

修士論文

文字の回転に対して頑健な
漢字認識手法

平成 24 年度修了

三重大学大学院工学研究科

博士前期課程 情報工学専攻

山村 昌史

はじめに

新聞，書籍，地籍図などの多くの紙媒体から文章情報を取り出し，計算機上で電子テキスト化する手段として OCR がある．電子テキスト化すれば，文書の再利用や紙媒体の保管場所の削減など，大きな利便性があるため，コンピュータや周辺機器の高性能化，低価格化に伴って，多くの OCR ソフトウェアが製品化され，様々な文書の電子化のために広く活用されている．しかし，地籍図やポスターなどでは，特定の文字が傾いていたり，文字列全体が曲線状や環状に記述されていたりする場合がある．既存の OCR で用いられる文字認識手法では回転した文字の認識を想定していないため，何らかの補正処理を行ってから認識する必要がある．

このような文字の回転に対処する方法として，行全体の傾きを推定することで，文字列の傾きを補正してから認識する方法がある．また回転に対して不変な特徴量を抽出することで，この問題に対処する手法も提案されている．しかし前者の場合，曲線に沿って文字が書かれているときには，文字単位の回転角度が異なるため対応できない．また後者の場合は，文字の回転角度情報が得られず，‘6’ と ‘9’ のような点対称の文字の区別が難しくなり，十分な認識精度が得られているとは言えない．

本研究では回転文字を認識するためのシンプルで効果的な手法を提案する．提案手法では，学習サンプルを人工的に回転させ，濃度こう配特徴を求め，文字種ごとに識別器を構成し擬似ベイズ識別関数で認識を行う．実験では，英数字・仮名文字・記号・第 1 水準漢字の計 3410 字種の印刷文字画像を対象とし，各字種 10 度間隔で生成した回転文字を学習させた識別器を用いて，テスト用回転文字を対象とした認識を行い本手法の有効性を調査した．その結果，認識率は 97.76% となり，上位 3 位までの累積認識率では，99.56% が得られた．さらに回転文字を学習しても通常の文字の認識への悪影響がほとんどないことも確認できた．また地籍図中の画像から切り出した回転文字を対象とした認識実験においても，従来の正立画像のみを学習した識別器では 49.58% であった認識率が，回転文字を学習した場合，92.37% と大きく向上した．

本手法では，ある文字種の回転文字はすべて同一クラスとして学習しているため，クラス数が少ないときは，回転文字を学習していない場合と比べて 1 文字の認識に要する処理

時間は変わらない。一方、認識対象を漢字を含む 3410 字種にする場合、実用的な認識速度にするために、計算量の少ない識別関数で候補字種を絞り込む大分類処理を行う。回転文字を学習していない場合、線形識別関数を用いれば認識率をあまり落とさずに 50 字種まで候補を絞り込めるが、回転文字を学習した場合は 1500 字種程度にしか絞り込めないことがわかった。そこで認識時に最も時間のかかる擬似ベイズ識別関数を OpenMP で並列化したところ、処理時間を並列化前の 21,4% に短縮できた。

本手法では回転文字と回転していない文字を同一クラスとして学習しているため、6 と 9 のように点対称の関係にある字種間の区別が難しい。よって今後の課題として、文字列情報を用いた点対称文字の認識精度向上や、GPGPU による認識処理の高速化を行うことが挙げられる。

目次

はじめに	i
第 1 章 序論	1
1.1 研究の背景	1
1.2 研究の目的	2
1.3 本論文の構成	3
第 2 章 提案手法	4
2.1 学習・認識処理の流れ	4
2.2 学習用回転文字の生成	5
2.3 濃度こう配特徴	7
2.4 擬似ベイズ識別関数	10
2.5 認識処理の高速化	12
第 3 章 実験	14
3.1 認識実験	14
3.2 実画像中の回転文字認識	18
3.3 大分類処理による高速化	20
3.4 並列処理による高速化	23
第 4 章 まとめと今後の課題	25
4.1 まとめ	25
4.2 今後の課題	26
付録 A OpenMP	28
A.1 OpenMP の概要	28
A.2 OpenMP の実装	28
付録 B 実験データとプログラム	29

B.1	文字画像サンプルデータ	29
B.2	プログラム	30
	謝辞	31
	参考文献	32

第 1 章

序論

1.1 研究の背景

近年の情報化社会の発展に伴い，日常生活において幅広く使用されている新聞，雑誌，地籍や地籍図などの紙媒体資料の電子化を行う機会が増えている．紙媒体から文章情報を取りだし，計算機上で電子テキストする手段として OCR がある．コンピュータやモバイルデバイス及び周辺機器の高性能化，低価格化によって，多くの OCR ソフトウェアが商品化され，様々な文書の電子化に対して広く活用されている．文書画像の電子テキストは，紙媒体と比較して文章の再利用や，保管場所の削減に関して大きなアドバンテージがあるため，文書内の文字抽出と，文字認識は非常に重要な技術となっている．現在，文字認識手法は，数多くの言語に対応し，手書き文字・署名照合にも用いられるようになった．

しかし，人間のように文字認識を行うためにはいくつかの問題があり，その一つに回転した文字の認識がある．地籍図 (図 1.1) やポスター (図 1.2)，書籍内の文章などでは，見る人の注目を引くために特定の文字を傾けたり，文字列全体が曲線状や環状に記述されている場合があるが，通常の OCR で用いられる文字認識手法は回転した文字の認識を想定しておらず，あらかじめ学習されている文字と認識対象文字の間で特徴量が大きく異なってしまう．

このような文字の回転に対処するための従来手法として，行全体の傾きを推定し，文字列の傾きを補正してから認識する手法が提案されている [1][2]．またニューラルネットワークを用いて回転文字を認識する手法 [3] や，画像の回転に対して不変な特徴を抽出することでこの問題に対処する手法も提案されている [4][5]．

しかし文字列の傾きの角度を推定する手法では，曲線に沿って文字が書かれているときには，文字単位の回転角度が異なるため対応できず，適用できる場面が限られる．また回転に不変な特徴量を抽出する手法の場合は，文字の回転角度情報が得られず，'6' と '9' のような点対称の文字の区別が難しくなり，十分な認識精度が得られているとは言えない．

また、漢字を対象とした認識手法 [6] も提案されているが、他は英数字を対象としたものが多く、多字種への適用例は少ない。

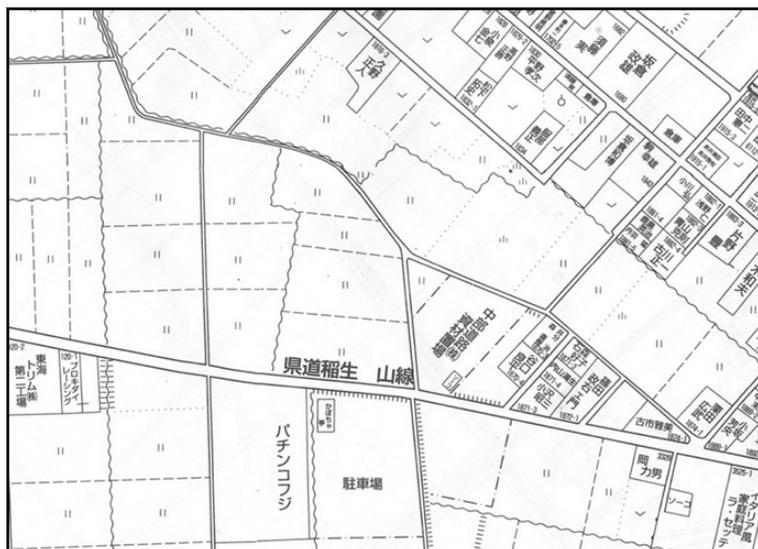


図 1.1: 地図の回転文字例



図 1.2: ポスターの回転文字例

1.2 研究の目的

本研究では回転文字を認識するためのシンプルで効果的な手法を提案し、英数字・漢字・仮名文字・記号に対する性能を検証する。提案手法では、学習サンプルを人工的に回転させ、濃度こう配特徴 (若林の HOG)[8] を求め、文字種単位で全回転角度の文字を学習した識別器を構成し、疑似ベイズ識別関数で認識を行う。一つの文字種の回転文字はすべて同一のクラスとして学習するため、クラス数が少ないときは回転文字を学習していない場合と比べて、1文字の認識に要する処理時間を変えずに認識を行うことができる。

識別器に回転文字を学習させておく方法には、パラメトリック固有空間法による回転文字の認識 [7] がある。この手法では 2 値パターンベクトルを特徴ベクトルとし、全クラス全サンプルの共分散行列で主成分分析を行って次元削減を行った超平面に特徴ベクトルを投影し、各クラスの単一フォントを回転させた軌跡と未知サンプルの間の最短ユークリッド距離を用いて識別を行うのに対して、提案手法ではこう配特徴を特徴ベクトルとし、原特徴空間でクラスごとに複数フォントの分布を近似する超平面を求め、疑似ベイズ識別関数によって識別を行う点が異なる。

また漢字を含む多字種の認識を行うことにより生じる、疑似ベイズ識別関数の各クラス間とのマッチング処理時間増加の問題に対処するために、処理時間が小さい識別関数を用いた大分類処理による候補字種の限定や OpenMP による並列処理の実装を行い、処理時

間の削減を試みる。

1.3 本論文の構成

第2章では、提案手法の処理の流れと、本手法で用いる濃度こう配特徴と疑似ベイズ識別関数、認識の高速化手法について説明する。第3章では、実験の条件と方法、結果について記述する。第4章では、まとめとして実験の考察と今後の課題について述べる。

第 2 章

提案手法

2.1 学習・認識処理の流れ

提案手法の学習と認識の流れを図 2.1 に示す．まず学習用データとして各字種の複数の文字サンプルから様々な角度の回転文字を人工的に生成する．次にそれらの画像に対して画像サイズの正規化などの前処理を行った後，特徴抽出を行い各字種の平均ベクトル，共分散行列の固有値，固有ベクトルを求め学習辞書とする．ある文字種の回転文字はすべて同一のクラスとして学習するため，クラス数が少ないときは回転文字を学習していない場合と比べて 1 文字の認識に要する処理時間を変えずに認識を行うことができる．評価する文字に対しても学習時と同じ手法で特徴抽出を行い，学習辞書とマッチングすることで分類を行う．特徴ベクトルには，濃度こう配特徴 (392 次元) を用いる [8]．また分類のための識別関数には擬似ベイズ識別関数 [9] を用いる．これらの詳細については次節で述べる．

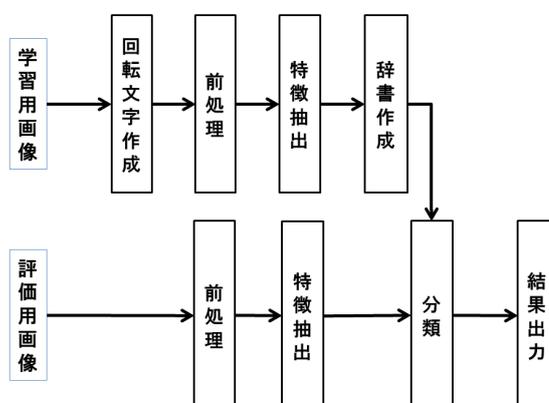


図 2.1: 学習と認識の流れ

2.2 学習用回転文字の生成

本手法では、学習文字画像を計算機上で回転させた画像を生成し、それらの画像から濃度こう配特徴を抽出して、回転文字の特徴を学習し様々な回転文字認識に対応できる識別器を構成する。文字の回転については、各学習画像サンプルに対して、角度 $\Delta\theta$ ごと、回転パターンを計 $360/\Delta\theta$ 個作成する。回転処理は元画像の重心 (cx, cy) まわりに行い、回転前の座標 (x, y) と回転後の座標 (x', y') は下式 (2.1) のような関係がある。回転後の座標 (x', y') を走査し、回転前の画像の座標 (x, y) の画素値を取得する。

$$\begin{aligned} x &= (x' - cx) * \cos \theta - (y' - cy) * \sin \theta + cx \\ y &= (x' - cx) * \sin \theta + (y' - cy) * \cos \theta + cy \end{aligned} \quad (2.1)$$

本研究では、 $\Delta\theta = 10^\circ$ に設定して、回転文字サンプルを生成する。例として数字 '1' を $\Delta\theta = 10^\circ$ として回転させた場合の例を図 2.2 に示す。

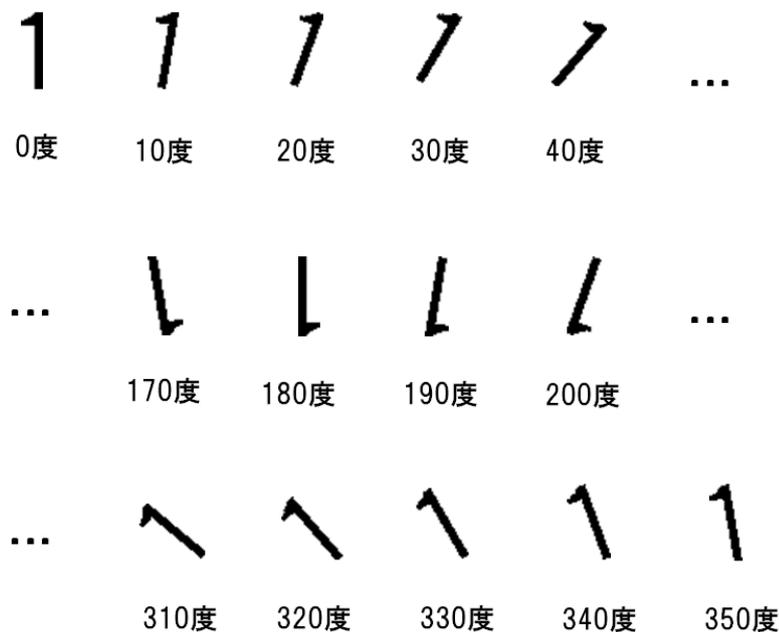


図 2.2: 回転文字サンプル

また (x, y) が整数値でない場合には画素値の線形補間を行って、文字輪郭のジャギーやかすれを抑制している。線形補間は求める座標 (x, y) の周辺 4 画素を使って、輝度値を直線的に補間して値を求める手法である。座標 (x, y) の輝度値 D は、周辺の 4 画素の輝度値 $C0 \sim C3$ と図 2.3 の距離 dx, dy を用いて、式 (2.2) のように表される。

$$D = (1 - dy) * ((1 - dx) * C0 + dx * C1) + dy * ((1 - dx) * C2 + dx * C3) \quad (2.2)$$

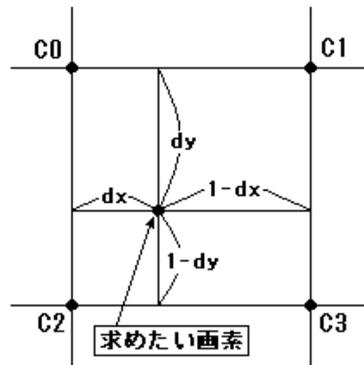


図 2.3: 輝度値の線形補間

図 2.4 の原画像に対して右まわり 10° の回転処理を行ったとき、線形補間なしとありの場合の画像をを図 2.5, 図 2.6 に示す。線形補間ありの方が文字輪郭がなめらかであることは明らかである。



図 2.4: 原画像



図 2.5: 線形補間なし



図 2.6: 線形補間あり

2.3 濃度こう配特徴

2.3.1 濃度こう配特徴とは

濃度こう配特徴とは、濃淡画像のこう配をその向きで量子化し、小領域ごとに強さを加算した特徴ベクトルである。方向を用いる特徴量として、加重方向指数ヒストグラム [9] があるが、2値画像の輪郭線を追跡するため、方向量子化数は多くて8方向までとなり、それ以上方向分解能を高めることが困難である。しかし、濃度こう配特徴では、画像の各画素点での濃度値のこう配を計算し、それを量子化することで16方向、32方向といった高い方向分解能をもつ特徴量が得られる。特に曲線の多い文字の認識において高い精度が得られることが知られている [8]。

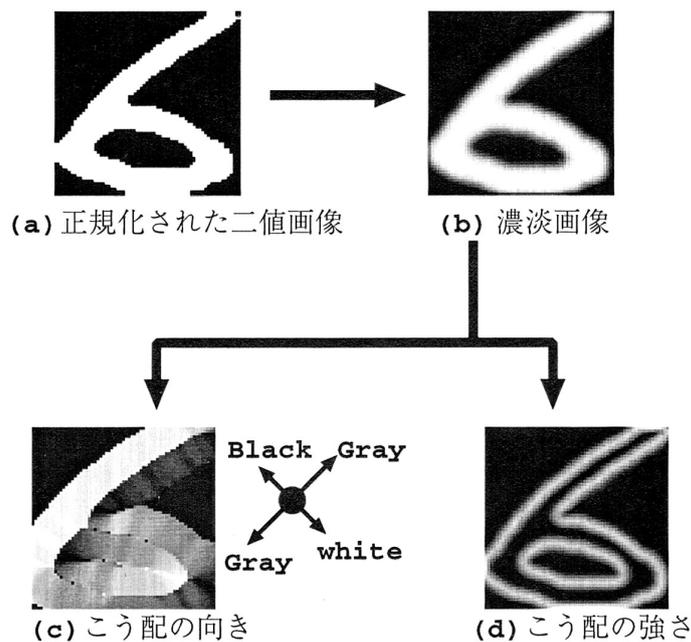


図 2.7: 濃淡画像から得られるこう配強度と方向

2.3.2 濃度こう配特徴の抽出方法

濃度こう配特徴の抽出手順を以下に示す．

- (1) 原画像 (2 値) に対して 2×2 の平均値フィルタによる平滑化処理を行い濃淡画像を作成する．
- (2) 山田の非線形正規化 [12] により，原画像から求めた 2 次元的線密度の水平，垂直方向への投影をそれぞれ均等化する変換関数を作成し，濃淡画像を非線形に正規化する．正規化画像サイズは 147×147 とする．
- (3) 3×3 の平均値フィルタを 3 回施して正規化画像を平滑化する．
- (4) 平均値が 0，最大値が 1 になるように各画素の濃度値を正規化する．
- (5) 式 (2.3) で表される Roberts フィルタを適用し，画像の濃度こう配の向き $\theta(i, j)$ と強さ $f(i, j)$ を求める．

$$\begin{aligned}\Delta u &= g(i, j) - g(i + 1, j + 1) \\ \Delta v &= g(i, j + 1) - g(i + 1, j) \\ \theta(i, j) &= \tan^{-1} \left(\frac{\Delta v}{\Delta u} \right) \\ f(i, j) &= \sqrt{(\Delta u)^2 + (\Delta v)^2}\end{aligned}\tag{2.3}$$

ここで $g(i, j)$ は処理対象画素 (i, j) での濃度値を表す．

- (6) こう配の方向を $\pi/16$ 刻みで向きを考慮した 32 方向に量子化する．また画像を 49×49 のブロックに分割し，こう配の強さを加算して $32 \times 49 \times 49$ 次元の局所方向ヒストグラムを作成する．
- (7) 加重フィルタ [1 4 6 4 1] を 1 方向おきに掛けることでヒストグラムを 32 方向から 16 方向に再標本化する．その後，更に加重フィルタ [1 2 1] を 1 方向おきに掛けて 16 方向から 8 方向にする．16 方向から 8 方向へ方向数を削減する場合の，加重フィルタの掛け方を図 2.8 に示す．

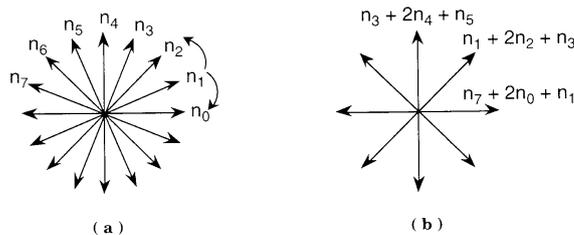


図 2.8: 方向量子化数の削減

- (8) 上記のヒストグラムに対して, 31×31 の 2 次元ガウスフィルタ (図 2.9) を縦・横 7 ブロックおきに施すことで (図 2.10), ブロック数を 49×49 から 7×7 に削減して 392 次元 (8 方向 \times 7 ブロック \times 7 ブロック) の特徴ベクトルを求める.

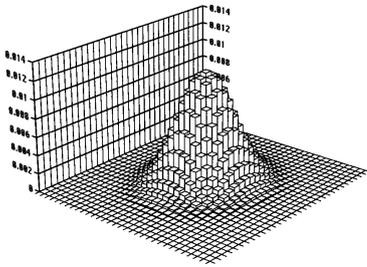
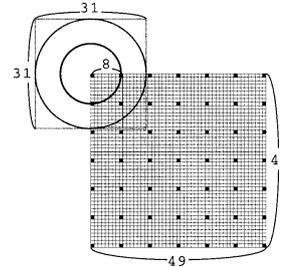
図 2.9: 31×31 の 2 次元ガウスフィルタ

図 2.10: 標本点の配置

- (9) 擬似ベイズ識別関数は, 観測値が多変量正規分布であるという仮定のもとでベイズの決定則から導かれる. 従って観測値が多変量正規分布でない場合には, その性能が低下する. そのため, より正規分布らしく見えるようにする変数変換処理が有効となる. そこで式 (2.4) の変数変換 (べき乗) により, 特徴量の分布を正規分布に近づける.

$$y = x^\nu \quad (2.4)$$

ここで x は変換される値であり, ν は指数となっている. 本研究では, $\nu = 0.5$ として変数変換を行っている.

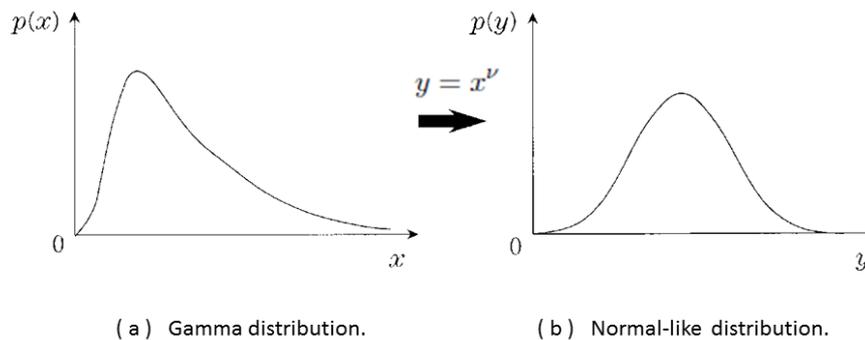


図 2.11: 変数変換

2.4 擬似ベイズ識別関数

識別関数には式 (2.5) の擬似ベイズ識別関数 (MQDF) を用いる。この識別関数は、分布パラメータのうち母集団の共分散行列が未知の正規分布に対する最適識別関数から導出された近似式 [7] で、識別精度を損なうことなく計算量を大幅に削減できるのが特長である [8]。

$$g(X) = (N + N_0 + n - 1) \ln \left[1 + \frac{1}{N_0 \sigma^2} \left[\|X - M\|^2 - \sum_{i=1}^k \frac{\lambda_i}{\lambda_i + \frac{N_0}{N} \sigma^2} \{\Phi_i^T (X - M)\}^2 \right] \right] + \sum_{i=1}^k \ln(\lambda_i + \frac{N_0}{N} \sigma^2) - 2 \ln P(\omega) \quad (2.5)$$

X : 入力文字の特徴ベクトル

n : 特徴ベクトルの次元数

N : 各クラスの学習サンプル数

M : 平均ベクトル

Φ_i, λ_i : 共分散行列の第 i 固有ベクトルと第 i 固有値

N_0 : σ^2 の信頼度定数

k : 識別に用いる固有ベクトル数

$P(\omega)$: クラス ω の事前確率

最適識別関数はベイズ学習に基づく識別関数であるため事前分布が導入されているが、式 (2.5) での σ^2 は特徴ベクトル X の事前分布を球状と仮定した場合の分散であり、 N_0 は σ^2 の信頼度を表す定数で信頼度定数と呼ぶべきものである。

本研究では、 M は標本の平均ベクトルで代用し、 σ^2 の値は全字種、全固有値の平均を用いる。 $P(\omega)$ を含む項はクラス間で共通として省略する。また N_0 の値を、

$$N_0 = \frac{\alpha}{1 - \alpha} N \quad (2.6)$$

とし、 $0 < \alpha < 1$ の範囲で α を選択する。

ここで α の値が 0 に近づくとき，式 (2.5) は投影距離法や部分空間法に近い性質とを示し， α の値が 1 に近づくとき，ユークリッド距離による識別を近似することが分かっている．

実験では， α の値を 0~1 まで変化させて，特徴ベクトルの母集団の分布の変化に対応している．そのため，カテゴリーの決定境界は投影距離法とユークリッド距離の中間的な超平面になる．

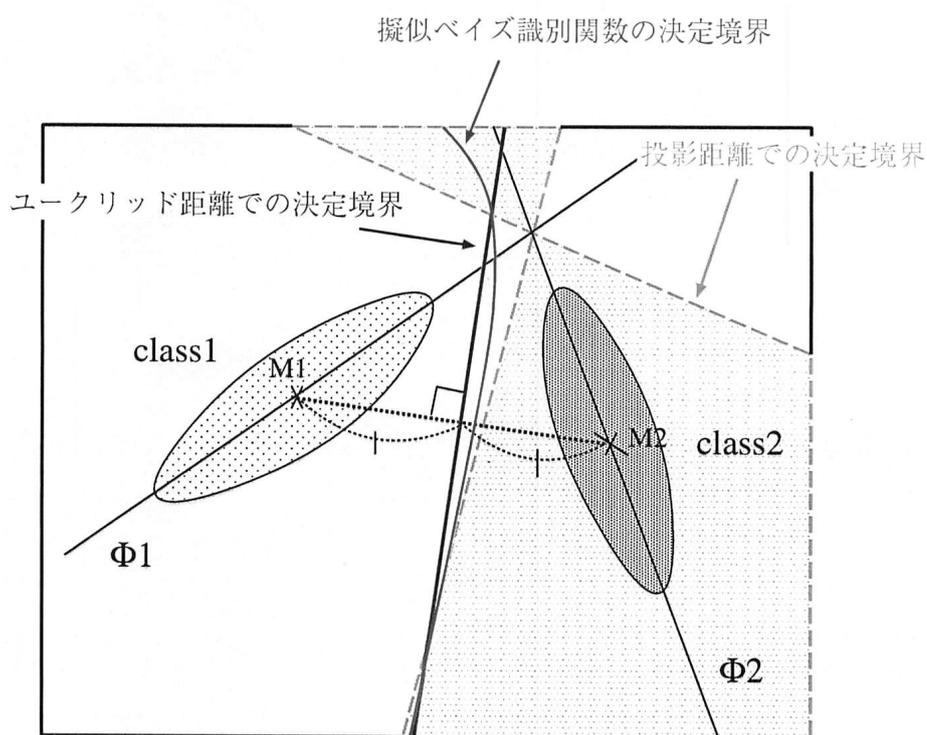


図 2.12: 擬似ベイズ識別関数の決定境界 (2 次元)

2.5 認識処理の高速化

2.5.1 大分類処理による高速化

漢字・仮名文字はクラス数が多く、計算量の多い識別関数を全クラスに対して適用することは認識速度が大幅に低下してしまう。そのため、あらかじめ計算量の少ない識別関数による大分類処理で候補字種を絞った後に詳細識別を行うことによって、認識速度を向上させる。計算量の少ない識別関数としては、シティーブロック距離、ユークリッド距離、線形識別関数などがある。本研究では、以下に定義するユークリッド距離 (2.7) と線形識別関数 (2.8) を用いて大分類を行う。

・ユークリッド距離

$$g_l(X) = \|X - M\| = \sqrt{\sum_{i=1}^n (x_i - m_i)^2} \quad (2.7)$$

X : 入力文字の特徴ベクトル

M : 第 l クラスの平均ベクトル

n : 特徴ベクトルの次元数

・線形識別関数

$$\begin{aligned} g_l(X) &= W_l^T X + W_{l0} \\ W_l &= S_\omega^{-1} M_l \\ W_{l0} &= -\frac{1}{2} M_l^T S_\omega^{-1} M_l \end{aligned} \quad (2.8)$$

X : 入力文字の特徴ベクトル

l : クラスを表す添え字

M_l : 第 l クラスの平均ベクトル

S_ω : 級内共分散行列

線形識別関数の 1 クラスあたりの計算量は、 n 次元の内積計算 1 回 (積和演算が 1 回) と W_{l0} の加算が 1 回である。またユークリッド距離の計算量は減算・積和演算が n 回である。

2.5.2 処理の並列化による高速化

文字認識を行う際、認識対象の文字から得られた特徴ベクトルと各クラスの学習辞書との相違度を算出して分類処理を行っている。従来の処理手順では、各クラスとの間の相違度計算を逐次的に処理しているため、複数のプロセッサを使用できる環境であってもCPUは一つしか使われていなかった。そこで本研究では認識部で処理時間のかかる疑似ベイズ識別関数の計算にOpenMPによる並列化を実装する。逐次処理の記述されたプログラムを並列化した場合の流れを図2.13に示す。並列化される部分を並列リージョンと呼び、並列化された実行単位をスレッドという。本研究では認識処理で疑似ベイズ識別関数で評価値を求める場合に、各クラスとの間で逐次的に計算を行っていた処理を並列化することで実処理時間の削減を行う。これにより、1文字あたりの処理時間の高速化が期待できる。OpenMPについての詳細は付録Aにて説明する。

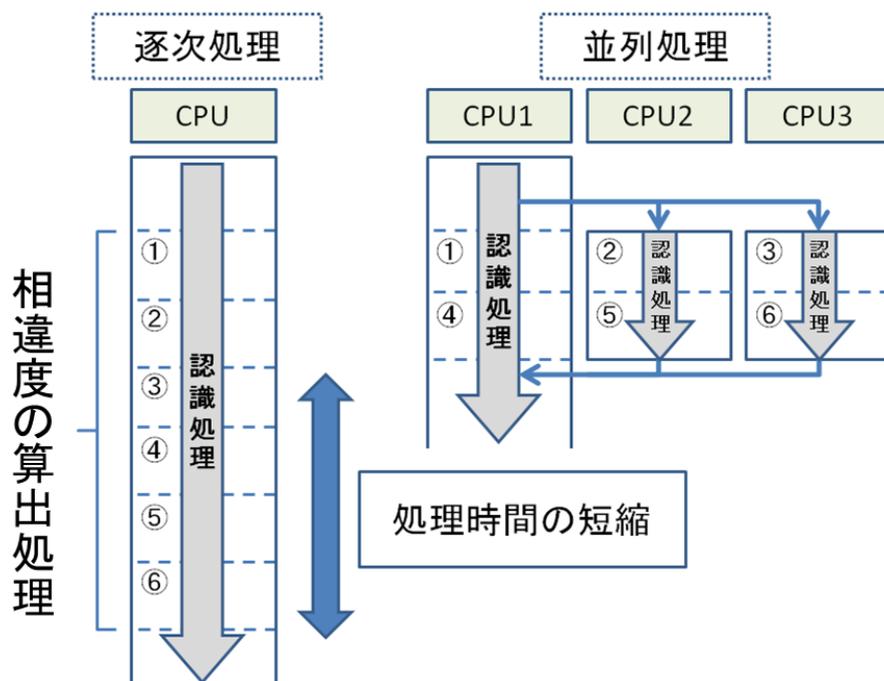


図 2.13: 並列処理の流れ

第 3 章

実験

3.1 認識実験

回転文字を学習することで，学習しない場合に比べてどの程度回転文字の認識率が向上するか，また回転していない文字の認識率が低下するか調べる．そのために，回転文字を学習していない辞書，学習した辞書と，回転文字を含まない評価データ，回転文字を含む評価データの組み合わせで実験を行う．本実験では，先に記述した擬似ベイズ識別関数で用いる固有ベクトル数を $k = 90$ で固定し，擬似ベイズ識別関数のパラメータ α の値を 0.1~0.9 まで 0.1 刻みで変化させて認識率がどのように変化するかを調べた．

3.1.1 実験データ

本実験では，300dpi でスキャンした活字文章画像から切り出した印刷文字データを使用する．対象となる字種は英数字・仮名文字・JIS 第 1 水準漢字・記号を含む合計 3410 字種である．第 1 水準漢字とは常用漢字や人名用漢字など比較的使用頻度の高い漢字群を表す．それぞれの字種数の内訳は次のようになっている．

分類	字種数
英数字	62
仮名文字	147
記号	236
第 1 水準漢字	2965
合計	3410

学習・評価用画像例を図 3.1 に示す．サンプルの中には，図の例に示すような様々なフォント，サイズの文字画像が含まれている．これにより，単一フォントだけでなく，マルチフォントに対する回転文字認識の性能評価を行う．また，各字種に対してそれぞれ学習用画像データに 150 サンプル程度の文字画像を使用し，それぞれの画像に対して 10° ごとに回転させたものを生成し学習に用いる．回転文字の学習用画像の枚数は合計で，17,818,374 枚となる．評価用画像データには，各字種 10 サンプル程度の文字画像を使用し，学習用データと同様に 10° ごとに回転させたものを生成し評価に用いる．回転文字の評価用画像の枚数は合計で，992,556 枚となる．



図 3.1: 文字サンプル

3.1.2 実験結果と考察

認識実験の結果を表 3.1 に示す．回転文字の認識では，回転文字を学習していないとき，12.17%($\alpha = 0.1$) だった認識率が回転を学習したとき 97.76%($\alpha = 0.1$) となり，大きく向上した．また回転していない文字の認識率は，回転を学習していないとき 99.41% だったものが，回転を学習したとき 97.87%($\alpha = 0.1$) とやや低下した．誤認識では，‘6’ と ‘9’ や ‘一’ と ‘1’ のように回転すると同じ形になる点対称字種間の誤認識が多く発生した．これは，回転文字を学習することで二つのクラスの特徴ベクトルの分布が同じになってしまったためだと考えられる．次にこれらの回転によって類似する文字間の誤認識が全体のパフォーマンスに与える影響を確認するために，認識結果の上位 5 位までの累積認識率を求め，図 3.3 に示す．その結果上位候補 3 位までの累積認識率は 99.56% となり，回転していない文字の認識における上位候補一位での認識率と同程度の結果が得られた．この結果から回転により類似する字種が認識率に与える影響が大きいことが分かった．

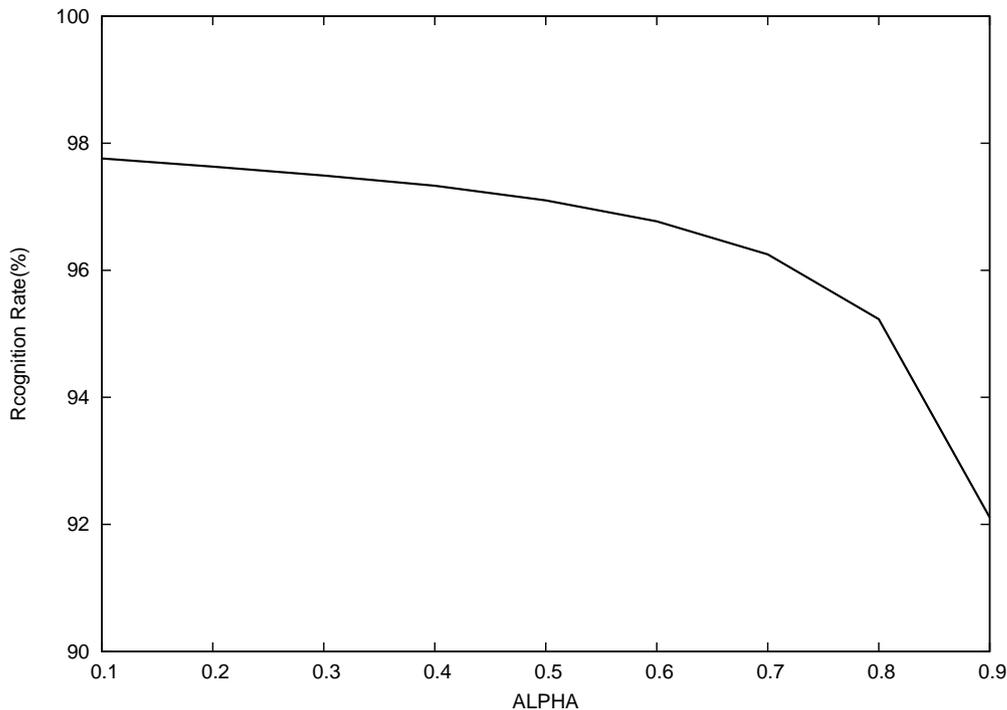


図 3.2: 擬似ベイズ識別関数の と認識率の関係

表 3.1: 認識率 (単位 [%])

学習 \ 評価	回転文字含まない	回転文字含む
回転文字含まない	99.41	12.17
回転文字含む	97.87	97.76

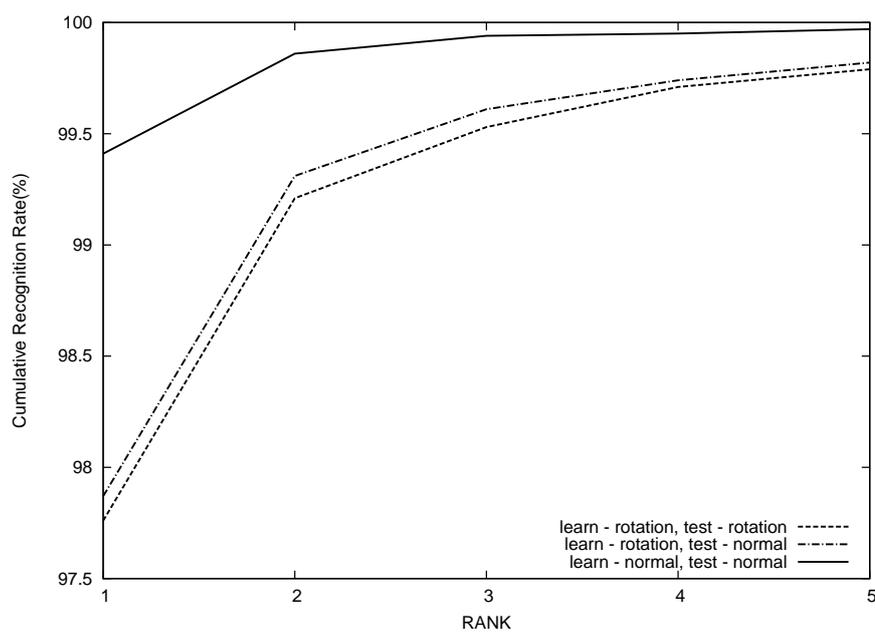


図 3.3: 上位 5 位までの累積認識率

表 3.2: 誤認識上位 5 組

正解字種	誤認識字種	誤認識数
9	6	223
/	\	196
]	[196
,	'	183
()	173

3.2 実画像中の回転文字認識

前節では計算機上で仮想的に回転させた文字を評価用画像として用い、本手法の有効性を確認した。次に応用例として、実際の画像中に含まれる回転文字を正しく認識できるかどうか検証するために、図 3.4 に示すような地籍図などの画像から外接枠で切り出された文字を評価用画像として認識実験を行う。本実験では 5 枚の画像中から切り出しに成功した活字印刷文字画像 236 文字を評価用データに用いた。切り出した文字画像例を図 3.5 に示す。

3.2.1 結果と考察

実験を行った結果の、認識率を表 3.3 に示す。また、回転文字を学習した場合の認識実験における誤認識例について表 3.4 に示す。従来の回転文字を学習しなかった場合認識率は 49.58% だったのに対し、回転文字を学習したときの認識率が 92.37% となり認識精度は大きく向上し、この結果から回転文字を学習することで実画像中の回転文字に対しても本手法の有効性を確認できた。しかし、今回の実験では評価画像のサンプル数が少ないため、より多くの画像に対して検証を行う必要があると思われる。また、誤認識例の中には切り出しの時点でかすれた文字など低画質な文字を含む場合があり、回転文字だけでなくこのような問題点への対応も行う必要がある。

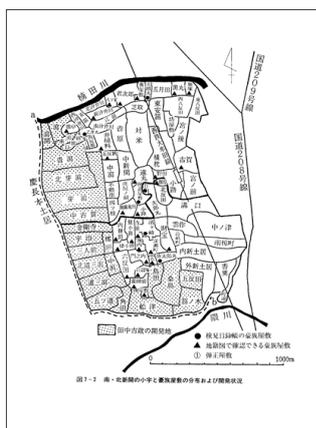


図 3.4: 対象画像



図 3.5: 切り出した文字画像例

表 3.3: 認識結果

	認識率 (%)	認識枚数 (枚)
回転文字を学習しなかった場合	49.58	117 / 236
回転文字を学習した場合	92.37	218 / 236

表 3.4: 誤認識例

正解字種	誤認識字種
か 	&
て 	τ
学 	&
口 	u
向 	4
向 	可
田 	円
未 	¥
未 	キ
未 	乗
来 	4

3.3 大分類処理による高速化

3.3.1 実験条件

大分類による高速化の効果を確認するために、ユークリッド距離と線形識別関数で大分類を行った場合の処理時間の比較実験を行った。大分類後に行う詳細分類には擬似ベイズ識別関数を使用し、各パラメータの値は $k = 90, \alpha = 0.1$ とする。また回転文字学習による大分類処理への影響を確認するために、回転文字を学習していない場合と回転文字を学習した場合の、候補字種と認識率の推移についての比較を行う。対象となる評価用データは前節の認識実験で用いた活字印刷文字画像を使用し、回転文字学習をしない場合の実験では、回転文字を含まない 13,831 枚に対して実験を行い、回転文字学習を行う場合には回転文字を含む 497,916 枚に対して実験を行う。処理時間の計測には、Linux の `time` コマンドを使用して CPU 時間の計測を行い、上記の認識枚数の平均値から値を算出する。

3.3.2 結果と考察

回転文字を学習しない識別器に対して、ユークリッド距離と線形識別関数を使用し大分類処理を行った場合の、認識率と処理時間の推移を図 3.6、図 3.7 に示す。次に回転文字を学習した識別器に対してユークリッド距離と線形識別関数を使用し大分類を行った結果の、認識率と処理時間の推移を図 3.8、図 3.9 に示す。回転していない文字のみを学習した場合、線形識別関数で詳細分類の候補数を 50 候補に設定した場合に認識率が収束した。また処理時間については大分類を行う以前は 120.58[ms] であったのに対し大分類後は 5.12[ms] にまで短縮することができた。また回転文字を学習した識別器においては、線形識別関数で詳細分類後の候補数を 1500 候補に設定した場合に認識率が大分類前と同等の認識率が得られた。またそのときの処理時間は 120.58[ms] から 56.58[ms] まで処理時間を短縮させることができたが、回転文字を学習しない場合の大分類処理時と比べると、大きく絞り込みを行うことができなかった。この原因として、回転文字を学習したために、各字種の特徴量が特徴空間上に分布する範囲が広がってしまい、各クラス間の分離性が悪くなったことでユークリッド距離や線形識別関数では上位の候補に絞り込みが難しくなってしまったためであると思われる。

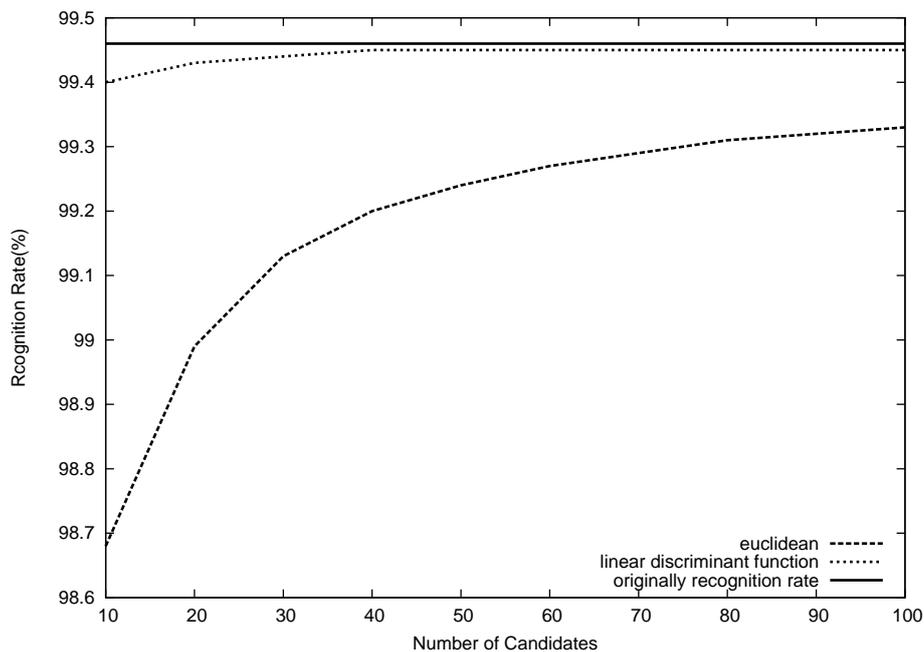


図 3.6: 大分類の候補字種数と認識率の関係 (回転文字学習なし)

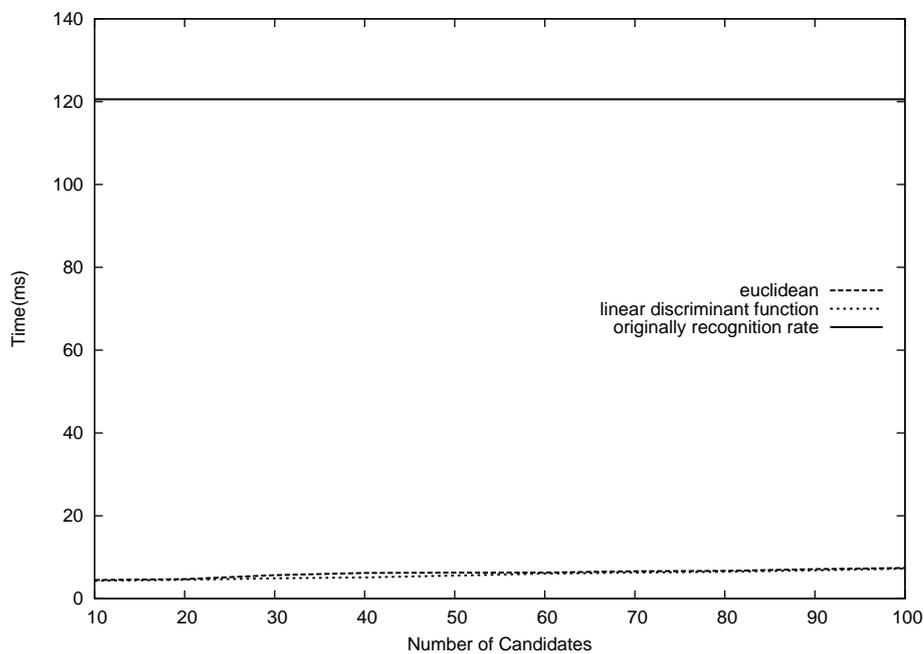


図 3.7: 大分類の候補字種数と処理時間の関係 (回転文字学習なし)

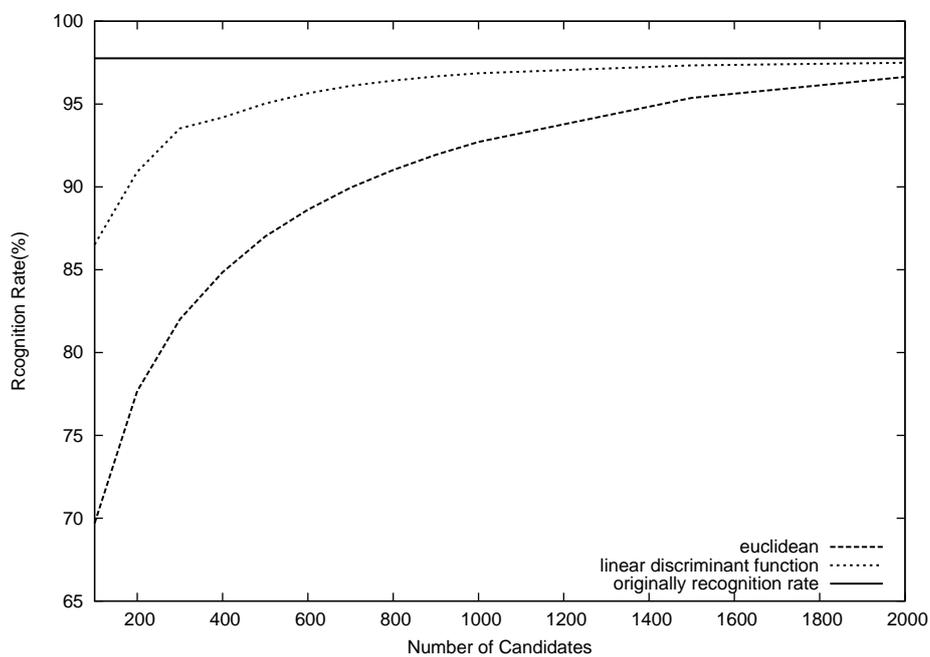


図 3.8: 大分類の候補字種数と認識率の関係 (回転文字学習あり)

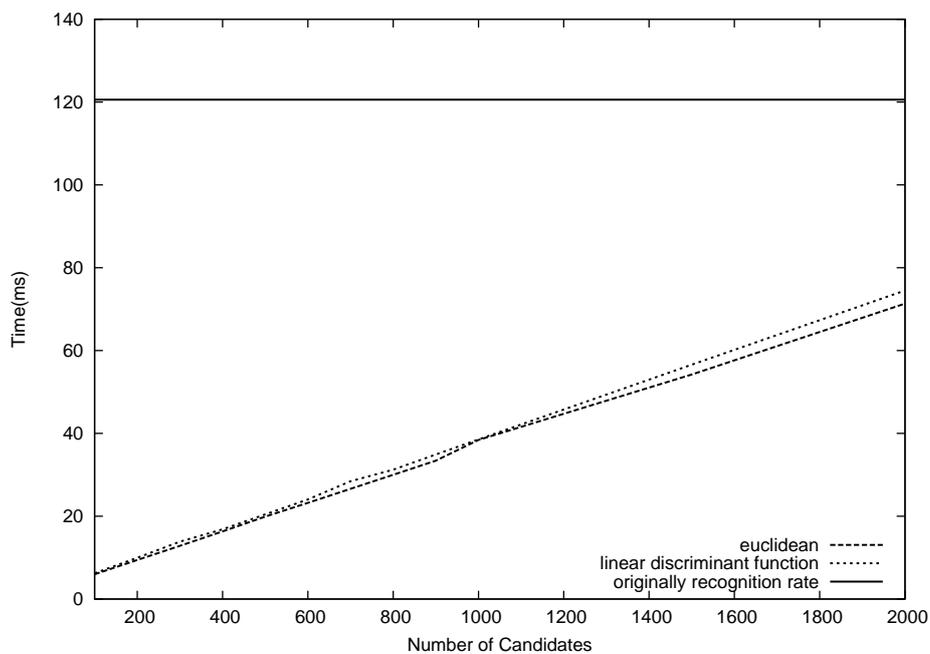


図 3.9: 大分類の候補字種数と処理時間の関係 (回転文字学習あり)

3.4 並列処理による高速化

3.4.1 実験条件

回文字認識における擬似ベイズ識別関数を用いる認識部の並列化の効果を検証するために、OpenMP によるマッチング処理の並列化を行い、認識処理にかかる時間を計測した。擬似ベイズ識別関数の各パラメータの値は $k = 90, \alpha = 0.1$ とする。また本実験では、並列処理による認識部高速化の効果のみを確認するために、前節の大分類処理による高速化については同時に行わないものとする。処理時間の測定には Xeon X5670@2.93GHz(6core) を搭載した計算機を使用する。並列処理のスレッド数は 1~6 までの 6 通りで実験した。対象のデータには前節の認識実験で用いた活字印刷画像サンプルデータ 13,831 枚を使用し、1 文字あたりの平均値を算出する。時間の計測には、大分類処理と同様に Linux の time コマンドを使用し CPU の処理時間である user time を計測する。評価方法には、逐次処理と比較してどの程度処理時間が短縮したかを表すための、以下の時間短縮率で求める。

$$\text{時間短縮率 [\%]} = \frac{\text{並列処理時間}}{\text{逐次処理時間}} \times 100 \quad (3.1)$$

3.4.2 結果と考察

認識部のマッチング処理に対して並列処理を実装した結果を、表 3.5 に示す。またスレッド数ごとの認識部処理時間を計測した結果と、実際に並列処理を遅延なしに実行できた場合の理想的な処理時間を理想値とし、その一つの処理時間の推移を図 3.10 に示す。1 スレッドで認識処理を行った場合の処理時間は 1 文字あたり 129.98[ms] であったが、並列数を増加させるごとに処理時間が減少し、6 スレッドで並列化した場合処理時間は 1 文字あたり 27.87[ms] となった。またこのとき、並列化による時間短縮率は 21.4% となりスレッド数の増加に伴い処理時間の減少が確認できた。また理想値どおりの時間短縮率が得られなかった原因としてメモリアクセスの衝突や同期処理などのオーバーヘッドが発生しているためであると思われる。

表 3.5: 認識部の並列処理結果

並列スレッド数	1	2	3	4	5	6
1文字あたりの処理時間 [ms]	129.98	65.49	44.60	34.93	30.25	27.87
時間短縮率 [%]	100	50.4	34.3	26.9	23.3	21.4
理想値 [ms]	129.98	64.99	43.33	32.50	25.99	21.66

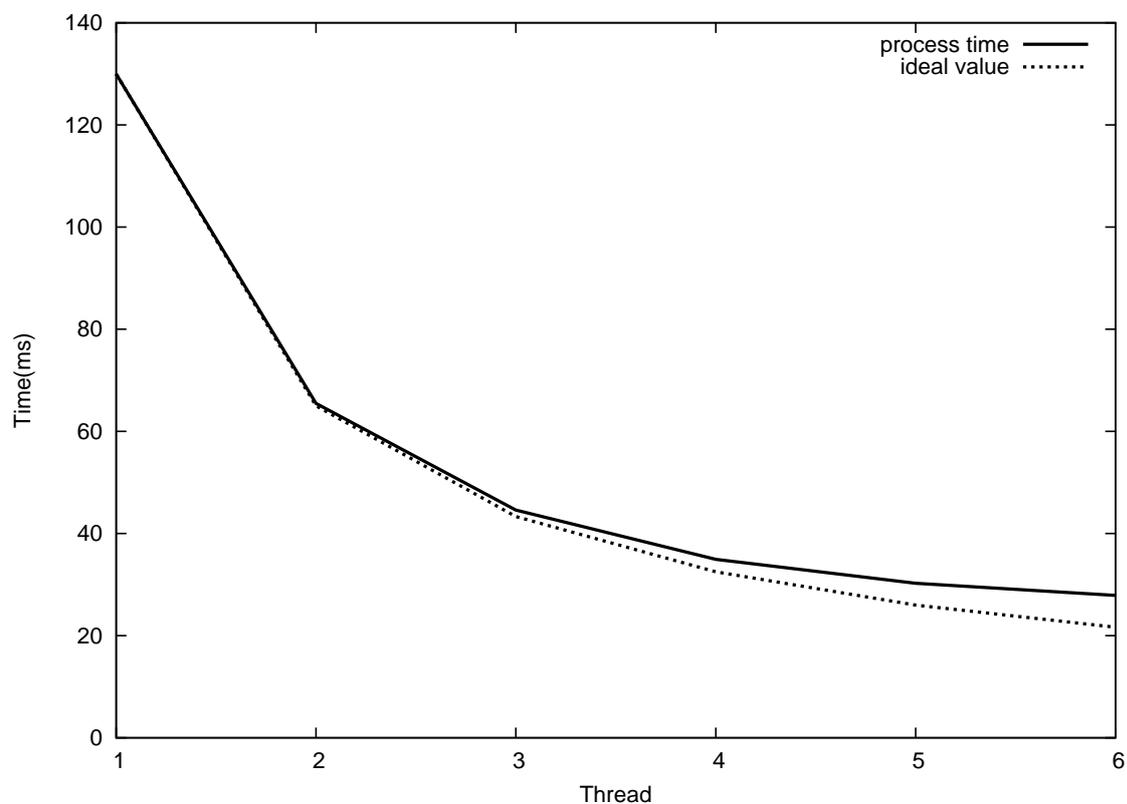


図 3.10: 並列スレッド数と認識部処理時間の関係

第 4 章

まとめと今後の課題

4.1 まとめ

本研究では回転文字を認識するために、濃度こう配特徴から得られた様々な角度の文字特徴を学習した識別器に対して擬似ベイズ識別関数を用いることで、回転に頑健な文字認識を実現する手法を提案し有効性を検証した。評価実験として活字印刷漢字 3410 字種を対象とする認識を行ったところ、10 度間隔で回転した文字を学習させることにより、認識率は 97.76% となり回転文字を学習させない場合の認識率 12.17% から大きく向上し、回転した文字の認識にも対応できるようになったが、‘6’ と ‘9’ のような点対称文字間の誤認識が多く発生した。これは、回転文字を学習することで二つのクラスの特徴ベクトルの分布が類似してしまっただと考えられる。しかし、上位候補 3 位までの累積認識率を確認した結果 99.56% と高い認識率が得られたことから、回転により特徴が類似する字種間の影響があっても早期に認識率が収束することが分かった。このことから提案手法は回転文字に対して有効であることが確認できた。また、実際の画像中の回転文字を評価対象として認識実験をところ、従来の回転文字特徴を学習しない場合には 49.58% であった認識率が、回転文字を学習した場合 92.37% と大きく向上し高い認識精度が得られることが検証できた。今回の実験では評価用文字のサンプル数が少なかったこともあり、今後更なるサンプル数の追加を行い実験を行う必要があると思われる。

次に多クラス分類における認識処理時間への対応として、大分類処理と並列化による処理時間の削減を検証した。大分類処理では線形識別関数を用いた場合、回転文字を学習しない場合には、詳細分類の候補数を 50 字種程度に絞り込むことが可能であったが、回転文字を学習した場合には、詳細分類の候補数を約 1,500 字種程度にしか絞り込むことができないことがわかった。並列処理による認識部の高速化の検討では OpenMP による並列化を実装しその効果の検証を行った。その結果、6 並列処理を行ったとき 21.4% の時間短縮率となり認識部の並列処理化による効果を確認できた。本研究では CPU のみを用いた

並列化を検証したが、今後更なる高速化を目指すために、GPU を利用した高速化を行うことなどが求められる。

4.2 今後の課題

今後の課題として以下のもの挙げられる。

- 単独では角度推定が困難な文字への対処
実際の画像から切り出した文字に対する認識では、一文字単位で抽出したものを認識しており、文字列や単語での抽出を行ってはいない。そのため、回転することで他の字種に類似する文字の識別が困難である。よって、文字列や単語単位での抽出を行い、近隣の文字と認識結果を比較することで点対称字種間の識別精度を向上させることを考えている。
- 地籍図のつなぎ合わせへの応用
地籍図のつなぎ合わせの研究 [13] では、地籍図の境界線や図中の方眼格子を利用して各地籍図のつなぎ合わせを行っているが、より高速で高精度な手法の実現のために、地籍図内の隣接区画のラベルを識別することによりあらかじめ候補を絞ることが課題となっている。これらのラベルは地籍図にそって書かれており、文字が回転しているものが多い(図 4.1)。そのため、ラベルの文字を認識するための手法として本研究を利用することが今後の課題として挙げられる。



図 4.1: 地籍図内のラベル例

- GPGPU による認識処理の高速化

本研究では、認識処理速度を向上させるため大分類処理による識別候補の削減や、並列処理による認識を検証した。更なる高速化を目指すために、GPGPU によって認識処理を行うことが挙げられる。GPGPU はグラフィックス描画用の GPU (Graphics Processing Unit) の演算性能を、汎用的に利用する概念、技術であり、大量の計算を並列化して高速に実行することができる。よって GPU の性能を活かして、どの程度高速化できるか検証することが今後の課題として挙げられる。

- 認識手法の改良

今後、更に回転文字認識の精度を向上させるために、手法の改良が求められる。そのために、濃度こう配特徴抽出時のブロック分割の方法や再標本化時に行うガウスフィルタなどを調整することで、より回転文字に適した特徴量の抽出法を考案し比較実験を行うことが挙げられる。

付録 A

OpenMP

A.1 OpenMP の概要

OpenMP(Open Multi-Processing) は共有メモリ型並列プログラムを作成, 実行するための枠組みである [14]. OpenMP の特徴として以下のことが挙げられる.

- (1) 既存の逐次プログラムをベースに並列プログラムを作ることができる
- (2) 指示文を使って、スレッドを生成、制御することができ、スレッドライブラリなどを使うよりも簡単にスレッドプログラミングができる
- (3) 逐次プログラムから段階的に並列化していくことが可能
- (4) 基本的に、OpenMP の指示文を無視することにより、逐次と並列プログラムを同じソースで管理するが可能

A.2 OpenMP の実装

OpenMP は共有メモリ上でプロセッサによる複数の実行の流れを制御するプログラムを記述する. OpenMP 共有メモリ上でプロセッサによる複数の実行の流れの制御をプリAGMA・ディレクティブ (#pragma) と呼ばれるコンパイラに対する指示文で行う. OpenMP の指示文は、プログラム内で並列化を行う場所に

```
#pragma omp OpenMP 指示文 ...
```

を挿入することで並列化の方法を指定する. 本実験では、処理分散指示文である

```
#pragma omp parallel for
```

を用いることで疑似バイズ識別関数における各評価値算出処理の並列化を行っている.

付録 B

実験データとプログラム

本研究で使用した画像データとプログラムを/net/xserve0/users/yamamura/MasterStudy/のディレクトリに置く．またその内容は以下の通りである．

B.1 文字画像サンプルデータ

MasterStudy/data/

数字・アルファベット・記号・仮名文字・漢字の各文字画像のサンプルデータ．各字種のディレクトリ名は Shift-jis コードでラベル付けされている．

```
data/  
|  
|-- data_alphanumeric/  
|   # 英数字文字データ  
|-- data_kana/  
|   # 仮名文字データ  
|-- data_kanji1/  
|   # 漢字データ 1 (Shift-jis コード 889F ~ 8FFC)  
|-- data_kanji2/  
|   # 漢字データ 2 (Shift-jis コード 9040 ~ 9872)  
|-- data_symbol/  
|   # 記号文字データ
```

B.2 プログラム

- ・回転文字認識プログラム

MasterStudy/koubai392_16to8_CDA/

ディレクトリ内のプログラムは以下の通りになっている．それぞれの詳細は README に記載する．

```
koubai392_16to8_CDA/
|
|-- cfex.cpp
|   # 画像の回転処理を行った後，特徴抽出を行うプログラム
|-- ffunc.h
|   # 特徴抽出プログラムヘッダファイル
|-- Dictionary.c
|   # 学習辞書作成を行うプログラム
|-- Recog.c
|   # 認識処理を行うプログラム
|-- smart/
|   # Fortran サブルーチンのライブラリ
|-- autogfort
|   # 特徴抽出・学習辞書作成・認識の一連の処理を制御するシェルスクリプト
|-- README
|   # 各プログラムの詳細
```

謝辞

本論文は様々な方々の協力により執筆することができました。研究を進める過程において多くの助言と提案，ご指導をしていただきました木村文隆教授，研究分野における様々な技術をご教授下さった若林哲史准教授，資料作成やプレゼンテーションのコツや研究における様々なアドバイスを下さった大山航助教に深く感謝いたします。

また，毎回のディスカッションにおいて専門的観点から様々な御指導をしていただきました三宅康二名誉教授，日頃お世話になった田中みゆき事務官，多くの有益なアドバイスや御指導と楽しい研究室生活を与えて下さったヒューマンインタフェース研究室の皆様にも深く感謝いたします。

最後になりましたが，長きにわたる学生生活を支えてくれた両親に対し今一度感謝の意を表し本論文の結びとさせていただきます。

参考文献

- [1] 林 俊成, 成田 誠之助: “最小 2 乗法を用いた文字列傾き補正”, 電子情報通信学会総合大会, D-12-28, 2000
- [2] 土橋 外志正, 久保田 浩明: “文書画像に対する高速な傾き補正手法”, 電子情報通信学会総合大会, D-12-27, 2000
- [3] 鳥潟 文男, 木村 重規, 鈴木 章夫, 矢野 雅文: “Hough 変換を用いた回転と大きさに不変な文字認識ニューラルネットワーク”, PRMU99-88, pp1-6, 1999
- [4] 横田 智彦, 大山 航, 若林 哲史, 木村 文隆: “回転不変濃度こう配特徴とその顔検出への応用”, MIRU, ISI-3 pp428-433, 2009
- [5] P. P. Roy, U. Pal, J. Llads and F. Kimura, “Convex Hull based Approach for Multi-Oriented Character Recognition from Graphical Documents,” International Conference on Pattern Recognition (ICPR), 2008
- [6] Tai-Ning Yang, Sheng-De Wang, “A rotation invariant printed Chinese character recognition system,” Pattern Recognition Letters 22, pp85-95, 2001
- [7] 篠川 敏行, 長谷 博行, 米田 政明: “パラメトリック固有空間による回転文字の認識”, 信学技報 PRMU102(707), pp73-78, 2003
- [8] 澤 和宏, 若林 哲史, 鶴岡 信治, 木村 文隆, 三宅康二: “こう配特徴ベクトルと変動吸収共分散行列による手書き漢字認識の高精度化”, 電子情報通信学会論文誌 D-I I, Vol.J82-D-II, No.11, pp.1-12
- [9] 鶴岡 伸治, 栗田 昌徳, 原田 智夫, 木村 文隆, 三宅 康二: “加重方向指数ヒストグラム法による手書き漢字・ひらがな認識”, 電子情報通信学会論文誌 D, Vol.J70-D, No.7, pp.1390-1397
- [10] 若林 哲史, 鶴岡 伸治, 木村 文隆, 三宅 康二: “特徴量の次増加による手書き数字認識の高精度化”, 電子情報通信学会論文誌 D-II, Vol.J77-D-II, No.10, pp.2046-2053

- [11] 若林 哲史, 鶴岡 信治, 木村 文隆, 三宅 康二, “手書き数字認識における特徴選択に関する考察”, 信学論 (D-II), vol.J78-D-II, no.11, pp.1627-1638, 1995
- [12] 山田 博三, 斎藤 泰一, 山本 和彦, “非線形正規化法の改良”, 昭 53 信学春季全大, D-439
- [13] 市川貴大, 大山航, 若林哲史, 木村文隆: “方眼格子を利用する地籍図のつなぎ合わせ”, 平成 22 年度電気関係学会東海支部連合大会, Q3-4, 2010
- [14] 北山洋幸, “OpenMP 入門 マルチコア CPU 時代の並列プログラミング”, 秀和システム, 2009