画像に対して作成した凸閉包図形³⁾(任意の2点を結んだ直線上のすべての点とその図形中に存在するような図形)から数字のストロークで囲まれた背景領域を除いた画像の重心とを比べ、ほぼ重心が同位置にあれば0と認識する(図5参照)。第二の認識処理は4を認識するための処理で、まず、数字領域の細線化処理を行う。4の場合は図6に示すように長いひげが生じる場合があるので、端点数が4以下になるまで端点部分の画素を削除する。次に任意の端点を数画素削った場合に端点でなくなる時点の座標を調べて、最も近距離に位置する2つの座標を求める。ストローク部分を削除した場合にこれら2つの座標が得られると考え、それ以外はひげと見なし削除する。最後に、交点の有無を調べ、あれば4と認識する。第三は6を認識するための処理で、数字領域の細線化処理を行い、端点数が1以下

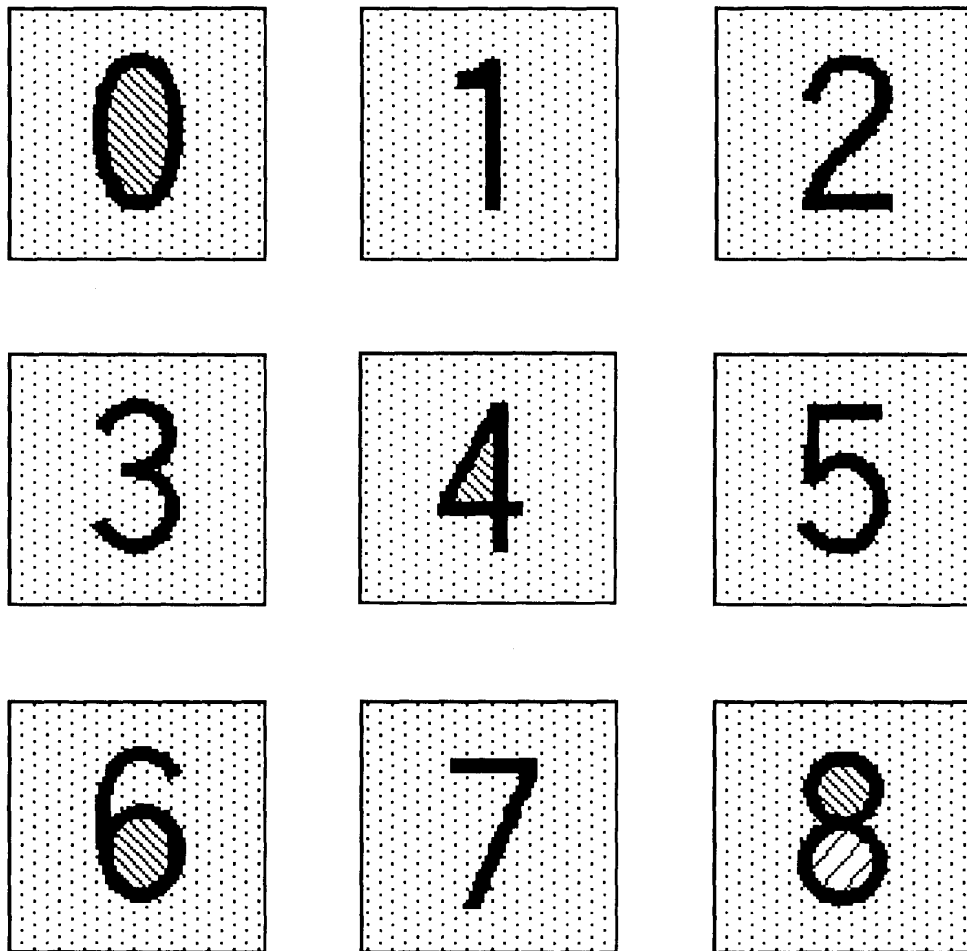
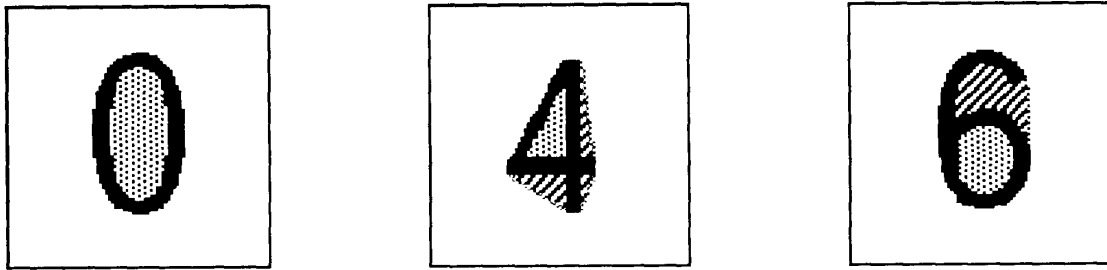


図4 入力画像の背景領域に対するラベリング処理例



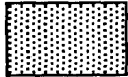


 の領域の重心と  と  の領域を合わせた領域の重心とを比較する

図5 0の認識処理

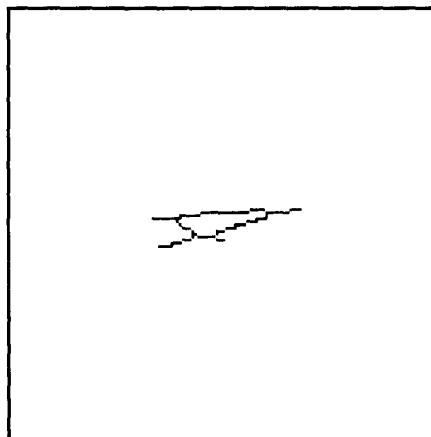


図6 回転処理した後に数字領域を細線化処理した画像例
 (カテゴリ=4: $\theta_z=50^\circ$, $\theta_x=70^\circ$, $\theta_y=0^\circ$)

になるまで端点部分の画素を削除する。次に、端点数が0になるまで端点部分の画素を削除し、このとき削除される画素数を計数し、あるしきい値以上に多く削除されていれば6と認識する。

1, 2, 3, 5, 7のグループについては、更にグループ分けの処理を施す。まず、数字領域の細線化処理を行い、端点数が2になるまで端点部分の画素を削除する。この時点で端点数が2にならない場合は棄却する。次に、各々の端点を直線で結び、背景領域に対してラベリング処理を行い、ラベル数が3となる場合は2, 5のグループに、ラベル数が2となる場合は1, 3, 7のグループに分類する。また、それ以外のラベル数となった場合は棄却する(図7参照)。

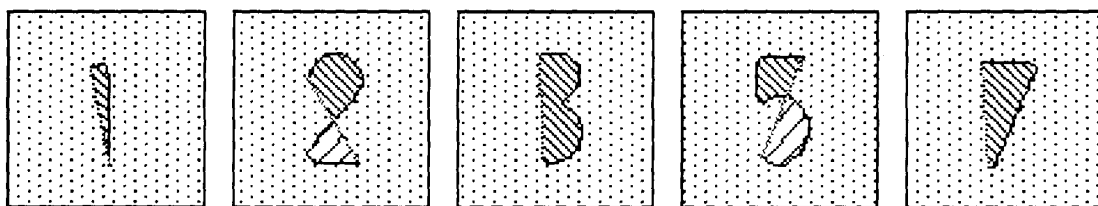


図7 2つの端点を結んだ後に背景領域をラベリング処理した画像例

2, 5のグループに属すると判定された入力画像に対しては、数字領域の細線化処理を行い、端点数が2になるまで端点部分の画素を削除する。そして、2つの端点を結んだ直線を考え、2つの端点部分の画素を数画素除去した後に再度2つの端点を結んだ直線が、正の方向（右回り）に回転した場合は2と認識し、逆に、負の方向（左回り）に回転した場合は5と認識する（図8参照）。

1, 3, 7のグループに属すると判定された入力画像に対しては、以下に述べる2つの処理を施し、どれか1つの処理で認識された場合はその数字と認識し、2つの処理で共に認識されなかった場合は7と認識する。また、2つの処理で共に認識された場合は棄却する。第一の認識処理は1を認識するための処理である。まず、数字領域に対して細線化処理を行い、端点数が2以下になるまで端点部分の画素を削除する。そして、端点と端点を結んだ直線が入力画像の数字のストロークに含まれれば1と認識する。第二の認識処理は3を認識するための処理で、第一の処理と同様に、最初に数字領域に対して細線化処理を行い、端点数が2になるまで端点部分の画素を削除する。次に、端点を1画素ずつ削りながら2つの端点を直線で結び、その画像の背景領域に対してラベリング処理を行うという手順をラベル数が1になるまで繰り返す。そして、そのつどラベル数を計数し、ラベル数が3になることがあれば3と認識する（図9参照）。

4. 実 験

実験はワークステーション NWS-5000 VI (SONY) を用いて行った。使用言語はC言語である。

認識実験の結果を表1に示す。

1カテゴリーにつき1296のサンプル画像を用い、平均で95.99%の正読率、2.54%の誤読率、1.47%の棄却率が得られた。表1からも判るように、1と7に関しては、互いによく誤読している。その理由は、1と7は回転処理を施すと回転角によっては互いに見分けのつかない画像が生じたためである（図10, 11参照）。また、4に関しては棄却数が他のカテゴリーに比べ多い。その理由の一つに、数字領域を細線化処理した場合に数字の骨格となる中心線部分が短かい画像は、ひげを削る処理の際にストロークがひげとみなされ削除され、交点が見つからないことが挙げられる。

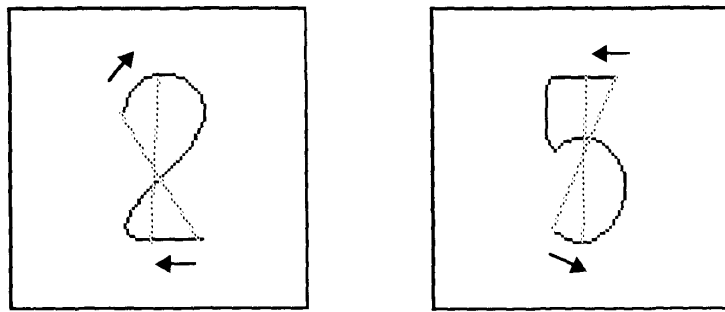


図8 2と5の認識処理

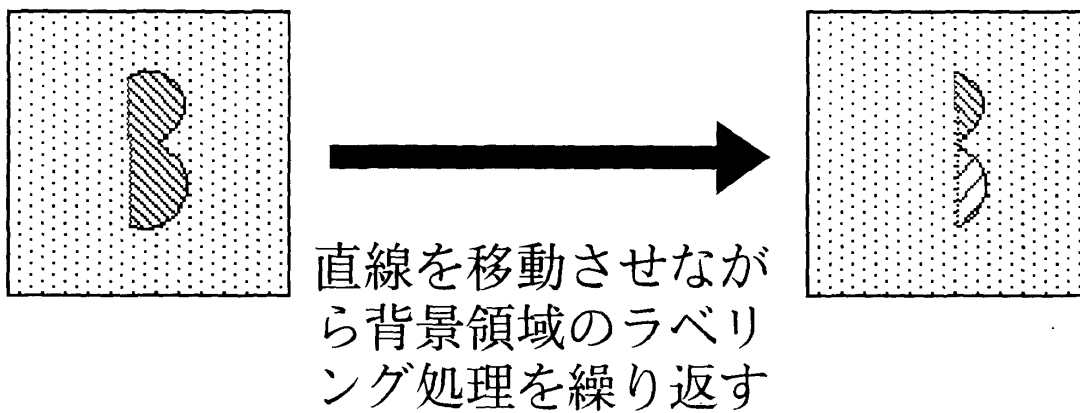


図9 3の認識処理

表1 実験結果

	0	1	2	3	4	5	6	7	8	棄却数	正読率	誤読率	棄却率
0	1296	0	0	0	0	0	0	0	0	0	100.0	0.00	0.00
1	0	1083	11	0	0	0	0	186	0	16	83.56	15.20	1.23
2	0	1	1274	0	0	0	0	4	0	17	98.30	0.39	1.31
3	0	0	0	1281	0	1	0	14	0	0	98.84	1.16	0.00
4	0	0	0	0	1172	0	5	0	0	119	90.43	0.39	9.18
5	0	0	0	2	0	1264	0	15	0	15	97.53	1.31	1.16
6	0	0	0	0	0	0	1295	0	0	1	99.92	0.00	0.08
7	0	22	2	33	0	0	0	1235	0	4	95.29	4.40	0.31
8	0	0	0	0	0	0	0	0	1296	0	100.0	0.00	0.00

(単位は正読率, 誤読率, 棄却率は「%」, その他は「個」である。)

5. むすび

本研究では3次元的に様々な傾いた印刷数字の認識に関する基礎的な検討を行った。11664の画像サンプルを用いた認識実験より平均95.99%の正読率を得た。

今後の課題として、誤読率及び棄却率を減少させるために、各数字を認識するための処理を増やすことを考えている。また、他の細明朝体やOsakaフォントなどにも対応した、マルチフォント対応の認識アルゴリズムへの改良を行う予定である。

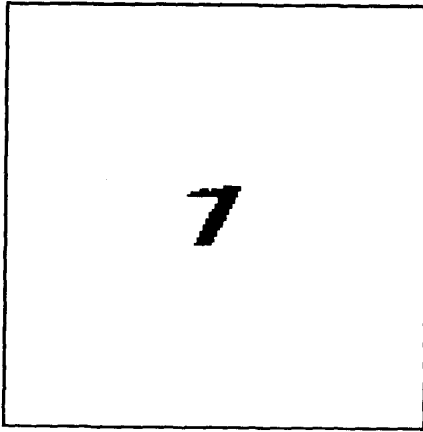


図10 7と誤読した画像 (カテゴリ-1
: $\theta_z=10^\circ$, $\theta_x=70^\circ$, $\theta_y=0^\circ$)

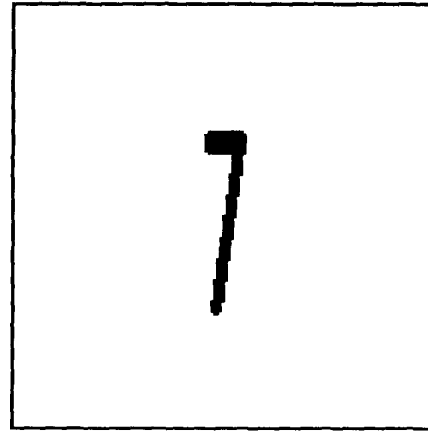


図11 1と誤読した画像 (カテゴリ-7
: $\theta_z=0^\circ$, $\theta_x=0^\circ$, $\theta_y=70^\circ$)

参考文献

- 1) 尾上博和, 塩野 充: “自動車画像からのナンバープレートの抽出とその漢字を含む全文字の切り出しと認識”, 電子情報通信学会論文誌, Vol. J 77-D-II, No. 3, pp.483-492 (1994-03).
- 2) アイテック情報技術教育研究所: “コンピュータグラフィックス入門”, アイテック (1995).
- 3) 安居院猛, 中嶋正之: “画像情報処理”, 森北出版, pp.193-194 (1994).

A Recognition Algorithm for Printed Numerals at Various Angles using a Recognition Tree

Takashi HAMAMOTO, Hiromoto KAWASHIMA*, Mitsuru OHKURA*
and Mitsuru SHIONO*

Graduate School of Engineering,

**Department of Information & Computer Engineering,*

Okayama University of Science,

Ridai-cho 1-1, Okayama 700, Japan

(Received October 7, 1996)

Currently, there has been a great deal of research concerning the extraction of character lines from images of everyday scenes. In such images, however, the character lines will appear at various angles, and so we generally have to conduct revision processing for the angles to recognize such characters. In this paper, we have tried to recognize printed numerals at various angles without conducting this revision. The principle methods of our recognition algorithm are labeling, thinning, the detection of the center of gravity of images and the making of convex hull images. We obtained an average recognition rate of 95.99 (%) with the algorithm using 1296 samples for each category.