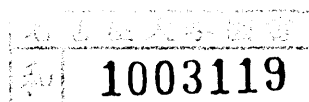


手書き文字の機械認識における 特徴抽出と識別関数の研究

鶴 岡 信 治

手書き文字の機械認識における
特徴抽出と識別関数の研究



鶴岡信治

序

本論文は、筆者が名古屋大学大学院工学研究科博士課程（前期課程）電気系専攻に在学中および、三重大学工学部助手に在職中に行った、「手書き文字の機械認識における特徴抽出と識別関数の研究」をまとめたものである。

本文は次のような内容から構成されている。

第1章では、文字認識および文字認識アルゴリズムの研究についての概要と歴史を述べ、本論文の位置づけを行う。1.2では、文字認識研究の学問的・社会的意義について述べ、1.3で、文字認識についての歴史と現状を簡単に紹介する。1.4においては、現在広く使用されている代表的な文字認識の方法（パターン整合法、構造解析的手法）の概要について解説する。そして、1.5で、筆者の行った研究の目的及び手法、その特色について述べ、本論文の位置づけを行う。

第2章では、文字認識に適した細線化アルゴリズムについての詳細な議論を行い、新しい細線化アルゴリズムを提案する。そして、これを統計的な手書き文字の認識システム（ストローク構造解析法）に用いたときの有用性について述べる。この文字認識システムは、細線化アルゴリズムを用いて文字線分の中心線を取り出し、折線近似し、“楷書において一筆で書かれる部分（ストローク）”を取り出して、ストロークの特徴点と標準パターンを対応づけし、対応づけるのできた特徴点の位置により文字を識別する方法である。2.2

では、2次元デジタル図形についての基礎的な諸概念について解説し、2.3で、文字認識に適した細線化アルゴリズムを開発するために、“デジタル2次元空間における線分の太さが1ビットの図形（線図形）とは何か？”を議論し、線図形を図形要素の近傍状態により、集合論的に定義する。そして、その定義から導出される2値図形と濃淡図形に対する細線化アルゴリズムについて述べる。また、2.4では他の細線化アルゴリズムとの比較を行い、文字線分の縮退・消失が少ない割に、ヒゲ（認識をする場合に不必要な線分）が少なく、処理時間も速いことを述べ、2.5で、文字認識システムに使用したとき正読率が向上することを示す。

第3章では、人間の字種識別基準を用いた自由手書きカタカナ文字認識について述べる。この方法は、標準パターンを作成するのに、従来の手法のように学習文字を統計的に分析するのではなく、字種ごとに計算機で発生させた変形文字（折線近似パターン）を被験者に見せ、特徴空間内での各字種の判読限界値（字種識別基準）を直接計測する方法であり、その判読限界値を基準とした識別関数（心理的距離）により、識別を行う方法である。従来、人間の判読限界値を直接計測した研究はなく、もちろん、識別にそれらを使用した研究もない。以下、3.2では、実際に筆記された文字から主成分分析を用い、字形の変動を分析し、人工的に変形文字を発生させる方法を述べる。3.3で、その変形文字を被験者に見せて計測した人間の判読限界値（字種識別基準）を紹介し、統計的手法で求めた字種識別基準と人間の字種識別基準の差異について議論する。3.4で、人間の字種識別基準を用いた識別関数（心理的距離）について述べ、3.5で自由手書きカタカナ文字を対象とした認識実験を

通じて、統計的手法よりも本手法の方が良い正読率を示すことを述べる。

第4章では、手書き漢字認識のために開発した”加重方向指数ヒストグラム法”について述べる。この方法は、第2章、第3章で述べた方法とは異なり、文字線分の輪郭線に注目して特徴抽出する方法であり、位相幾何学的情報に鈍感であり、手書き漢字によく見られる、続けや切れのある文字でも安定して特徴量が抽出できる方法である。また、この特徴抽出法はボトムアップ処理を主体にしており、容易に並列処理ができ、文字の高速認識ができる特色がある。4.2では、その特徴抽出の方法について述べ、4.3で、この特徴量に適した新たに開発した識別関数（疑似ベイズ識別関数）について議論する。4.4では、手書き漢字・ひらがなに対する識別実験の詳細を紹介し、トップレベルの認識性能が実現できることを示す。

第5章では、前章までの議論のように1つのシステムで不特定多数の筆記者の文字を読み取る場合とは異なり、1台の文字認識装置を限定された筆記者が使用することを想定した議論を行う。すなわち、第4章で述べた加重方向指数ヒストグラム法を特定の筆記者が書いた文字に適応させる方法について述べる。ここでは、3種類の適応方法（更新型辞書、修正型辞書、混合型辞書）を使用して、自由手書きひらがな文字（46字種）を対象として、認識実験をし、その差異を比較、検討する。5.2では、筆記者の文字だけで学習した専用辞書と不特定多数の筆記者の文字で学習した汎用辞書による認識結果を比較し、専用辞書の方が効果的であることを示す。また、5.3で、3種類の筆記者の文字に適応する認識方法について

検討し、認識性能、使用記憶容量、計算量などを総合的に評価して、新しく提案した混合型辞書が最も有用であることを示す。

第6章では、字種数の多い手書き漢字認識で、第5章で述べた方法を有効に機能させるために、一つの提案をする。その提案とは、字種に依存しない筆記者の個性（書き癖）を少数の字種の文字より抽出する方法とその利用法に関する提案であり、小規模な実験ではその効果が実証できた。6.2で字種に依存しない筆記者の個性ベクトルの抽出方法と個人平均ベクトルの推定方法について提案し、加重方向指数ヒストグラム法を用いて、ひらがなの全字種に適用した結果を6.3に示し、筆記者の個性がある程度抽出できることを示す。また、6.4では推定した個人平均ベクトルを用いた自由手書きひらがな文字の認識実験の結果について述べ、汎用平均ベクトルを用いた場合より高い正読率が得られることを示す。

そして、第7章で、前章までの内容を要約すると共に、今後の課題について述べる。

目 次

序

第1章 まえがき	1
1. 1 はじめに	1
1. 2 文字認識研究の意義	2
1. 3 文字認識の歴史と現状	6
1. 4 文字認識方式の概要	10
1. 4. 1 文字認識過程	10
1. 4. 2 文字認識の対象	12
1. 4. 3 パターン整合法	17
1. 4. 4 構造解析的手法	21
1. 5 本研究の概要	24
1. 5. 1 本研究における文字認識手法	24
1. 5. 2 本研究の目的	26
1. 5. 3 使用した文字の品質	29
第2章 手書き文字認識に適した細線化アルゴリズムの開発	35
2. 1 はじめに	36
2. 2 図形の諸概念	38
2. 3 細線化アルゴリズム	42
2. 3. 1 線図形の定義	42
2. 3. 2 2値図形に対する細線化アルゴリズム	45
2. 3. 3 濃淡図形に対する細線化アルゴリズム	50

2. 3. 4	座標値表による高速化処理	52
2. 4	他の細線化法との比較	53
2. 4. 1	2値図形の細線化法	53
2. 4. 2	濃淡図形の細線化法	56
2. 5	文字認識システムへの応用	60
2. 6	むすび	63
2. A. 1	線図形の直観的定義と定義1との関係	64
2. A. 2	ストローク構造解析法の解説	67

第3章 人間の字種識別基準を用いた

	自由手書きカタカナ文字認識	75
3. 1	はじめに	76
3. 2	変形文字の発生	78
3. 2. 1	前処理	78
3. 2. 2	主特徴点と副特徴点	78
3. 2. 3	特徴ベクトル	79
3. 2. 4	変形文字の発生	80
3. 2. 5	主成分の意味づけ	81
3. 3	判読限界値	84
3. 3. 1	計測方法	84
3. 3. 2	計測結果	84
3. 4	識別方法	91
3. 4. 1	標準パターン	91
3. 4. 2	識別関数と棄却条件	91
3. 5	認識結果	96

3. 5. 1	判読限界値の修正	96
3. 5. 2	識別実験の結果	96
3. 5. 3	誤読・棄却の原因	97
3. 6	むすび	101

第4章 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	103
4. 1	はじめに	104
4. 2	加重方向指数ヒストグラム法	106
4. 2. 1	加重方向指数ヒストグラム法	107
4. 2. 2	3次元解釈	108
4. 2. 3	実現化	110
4. 3	識別関数	114
4. 3. 1	従来のベイズ識別関数の問題点	114
4. 3. 2	疑似ベイズ識別関数	115
4. 4	手書き漢字・ひらがなの識別実験	118
4. 4. 1	実験に使用したデータ	118
4. 4. 2	識別関数の比較	118
4. 4. 3	学習の効果	119
4. 4. 4	累積分類率	120
4. 4. 5	誤読文字	122
4. 4. 6	処理時間	123
4. 5	むすび	125
4. A. 1	輪郭追跡の方法について	127
4. A. 2	ベイズ識別関数の固有値・固有ベクトル	

表現について	130
第5章 筆記者に対して適応機能を持った自由手書き文字認識 ..	136
5. 1 はじめに	137
5. 2 専用辞書と汎用辞書	140
5. 2. 1 特徴抽出と識別関数	140
5. 2. 2 認識実験	142
5. 3 筆記者に適応する文字認識	146
5. 3. 1 2種類の適応型辞書	146
5. 3. 2 混合した適応型辞書	150
5. 3. 3 3種類の辞書の比較	153
5. 4 むすび	157
第6章 文字認識における筆記者の個性ベクトルの利用法	159
6. 1 はじめに	160
6. 2 筆記者の個性ベクトルの抽出と 個人平均ベクトルの推定	162
6. 2. 1 個性ベクトルの抽出手順	162
6. 2. 2 個人平均ベクトルの推定	164
6. 3 抽出した個性ベクトルの図表現と 推定した個人平均ベクトルの性質	166
6. 3. 1 抽出した個性ベクトルの図表現	166
6. 3. 2 推定した個人平均ベクトルの性質	168
6. 4 認識実験	172
6. 5 むすび	174

第7章	あとがき	176
-----	------	-----

謝辞	178
----	-----

文献	180
----	-----

付録	1. 変形文字パターンの例	193
	2. 字種ごとの主成分の図表現の例	199
	3. 加重方向指数ヒストグラムの例	204
	4. 主なプログラムの説明とソースリスト	209

第1章 まえがき

1. 1 はじめに

本章では、パターン認識といわれている研究分野の一部門である“文字認識”，すなわち“文字”を機械的に認識する方法に関する研究の概要を述べ、本論文の位置づけを行う。1. 2では、文字認識研究の学問的・社会的意義について述べ、1. 3で、印刷文字・手書き文字の機械認識の研究についての歴史を簡単に紹介する。1. 4においては、現在広く使用されている代表的な手書き文字認識のアルゴリズムの概要について解説する。そして、1. 5で、筆者の行った研究の目的及びその手法、特色、研究対象とした文字の品質について簡単に述べ、本論文の位置づけを行う。

1. 2 文字認識研究の意義

本論文で論じる“文字認識”とは、心理学や神経生理学⁽¹⁾、哲学の分野で論じられている認識論に関するものではなく、“工学的意味における文字認識”であり、知能機械に“特定の図形”を与え、その図形が文字であるとしたら、あらかじめ人間が与えておいた字種（数字で10字種、アルファベットで26字種、カタカナ・ひらがなで46字種、教育漢字で881字種、JIS第一水準の漢字で2965字種）のうち、どの字種に属するかを自動的に機械に判断させることをいう。

人間が文字を認識する場合には、文脈の理解と密接に結び付いており、1文字ずつでは曖昧なときにも文章の意味を理解しながら、判断している場合がよくあるが、本論文では、文脈情報を使用しないで、1文字ずつを切り離して機械的に認識する方法について議論する。すなわち、文字を“意味のない図形”として考え、図形の形状に注目して文字を認識する方法を研究する。

このような文字認識の研究を行うことについての意義として、次のような3つのことがあげられる。

(1) 人間の生体系の視覚情報処理過程の解明

人間が普段行っている情報処理を知的機械で実現する1つの方法として、まず、人間の生体系の情報処理機構を解明し、それを機械で実現できるように再構成するという手順が考えられる。し

かし、パターン認識問題のように、まだ人間の生体系の情報処理過程が解明されていない分野では、逆に知的機械で認識機構を模擬的に構成し、人間の生体系と同じ処理結果が得られるような認識機構が発見できれば、生体系の研究にその成果を生かし、相互に研究を発展させるという姿勢が考えられる。文字認識の研究は、このような立場から研究されており、人間と同程度に認識できるアルゴリズムを発見できれば、生体系で行われている視覚情報処理機構の解明に、大きく寄与することができる。

(2) 一般の画像・図形認識のための基礎的手法の開発

一般の画像を対象とした認識方法の研究が盛んになされているが、文字認識も”画像の中から抽象的概念を抽出したり、認識する”という点で共通しており、画像認識の一つの分野である。そのため、文字認識の分野で発見された概念や成果の多くは、一般の画像認識に対しても有用である場合が多く、その逆の場合も多い。現在、回路図、機械図面、建築図面、配管図、天気図、道路地図等の図面読み取りや、リモートセンシング画像、顕微鏡画像、医用画像、工業用検査画像等の画像処理が行われているが、これらの分野の研究と文字認識の研究は、将来にわたって、相互に刺激し合って発展していくと考えられる。

(3) 文字認識装置の実現

人間相互間の意志の伝達、知識の蓄積に、情報伝達媒体としての”文字”の果たしている役割は、余りにも大きい。”文字”がなかったならば、人類の文化、経済、産業、社会は、今日のように

高度に発達できなかったと思われる。また、今日計算機は社会の隅々まで浸透しつつあり、現代社会は高度情報化社会に向かいつつあるといわれている。このような社会の流れにおいて、計算機がより親しみのある機械となるには、“情報伝達の方法”が人間に対する方法と同様であることが望ましい。ここで、人間同士の間で広く使用されている情報交換の方法を並べると、①文字、②図形、③音声があり、そのなかで“文字”が、重要な情報を正確に伝える手段として使われている。

文字認識装置は、“文字”という媒体を通じて、人間と機械（計算機）がスムーズに情報交換を行うことを目的として開発された装置であり、この装置が身近なものとなれば、その社会的意義は大きい。例えば、事務処理で日常的に行われている“各種伝票の読み取り”、科学技術論文、特許情報などの“文献データ・ベース作成のための入力”、病院における“各種検査表、カルテの読み取り”、文字放送やビデオテックスなどの“情報の供給”など広い分野にわたって使用できる。

また、文字認識装置は情報圧縮機器であるとも考えられる。例えば、現在、光ファイリングシステムなどの情報蓄積方法は、ファクシミリと同様な形態が多く、文字を認識しないで、ドット・イメージのまま蓄積している。この方法では、記憶効率が悪く、例えば、1文字を 100×100 ドットで表し、2値化して1ドット＝1ビットとしても、10000ビット必要であり、認識してコード化したときの16ビットのJISコードなどと比べると、625倍の記憶容量が必要となる。

このようなことから、高精度な文字認識装置は広い社会分野で

望まれている装置である。

1. 3 文字認識の歴史と現状

”文字を機械に読み取らせる”試みは、人間が文字を考え出し、紙を発明して、”文字”を人間相互の主要な情報伝達媒体としてきたことから、自然な行動だと考えられる。

世界で初めて文字を読み取る研究を特許に届け出たのは、オーストリアのGustav Tauschekであり、1928年5月30日のことである⁽²⁾。⁽³⁾。彼は、証券や手形などに書かれた印刷数字を機械に読ませようという意図を明確に述べており、事務処理の合理化を当時既に考えていることが興味深い。その装置の原理図を図1. 1に示す。

紙面上の文字”1”を光で照らし、その反射光をレンズ系を通して倒立した像を受光器で受光する。レン

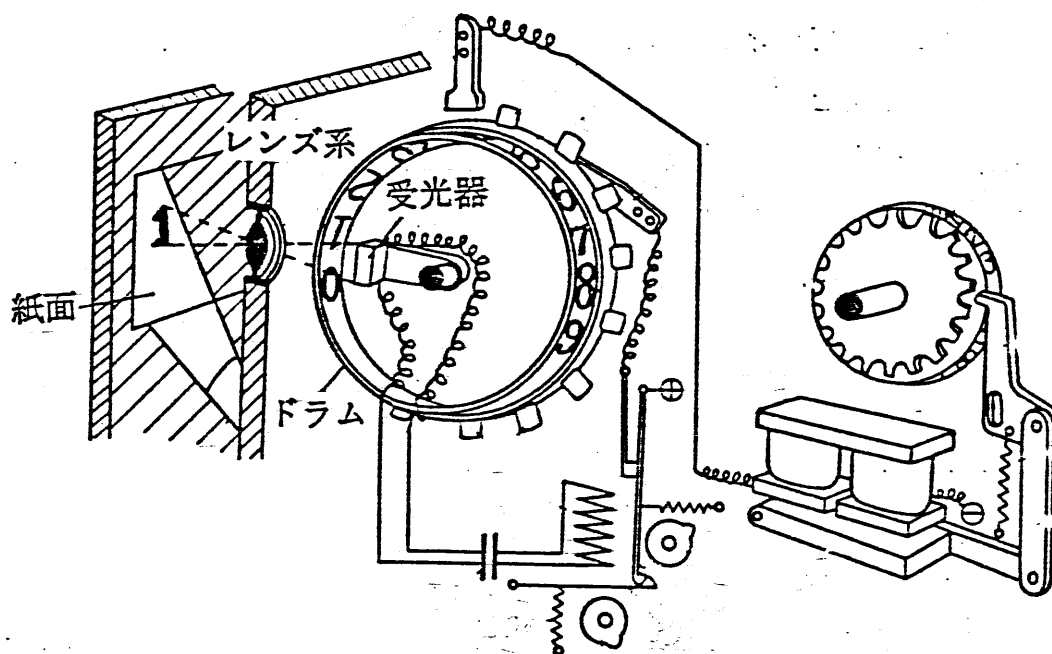


図1. 1 世界で初めて特許に出願されたOCR

ズと受光器の間には、文字の形が黒く書かれた透明な回転するドラムがあり、ドラム上の”1”の部分があると、文字の黒い部分で光が遮られるため、受光器へは光は届かなくなる。したがって、電流は流れなくなり、そのときのドラムの回転位置から、入力文字が”1”であったことが読み取れる。この認識方法は、現在、パターン整合（テンプレートマッチング）法とよばれている方法のもっとも原始的な文字認識の方法であり、機械光学的に実現した装置であった。

その後、1929年に米国のA T T社が機械光学式に電報文を読ませるために、数字以外に英字も読める特許を出願している。この方式は字種間の字形の互いに異なる点をあらかじめ装置の設計者が指定しておき、その点の濃度値の差異により字種を識別する方法であり、定点サンプリング法と呼ばれているパターン整合法の1つの方法である。

その後、米国のG E社なども1933年に文字認識装置の特許にしているが、本格的に文字認識の研究が始まったのは、デジタル計算機が出現してからである。それは、文字認識方式の開発には多くの試行錯誤が必要であり、膨大な文字データによるアルゴリズムの検証が研究を評価するために必要となるからである。

1955年には、ファーリントン社から光学式文字認識装置（Optical Character Reader：以下略して、OCR）が発表され、1957年にはイギリスのE R A社からもOCRが発表された。また同年G E社から磁気インクを用いた特殊な字体の文字を読む磁気インク文字読み取り装置（Magnetic Ink Character Reader：以下略して、MICR）が発表され、アメリカ銀行に実用機として納入され、小

切手などの数字を読み取るために用いられ、以後各社で活発な研究活動がなされた。

我が国では、1959年に電気試験所で初めてOCRの試みが、定点サンプリング法でなされ、多くの研究所・大学で研究がなされた。

1966年には、郵政省の郵便番号読み取り用手書き数字OCRを東京芝浦電気が作製し、不特定多数の人の書いた文字を扱う初めての実用的な大規模な応用であり、多くの人がOCRの存在を知り、後に日本電気も別の認識方法を用いて参加した。

1971年には、飯島ら⁽⁴⁾が印刷英数字を対象として、複合類似度法によるOCR (ASPECT/71) を試作し、高い評価を得た。

また、1971年より、10年間にわたって通産省工業技術院の大型プロジェクト「パターン情報処理システム」が行われ、我が国のパターン認識関係の研究レベルを大きく向上させた⁽⁵⁾。その結果、1970年代の後半で印刷英数字・カタカナ、手書き数字のOCRは、ほとんど完成し、1970年代末からはもう一段難しい研究対象である印刷漢字、手書きカタカナ文字、漢字認識の基礎研究が活発化してきた。

一方、外国の文字認識研究は印刷英数字の研究を終えると、外国では原稿用紙に文字を書く習慣はなく、文字をつづけて書くために、認識対象としては複雑すぎ、大きな興味の対象を失った。そのため、研究者の研究対象は署名の照合や一般の画像理解、コンピュータ・グラフィクス、自然言語処理、知識工学などに移行しており、文字認識の研究は我が国が世界の指導的研究をするようになった。

そして、現在では、印刷文字については、多くのメーカーから、

英数字カナ記号を読み取り対象とした製品が出荷されており⁽⁶⁾、金融機関での伝票処理や大学等での図書の管理システムなどに使用されている。また、印刷漢字を読み取り対象としたものも高価ではあるが、実用段階にきている。

一方、手書き文字では、郵便番号の読み取り装置のほかに、英数字カナ記号を対象としたOCRが、多くのメーカーから製品が出されているが、漢字まで含めないと利用範囲が制限されるため、現在のところまだ余り普及していない。しかし、現在各メーカーとも漢字までを認識対象とした装置を試作または検討しており、試験段階であるので、ハードウェアの価格が下がれば、一斉に市場に出てくるものと思われる。そして、OCRはキーボードに次ぐ計算機の標準的な入力手法の一つとなるという予測も、近い将来実現すると思われる。

1. 4 文字認識方式の概要

まず，一般的な文字認識過程の図式を示し，次に文字認識の対象について説明を行う．そして，本論文で議論しているオフライン認識方式の代表的な方法（パターン整合法，構造解析的手法）について簡単な説明を行う．

1. 4. 1 文字認識過程

一般に言われているパターン認識のうち，認識対象を文字に限定したパターン認識が”文字認識”であり，その認識過程の概要は一般の画像，音声を対象とするものとほぼ同じである．その処理の概略を図1. 2に示す．

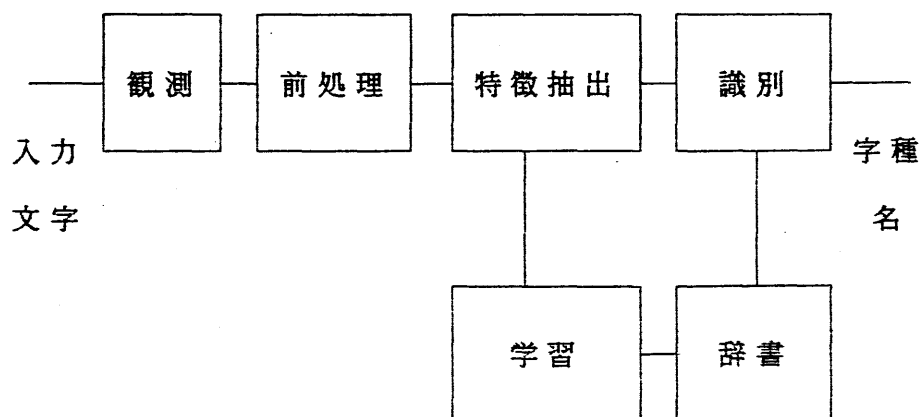


図1. 2 パターン認識過程の概略

(1) 観測

文字認識システムに与えられた入力文字はなんらかの手段で観測されて、電気信号に変換され、空間的な標本化と濃度値の量子化を行う。この過程を”観測”といい、オフラインの認識では、通常鉛筆、シャープペンシル、サインペンなどの筆記用具で紙の上に書かれた文字図形をイメージスキャナーなどの光電変換装置により、1文字当り 60×60 ドット程度（手書き漢字では 120×120 ドット程度）の分解能で電気信号にしている。またオンライン認識では、タブレット（筆記面）とスタイラス（ペン）によって筆点の x ・ y 座標の時系列を電気信号にしている。

(2) 前処理

電気信号にされた入力文字を画像の形式に変換し、観測過程（観測系に起因するもの、用紙に起因するもの）、文字の生成過程（印刷行程、筆記過程）に混在した様々な雑音、歪などを取り除く処理を”前処理”という。前処理の理想的なものは、筆記者の概念パターンを正確に抽出することであるが、この処理では、処理が複雑となることからトップダウン的な手法（概念パターンが明確となっており、それに沿って処理する方法）は使えず、パターン認識問題と考えた場合には、かなり難しい問題となる。具体的な処理としては、平均値フィルターによる濃度値の平滑化、しきい値による濃度値の2値化、幾何学的回転の補正、文字図形の位置・大きさの正規化、文字線分の太さの正規化（細線化処理など）などがある。

(3) 特徴抽出

前処理された文字図形から識別に使用する特徴ベクトル（特徴量ともいう）を抽出することを”特徴抽出”といい、特徴抽出の方法

の違いにより、様々な認識方法が提案されている。何を特徴量にするかは、研究者の経験や直観などにより、ヒューリスティック（発見的）に決められており、認識率の高いもの、次元数の少ないもの、抽出するのに計算時間の少ないもの、字種内の字形の変形が吸収できるものなどが、良いとされている。具体的な特徴ベクトルとしては、文字線分の画像上の2次元的な広がり状況、線図形における特徴点の位置関係、輪郭点画素の方向指数のヒストグラム、文字画素のx、y軸上などの周辺分布、文字画素の方向成分の2次元的分布状況等がある。

（４）識別

入力文字の特徴ベクトルを使用して、あらかじめ辞書に登録してある各字種ごとの標準パターンとの距離を計算し、最小の距離となる標準パターンの字種を入力文字の字種とする過程を”識別”といい、辞書を作成する過程を”学習”と言っており、そのときに用いる文字データを”学習文字”という。また、識別する距離には、①字種の代表値を1つ（平均値など）だけ用いる方法と、②字種の代表値を複数個（平均値と分散など）用いる方法があり、特徴ベクトルの次元数、学習文字数により、計算時間、特徴ベクトルの性質により、使い分けている。

1. 4. 2 文字認識の対象

文字認識装置は読み取り過程の違いにより、大きく2つに分けられており、（１）紙の上に印刷されたまたは書かれた文字を読み取るオフライン文字認識装置（光学式文字認識装置－OCR: Optica

l Character Reader) と, (2) 筆記する過程の時間的情報が利用できるオンライン文字認識装置がある (図 1. 3).

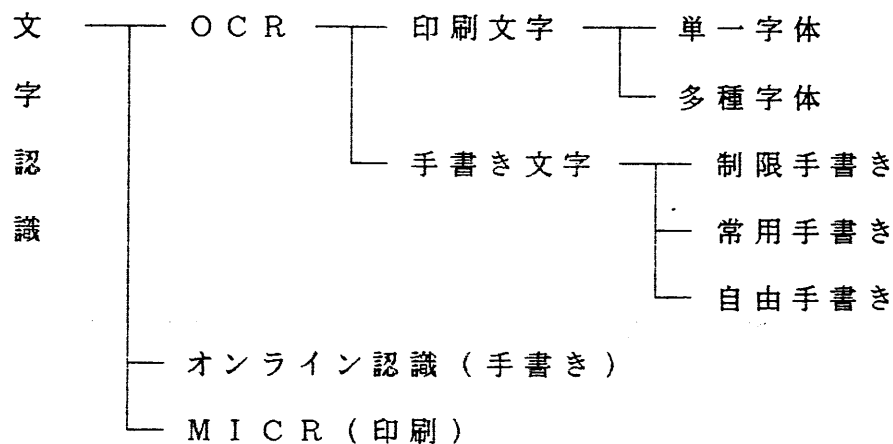


図 1. 3 文字認識の対象

(1) オフライン認識装置 (OCR)

オフライン認識装置の読み取り対象には, ①印刷文字 {単一字体 (シングルフォント), 多種字体 (マルチフォント, オムニフォント)}, と②手書き文字 {制限付き手書き, 常用手書き, 自由手書き} がある. この認識装置の特徴は, 筆記する時間的情報が認識処理においては利用できないので, 2次元の画像として処理をしなければならないことであり, 大きな情報量から重要な情報を抽出しなければならないことであり, 高度な情報処理技術を必要とする. しかし, この装置を利用して情報処理装置に情報を入力する場合には, "紙と筆記用具" さえあれば, 情報を記録することができるので, 計算機の前でなくても記録でき, 情報の発生する場所で従来人間が普通に行っている動作 (紙面上に筆記する) をすれば良いため, 使用者に

にとって心理的に受け入れやすいという利点がある。

次に手書き文字を分類すると、制限付き手書き文字、常用手書き文字、自由手書き文字があり、制限付き手書き文字は、文字線分の記入するガイドラインなどに沿って文字を記入する方法であり、数字の場合には考案されている（図1. 4）が⁽⁷⁾、字種が多くなるとうまく作れず、カタカナ、ひらがな、漢字では作られていない。

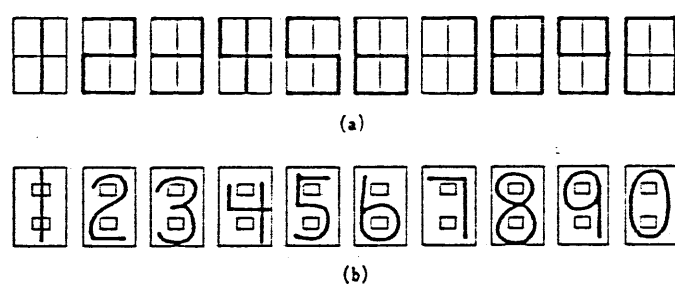


図1. 4 制限付き手書き文字

常用手書き文字とは、見本の字体（図1. 5）に沿って書かれた文字であり、識別しやすいような字種ごとの特徴があらかじめ説明された後で書かれたものであり、書き方規則が守られているならば、自由手書き文字に比べ、字体の多様性はかなり制限されており、類似字種の識別もしやすい。しかし、現実には書き方規則を守らない文字が書かれる場合が多く、装置としては自由手書き文字もかなり読み取るものとしなければならない。また、見本字体は英数字、カナ、記号までは作られているが、漢字では困難なので、漢字の認識では普通自由手書き文字である。

自由手書き文字は、字体の制限はないが、書かれる用紙上の文字の筆記位置の指定の仕方により、①原稿用紙の枠のように記入枠内に筆記するもの（枠内自由手書き文字）、②普通のノートのように

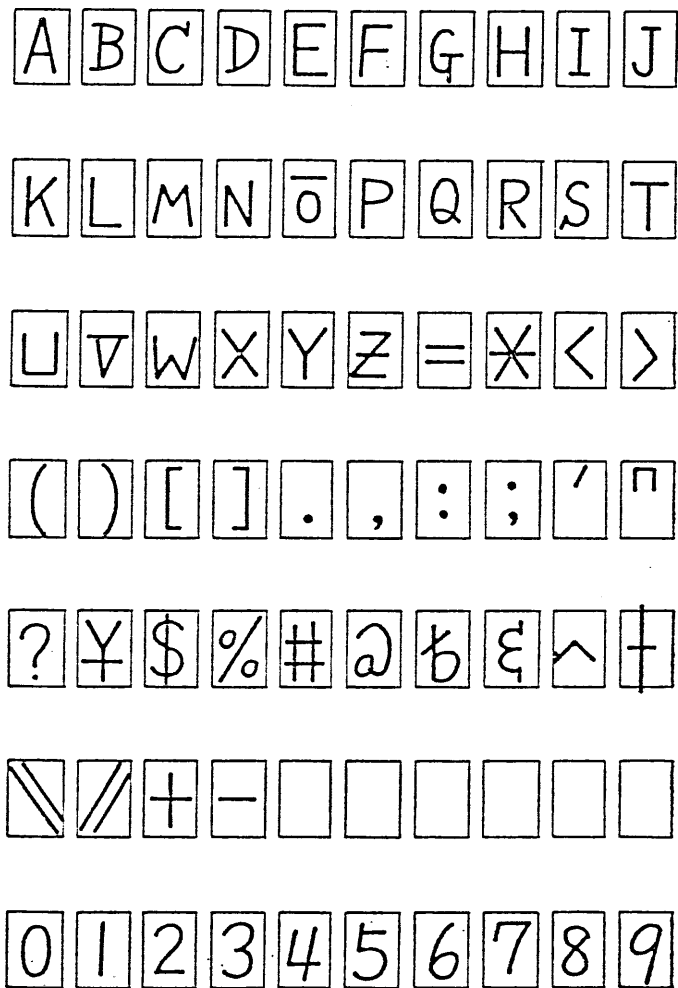


図 1. 5 手書き文字の見本字体（英数字記号）

横か縦の罫線に沿って筆記されるもの、③白紙の紙の上などのように全く罫線のないところに筆記されるものの3種類があり（図1.6）、文字の大きさ、縦横比、回転などの幾何学的変形や1文字分の画像領域の切り出し処理等に違いが生じる。

本論文では、自由手書き文字の中で最も基礎的な記入枠内に筆記された文字の認識に関する議論を主に行う。



図 1. 5 手書き文字の見本字体（カタカナ）

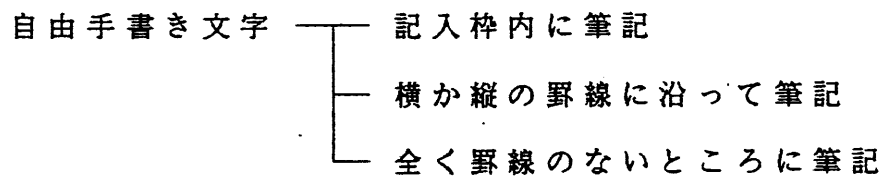


図 1. 6 手書き文字の分類

（２）オンライン認識装置とMICR

オンライン認識装置は筆記していく時間的過程が利用できるもので、筆の動かした状態（ストローク）を正確に取り出すことができ、画像処理の必要がなく、オフラインでの認識に比べ、少ない情報量を処理すればよい。そのため、計算量・記憶容量は少なく済み、小型の装置で実現することもできるが、装置を使用する場合には、必ず装置の前にいなければならない、キーボードと競合してしまい、入

力速度や疲労感の点から現在まだあまり普及していない。

M I C R (Magnetic Ink Character Reader:磁気インク文字読み取り装置)は磁気を帯びたインクで印刷された文字を読み取る装置であり、容易に改ざんできないため、銀行などの金融機関で手形や小切手などに使用されたが、特殊なインクを使用しているため、その用途は限られたものとなっている。

1. 4. 3 パターン整合法

パターン整合法(相関法, テンプレート・マッチング法ともいう)は、文字認識方法としてはもっとも基本的で自然なものであり、Gustav Tauschekの世界で始めての特許はこの方式であった。この方式は文字パターンの2次元的な濃度値をそのまま特徴量にしており、アナログ的な発想に基づく認識方式である。

(1) 単純類似度法

単純類似度法における文字の特徴ベクトル $f = (f_1, f_2, \dots, f_n)$ は、観測された図形を粗いメッシュで区切り、それぞれのメッシュの濃度値を特徴ベクトルの要素 $(f_i, i=1, \dots, n)$ としたものである(図1. 7)。この方法では、特徴抽出は極めて簡単であり、粗いメッシュの濃度値を求めることとなる。

次に類似度であるが、字種1の標準パターンの特徴ベクトルを g^1 とすると f とこの g^1 の単純類似度 $S_0(f, g^1)$ は、次式のよう

に定義されている。

$$S_0(f, g') = \frac{(f, g')}{\|f\| \|g'\|} \quad (1.1)$$

ここで、 $(,)$ は内積を示しており、 S_0 は -1 から 1 までの実数値をとる。また、 S_0 が 1 のときには f と g' は全く同じパターンであることを示しており、 -1 のときは白黒反転している図形である

f 1	f 2	f 3	f 4	f 5
f 6	f 7	f 8	f 9	f 10
f 11	f 12	f 13	f 14	f 15
f 16	f 17	f 18	f 19	f 20
f 21	f 22	f 23	f 24	f 25
f 26	f 27	f 28	f 29	f 30

図 1. 7 パターン整合法の特徴量 ($n=30$)

場合である。この類似度を準備されている全字種について求め、もっとも大きな類似度を示す字種 1^* を入力文字の字種名とする (式 (1. 2))。

$$S_0(f, g^{1*}) = \max_l \{ S_0(f, g^l) \} \quad (1.2)$$

この方法を実際の印刷文字などの認識に用いると、観測範囲内のわずかな文字線分の位置ずれや、かすれ、にじみなどによって正読率の低下が起こる。このような欠点をなくすために開発されたものが、次に述べる複合類似度法である。

(2) 複合類似度法

単純類似度法における字種の代表値は1つの特徴ベクトルであったが、代表値を複数個のベクトルにした類似度の1つに”複合類似度”があり、実用OCR(東芝ASPECT/71⁽⁴⁾)で使用されており、印刷文字認識に大きな効果を発揮している。また、他の手法と組み合わせて、印刷漢字⁽⁹⁾や手書き文字⁽¹⁰⁾の認識装置にも用いられており、手書きひらがな文字において、常用手書きひらがな文字データベース(ETL7)で98.1%の正読率が得られている^{(11), (12)}。

この方式について以下に説明する。

各字種のM個の代表パターン g_m^1 ($m=1, 2, \dots, M$)を用意する。入力文字 f と代表パターン g_m^1 の全体との単純類似度の2乗平均は

$$S_{1^1}(f) = \frac{1}{M} \sum_{m=1}^M S_0^2(f, g_m^1) \quad (1.3)$$

で表される。これは複数の代表パターンを使っているので、1つだ

けしか代表パターンを用いない場合に比べ、いろいろな変形が吸収できる可能性がある。この式を文字通り計算すると、 n 次元ベクトルの内積計算が M 回必要となる。しかし、文字の変形は多次元空間中の特定の部分空間にかたまっていると考えられるので、その特定の部分空間においてだけ類似度を計算するほうが、計算量は少なく認識性能はそれほど変化しない。この方法が”複合類似度法”といわれている方法である。次に、その計算式を示す。まず、式(1.3)の $S_{i'}$ は次のように変形できる。

$$S_{i'}(f) = \frac{1}{M} \sum_{m=1}^M \left\{ \frac{(f, g_{m'})}{\|f\| \|g_{m'}\|} \right\}^2$$

ここで、 $\|f\| = \|g_{m'}\| = 1$ とすると、

$$S_{i'}(f) = \frac{1}{M} \sum_{m=1}^M (f, g_{m'})^2 \quad (1.4)$$

となり、 $g_{m'}$ (n 次元ベクトル)の自己相関行列 K' ($n \times n$ 行列)の固有値 $\lambda_{i'}$ 、固有ベクトル $\psi_{i'}$ ($i=1, 2, \dots, n$)を用いて書き直すと

$$K' = \frac{1}{M} \sum_{m=1}^M g_{m'} \{g_{m'}\}^t \quad (1.5)$$

$$S_{i'}(f) = \frac{1}{M} \sum_{m=1}^M f^t g_{m'} \{g_{m'}\}^t f$$

$$\begin{aligned}
&= f^t - \sum_{m=1}^M g_m^t \{ g_m^t \}^t f \\
&= \sum_{i=1}^n \lambda_i^t (f, \psi_i^t)^2 \quad (1.6)
\end{aligned}$$

となる。この式において λ_i^t の小さな項を無視すると、第 n' ($< n$) 項までの和となり、

$$S_{i^t}(f) = \sum_{i=1}^{n'} \lambda_i^t (f, \psi_i^t)^2 \quad (1.7)$$

を得る。この式が実際の認識には使用され、印刷文字認識では、 $n'=3$ 程度で十分とされており、印刷文字の微少変形モデルから、 $\lambda_1 = \lambda_2 = \lambda_3$ であることが理論的に導かれている⁽¹³⁾。

また、ここで述べた多次元空間の情報を特定の部分空間で代表する考え方は、パターン認識理論の分野では、部分空間法⁽¹⁴⁾・⁽¹⁵⁾といわれており、統計学における主成分分析⁽¹⁶⁾とほぼ同じものである。

1. 4. 4 構造解析的手法

構造解析的手法とは、文字の特徴抽出をパターン整合法のように、濃度値の2次元分布としないで、もう少し自由な発想のもとに行ったものであり、文字パターンをいくつかの構成要素に分解（構造解析）し、識別するのに都合がよい特徴量を抽出して認識を行う方

法である。ここで、何を特徴量に選ぶかは、研究者の経験やセンスによるところが大きく、構造解析的手法を大きく分類すると、①文字線分の中心線に注目する方法と、②文字線分の輪郭線に注目する方法、③文字線分の書かれていない背景の解析を行う方法、④文字線分を構成する画素の方向性に注目する方法があり、現在盛んに研究されている。

① 文字線分の中心線に注目する方法

文字の中心線に注目する方法は、人間が文字を筆記するときの運筆状態を取り出そうとしたものであり、人間が文字を覚えている線分要素（ストローク）に分解するものであり、興味深い処理過程である。

ここに分類される認識方法としては、細線化法を用いた方法や線分を折線近似して処理する方法が入り、その代表的な方法に形式言語理論の文法規則を用いた認識方法^{(17)・(18)}、特徴点の統計的分布によりストロークを認識するストローク構造解析法⁽¹⁹⁾、ストローク密度特徴により認識する方法⁽²⁰⁾などがある。

② 文字線分の輪郭線に注目する方法

文字の輪郭線に注目する方法は、文字の中心線と輪郭線はほぼ1対1に対応していることに注目しており、中心線を取り出す処理が必要ないことから、最近よく用いられている方法である。この方法に分類される方法としては、凸閉包上の点（最外点）のヒストグラム、直線性、多角性、円形性などを用いた最外点リスト法⁽²¹⁾、多角形近似した輪郭線の対応づけをする弛緩法⁽²²⁾、粗いメッシュに

区切った領域内における輪郭点の方向指数のヒストグラムを用いた加重方向指数ヒストグラム法⁽²³⁾，等があり，手書き漢字でも，高い正読率が得られている。

③ 文字線分の背景部分の解析を行う方法

この方法は文字を形作る白い領域に注目した方法であり，大域的な特徴を重視した方法である。この方法の代表的なものに，Glucksmannの特性軌跡法⁽²⁴⁾があり，この方法を発展させた方法に，場の効果法⁽²⁵⁾，特徴集積法⁽²⁶⁾，反射法⁽²⁷⁾，セル特徴法⁽²⁸⁾，多元圧縮法⁽²⁹⁾などがあり，現在では手書き漢字にも適用されている。

④ 文字線分を構成する画素の方向性に注目する方法

上の2つの方法は文字線分の面情報を線情報に変換し，線情報の解析をしたものであるが，この方法は面情報を画素の方向性により分類し，方向性2次元パターンのマッチングにより認識を行うものである。この方法の代表的なものに，文字領域の指数化（4方向に分類）と背景の情報をあわせて画像を5種類の要素に分類して認識する方法⁽³⁰⁾，文字線分の方向セグメントを抽出する方法⁽³¹⁾，方向パターンとずらし類似度を用いた方法⁽³²⁾などがある。

特徴抽出の方法については，その他にもいろいろ提案されており，文献（33）に今までに開発されてきた種々の方法が体系的に紹介されている。また，学会誌においても文献（34），（35），（36）のような特集が発行されており，文字認識の方法について解説されている。

1. 5 本研究の概要

1. 5. 1 本研究における文字認識手法

本研究で論じられる認識方法は、前節で分類した構造解析的手法における①文字線分の中心線に注目する方法（ストローク構造解析法）と②文字線分の輪郭線に注目する方法（加重方向指数ヒストグラム法）であり、それらの方法の概略を以下に述べる。

（1）ストローク構造解析法⁽¹⁹⁾

この方法は、木村らによって開発されたものであり、細線化した文字パターンを下降的（トップダウン的）手法によりストローク抽出し、特徴点間の相対的位置や長さ、角度などの幾何学的情報を統計的に分析し、識別する方法であり、下降的に特徴抽出するところに特色がある。この方法は中心線を抽出する方法としては、最も高度な方法の一つであり、その基本構成を図1. 8に示す。

前処理部では、紙の上に書かれた入力文字を2次元のデジタル画像と考え、濃度値の2値化、ノイズ除去、位置・大きさの正規化、細線化を行い、折線近似パターンを得る。次に、特徴抽出部で、特徴点の候補点（文字ストロークの始点、終点、特徴的な屈折点）を字種ごとに用意した標準パターンの特徴点と動的計画法により下降式（トップダウン的）に対応づけを行い（図1. 9）、特徴点間の距離や角度、相対的なx y座標などの特徴量の計測を行う。そして、識別部で標準パターンとの統計的な距離値を計算し、字種の判定を

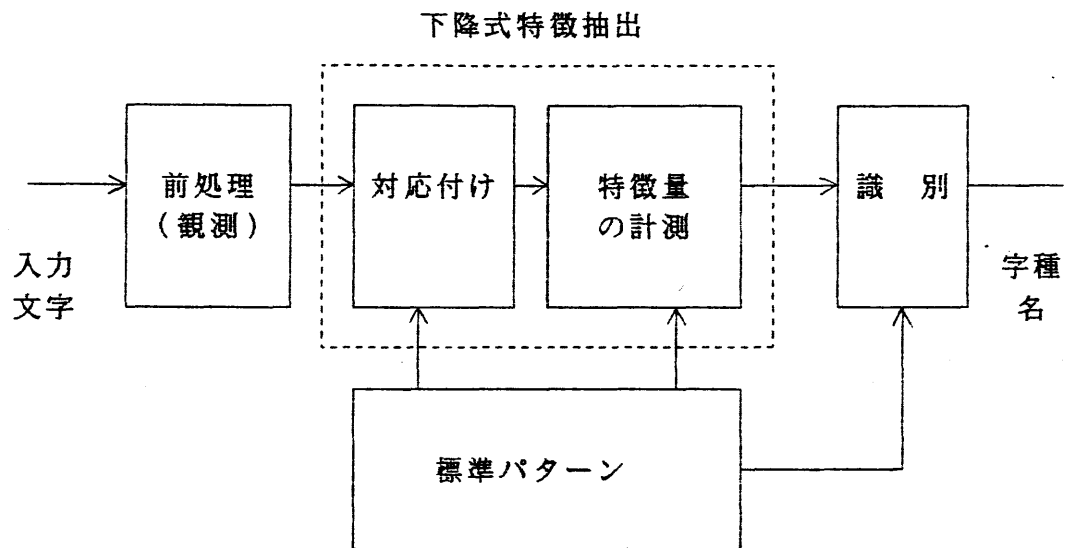
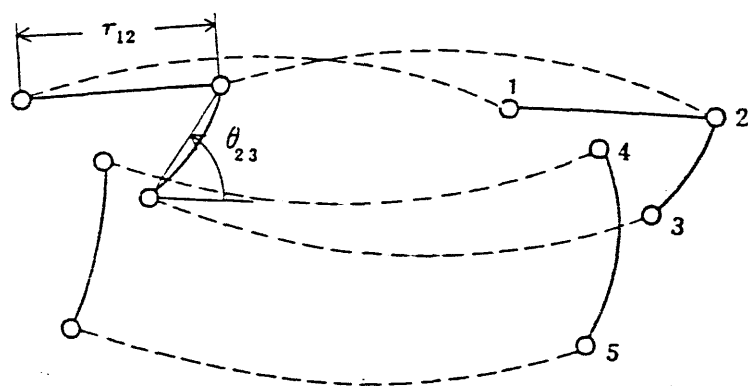


図 1. 8 ストローク構造解析法の概略



(a) 入力文字

(b) 標準パターン

図 1. 9 入力文字と標準パターンの対応付け

行う。

(2) 加重方向指数ヒストグラム法⁽²³⁾

この方法は、筆者らが手書き漢字の認識用に開発したものであり、(1)の方法のような高度な対応づけの方法を行っていないが、特徴は字種内で比較的安定して抽出できるように工夫してある。この方法の概略流れ図を図1. 10に示す。

この方法は文字線分の輪郭線に注目した方法であり、輪郭線上の画素の方向指数を求め、全画像を粗い小領域に分割し、各小領域における方向指数のヒストグラムを作成し、標準パターンとの統計的な距離値を計算して字種を識別する方法である。特徴量を求める計算量は(1)の方法に比べ数分の1以下で済み、字形の位相幾何学的な情報(線と線のつながり具合、穴の数、連結成分の数)に対しては鈍感な特徴量を採用している。また、識別関数には、従来の識別関数の問題点を解決した”疑似ベイズ識別関数”を使用している。この方法は手書き漢字の認識に有効であり、現時点でトップレベルの正読率を示している方法である。

1. 5. 2 本研究の目的

本研究の大きな目的は、”手書き文字”を人間と同じように読み取る機械を作成するために、高精度な文字認識のアルゴリズムを開発することであり、この研究を通じて、人間の生理的認識機構を数理的モデルにより解明することである。具体的な小目的を以下に示す。

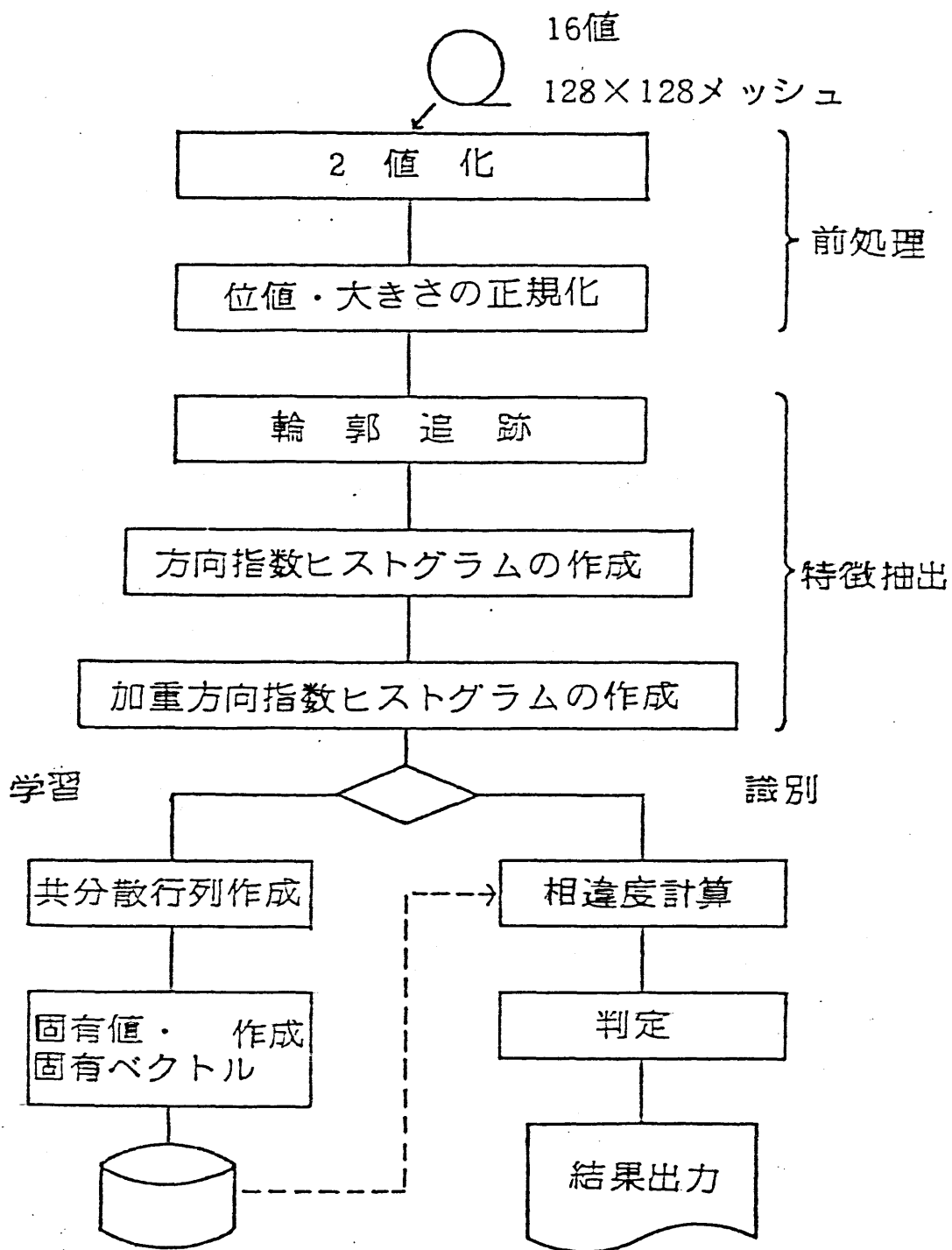


図 1. 10 加重方向指数ヒストグラム法の概要

(1) 線図形の基礎的検討と文字認識に適した細線化アルゴリズム
の開発 . . . (第 2 章)

ストローク構造解析法では、文字線分を細線化処理によって線幅 1 ビットの図形 (線図形) にしてから特徴抽出を行っており、細線化処理における線分の縮退や消失は不正確な特徴抽出の大きな原因となっていた。また、デジタル図形における ” 線図形 ” についての数理的な検討が不十分であった。そこで、筆者は ” 線図形の定義 ” を集合論的に検討し、文字認識システムに組み込んでよい性能となる細線化アルゴリズムを開発することを目的とした⁽³⁷⁾。

(2) 人間の字種識別基準の計測と識別関数の研究
. . . (第 3 章)

ほとんどの文字認識システムでは、学習文字を統計的に分析し、辞書を作成している。しかし、人間と同じ認識性能を示す文字認識システムを作成するには、人間の字種識別基準を辞書とするほうが良いと考えられる。そこで筆者は人間の字種識別基準を計測し、 ” 心理的距離値 ” の定義を行い、より高精度の認識システムを構成することを目的とした⁽³⁸⁾。

(3) 加重方向指数ヒストグラム法の開発とその評価
. . . (第 4 章)

ストローク構造解析法は下降式に対応づけを行う方法であり、漢字のように字種数が多いと特徴抽出するのに膨大な計算時間を要す。そこで、筆者は対応づけをしなくても字種内で安定に特徴量が抽出でき、人間の視神経の生理学的機構にヒントを得た特徴抽出方法と

従来の識別関数の問題点を解決する識別関数を考案し、手書き漢字を高精度で認識する方法を開発することを目的とした⁽²³⁾。

(4) 筆記者に適応する文字認識方法の確立

・・・(第5章, 第6章)

従来の多くの文字認識の研究ではOCRに不特定多数の筆記者の文字を読み取らせることを目的として、研究されてきたが、これはOCRが高価なものでありパーソナル機器として使用することは無理だという考え方を背景にしていると考えられる。しかし、近年の半導体技術の進歩によりハードウェアの価格は、急激に低下してきている。そこで、数年先にはOCRはパーソナル機器となると予測される。このような利用形態となると、筆記者に特有の書き癖や続け・省略を含んだ文字でも読み取れる可能性も出てきて、使用者の筆記時における負担は少なくなる。このような背景から、筆記者に適応する文字認識の方法を確立することを研究目的にした⁽³⁹⁾。

1. 5. 3 使用した文字の品質

本研究に使用した文字データベースは通産省工業技術院の電子技術総合研究所作成のETL1(第2, 3章), ETL8(第4章), ETL4と聖徳学園女子短期大学の吉村らと一緒に作成した特定筆記者の文字データベース(第5, 6章)であり、それらの概要を表1. 1に示す。

これらの文字は、いずれも見本字体を示さずに、記入枠から”はみださず、ていねいに”といった注意の下に書かれた枠内自由手書

き文字であり、手書き文字OCRを早期に実現するためのものではなく、手書き文字の基礎的性質を調べたり、基礎的研究をするために収集されたものである。

特定筆記者の文字を収集したOCRシート（A4版）の例を図1. 1. 1に示す。筆記枠の大きさは、縦6mm、横5mmであり、見本文字は提示しないで、“いろは”の順に横に同一の字種を26文字ずつ、HBの鉛筆で筆記したものである。また、縦10mm、横10mmの記入枠内に書かれた手書き教育漢字（ETL8）の各筆記者の字形⁽⁴⁸⁾（160人分）の例（位置・大きさの正規化後）を図1. 1. 2に示す。

表 1. 1 文字データベースの概要

データベース名	分類名	文字種	字種当り 筆記者数
ETL 1	自由手書き	英数字, カタカナ	1 4 4 5
ETL 8	手書き教育 漢字	漢字, ひらがな	1 6 0
ETL 4	自由手書き	ひらがな	1 2 0
特定筆記者	自由手書き	ひらがな	2 0 (26 文字 / 人 ・ 字種)

愛愛愛愛愛愛愛愛愛愛愛愛
愛愛愛愛愛愛愛愛愛愛愛愛
愛愛愛愛愛愛愛愛愛愛愛愛
愛愛愛愛愛愛愛愛愛愛愛愛
愛愛愛愛愛愛愛愛愛愛愛愛
愛愛愛愛愛愛愛愛愛愛愛愛
愛愛愛愛愛愛愛愛愛愛愛愛
愛愛愛愛愛愛愛愛愛愛愛愛
愛愛愛愛愛愛愛愛愛愛愛愛
愛愛愛愛愛愛愛愛愛愛愛愛

惡惡惡惡惡惡惡惡惡惡惡惡
惡惡惡惡惡惡惡惡惡惡惡惡
惡惡惡惡惡惡惡惡惡惡惡惡
惡惡惡惡惡惡惡惡惡惡惡惡
惡惡惡惡惡惡惡惡惡惡惡惡
惡惡惡惡惡惡惡惡惡惡惡惡
惡惡惡惡惡惡惡惡惡惡惡惡
惡惡惡惡惡惡惡惡惡惡惡惡
惡惡惡惡惡惡惡惡惡惡惡惡
惡惡惡惡惡惡惡惡惡惡惡惡

圧圧圧圧圧圧圧圧圧圧圧圧
圧圧圧圧圧圧圧圧圧圧圧圧
圧圧圧圧圧圧圧圧圧圧圧圧
圧圧圧圧圧圧圧圧圧圧圧圧
圧圧圧圧圧圧圧圧圧圧圧圧
圧圧圧圧圧圧圧圧圧圧圧圧
圧圧圧圧圧圧圧圧圧圧圧圧
圧圧圧圧圧圧圧圧圧圧圧圧
圧圧圧圧圧圧圧圧圧圧圧圧
圧圧圧圧圧圧圧圧圧圧圧圧

図1. 12 多数の筆記者が書いた文字例（ETL8の字形）

第2章 手書き文字認識に適した細線化アルゴリズムの開発

あらまし

細線化法は文字認識、画像処理等の分野で重要な役割を果たしている処理の一つである。しかし、従来は理論的根拠に乏しい直観的なマスク論理に基づく方法が主流であり、線図形の定義が不明確であった。本章では、正方格子上のデジタル図形における線図形を、所定の近傍状態をもつ図形要素のみから構成された図形と定義し、そのような線図形を抽出する2値図形と多値図形の細線化法を提案する。前者は不要なヒゲ状ノイズ枝を押さえても、線分の縮退・消失を少なくすることができ、又、処理時間も短いことが判明した。又、本細線化法を組み込んで、自由手書きカタカナ文字でストローク構造解析法により、認識実験を行った結果、文字線分の縮退・消失による誤読・棄却はほとんどなくなることが確認された。又、後者（多値図形の細線化法）では交点・分岐点付近のひずみが少なくなり、文字認識に有効であることが確認できた。

2. 1 はじめに

電子計算機による図形処理の基本的手法の中で、線状図形の中心線を幅1ビットで求める手法は重要な処理であり、文字認識、医用画像処理、航空写真解析、3次元物体の認識などで有効に利用されている。この手法は、1967年にBlum⁽¹⁾らが、連続平面上の図形を中心線で表現する考え方を提案したことに始まり、Rosenfeldらによってデジタル平面上での図形の連結性とこの手法との関連が指摘された⁽²⁾⁽³⁾。また1969年にはHilditchが逐次形の8連結細線化法を提案し⁽⁴⁾、1973年には横井がデジタル図形の連結数、曲率係数を提案し、それを用いた逐次形の細線化法を発表した⁽⁵⁾。

更に、並列形の細線化法⁽⁶⁾、濃淡図形の細線化法^{(7)・(8)}も国内外で数多く発表されており、その比較研究も行われている^{(9)・(10)}。

しかし、従来の手法は理論面、応用面のいずれにおいても十分洗練されているとは考えられない。まず、理論面では、“デジタル平面上における線図形(1ビットの線幅の図形)とは何か”という基本的問題について正面から議論したものはなく、点の除去判定条件および論理マスクは図形の連結性保存を除いて、実験的に設定されてきた。次に、応用面では、特に文字認識の前処理として用いた場合、次の3つの問題点を指摘することができる。

- (1) 文字心線の縮退・消失が著しい。
- (2) 図形の境界線の微妙な変化により、ヒゲ状ノイズ枝(以下ヒゲと呼ぶ)が発生する。
- (3) 交点が2つの分離した分岐点となったり、分岐点付近におい

てへこみが生じたりする。

特に（１）の問題は対応づけを行う文字認識システムでは、誤読・棄却の最大の原因となっており、ヒゲが多少発生しても、線分の縮退の少ないアルゴリズムが要求されている。又、これらの問題点と並んで処理時間を短縮することも重要である。

本章では、以上の背景を考慮して、従来明確にされていなかったデジタル２値図形における線図形を、連結数と８近傍に存在する１要素の個数により条件づけられた図形要素の集合と定義し、その定義を満足する線図形が得られる２値図形と多値図形の細線化法を提案する。次に、他のアルゴリズムと比較を行い、本アルゴリズムが手書き文字認識において有効な性質を持つことを示す。

以下、２．２においてデジタル図形に関する基本的概念の説明を行った後、２．３で線図形の定義、２値図形と多値図形の細線化アルゴリズムを示し、２．４で他のアルゴリズムとの比較を行う。２．５では自由手書きカタカナ文字認識システムに本アルゴリズムを組み込んだ実験について述べ、その有効性を示す。

2. 2 図形の諸概念

本節では、以下で述べる正方格子上的デジタル図形の諸概念について説明し、その記法を明示する。

(1) 図形 (picture or image) : 本章では、整数の組 (i, j) で各要素 (点ともいう) (element or point) の位置を表現できるデジタル図形 (digitized picture) を対象としており、その濃度値 (grey level) が f_{ij} で与えられる図形を $F = \{ f_{ij} \}$ と表す。特に、図形の全要素の濃度値が 0 又は 1 である図形を 2 値図形 (binary picture) $B = \{ b_{ij} \}$ と呼び、濃度値が 1 である要素 (1 要素: 1-Pixel) で "図 (object)" を、濃度値が 0 である要素 (0 要素: 0-Pixel) で "地 (背景) (background)" を表すものとする。

(2) 近傍 (neighborhood) : 注目点 (i, j) を x とするとき、それに隣接する要素 x_k ($k=1, 2, \dots, 8$) を図 2. 1 のように定義し、 x の 4 近傍 (4-neighborhood) $N^{(4)}(x)$, 8 近傍 (8-neighborhood) $N^{(8)}(x)$ を以下のように定義する。

$$\begin{aligned} N^{(4)}(x) &= \{ x_1, x_3, x_5, x_7 \} \\ &= \{ x_k \mid k \in S_1 \} \end{aligned} \tag{2.1}$$

$$\begin{aligned} N^{(8)}(x) &= \{ x_1, x_2, \dots, x_8 \} \\ &= \{ x_k \mid k \in S \} \end{aligned} \tag{2.2}$$

但し, S_1, S はそれぞれ添字集合
 $\{1, 3, 5, 7\}, \{1, 2, \dots, 8\}$ である.

2つの画素が互いに4/8近傍に
 存在するとき, これらは互いに4/
 8隣接する(4/8-adjacent)という.

	$j-1$	j	$j+1$
$i-1$	x_4	x_3	x_2
i	x_5	x	x_1
$i+1$	x_6	x_7	x_8

図2. 1 要素 x とその
 近傍 x_k

(3) 連結性(connectivity): 2値図形中の同じ値を持つ任意の2
 つの要素 a, b に対して, 要素 a と要素 b を結ぶ4/8隣接
 する画素の系列(sequence) (各要素間(p_i と p_{i+1})がすべ
 て要素 a と同じ値を持っている) $p_0 (= a), p_1, p_2, \dots,$
 $p_{n-1}, p_n (= b)$ が存在するとき, 画素 a と b は4/
 8連結している(4/8-connected)という.

また, 互いに連結している画素の集合を1つのクラスにま
 とめると, いくつかの1要素と0要素のクラスができる. こ
 の各々のクラスを連結成分(connected component)という.

そして, 0要素の連結成分で図形の縁(frame)の要素と連結
 していない成分を穴(hole)という.

なお, 2値図形において連結成分を考える場合には, 1要
 素の連結成分を4/8連結で考えるとき, 0要素の連結成分
 についてはこれと逆の8/4連結で考えないと, 8連結の1
 要素で囲まれている0要素が外側の連結成分と8連結で連結
 している, あるいは4連結で連結していない1要素によって

0 要素の連結性が損なわれているというような連結性の矛盾 (paradox) が生じる。

(4) 連結数⁽⁵⁾ (connectivity number): 要素 x の 4 連結数 $Nc^{(4)}(x)$, 8 連結数 $Nc^{(8)}(x)$ を次のように定義する。

$$Nc^{(4)}(x) = \sum_{k \in s_1} b(x_k) (1 - b(x_{k+1}) b(x_{k+2})) \quad (2.3)$$

$$Nc^{(8)}(x) = \sum_{k \in s_1} \overline{b(x_k)} (1 - \overline{b(x_{k+1})} \overline{b(x_{k+2})}) \quad (2.4)$$

但し, $b(x_9) = b(x_1)$, $\overline{b(x)} = 1 - b(x)$

[性質] 2 値図形において連結数が 1 の要素を除去しても, 図形の連結性 (連結成分数, 穴の数) は変化しない。

連結数とは, その要素を除去した (0 要素にした) ときに, その要素が属していた 1 要素の連結成分がいくつに分解されるかを示しており, 例えば, 連結数 2 の要素を除去すると, その要素が属していた連結成分は 2 つの連結成分に分解する。

(5) 図形の処理形態 (processing type): 入力図形の全要素に対して, m 回目までの処理を施した図形を $F^{(m)} = \{f^{(m)}_{ij}\}$ で表し, 初期図形を $F^{(0)} = \{f^{(0)}_{ij}\}$ で表す。

要素 x について, m 回目の処理を行うとき, "並列形処理 (parallel operation)" では, $F^{(m-1)}$ の値を用いて条件の判定を行うが, "逐次形処理 (sequential operation)" では, 既走査部分では $F^{(m)}$ の値を, 未走査部分では $F^{(m-1)}$ の値を

用いて条件の判定を行う。

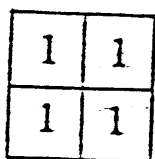
2. 3 細線化アルゴリズム

本節では、線幅1ビットの図形（以下、線図形(line figure)という）の定義とその定義に基づいた2値図形と多値図形に対する細線化アルゴリズムを提案する。

2. 3. 1 線図形の定義

線図形は本来2値図形に対して用いられる概念であるので、ここでは2値図形を対象として議論する。

4連結の線図形は、直観的には図2. 2のような 2×2 の1要素の塊を含まない図形であると考えられる〔直観的定義〕が、連結性を保ったまま線図形化できない図形（例えば、図2. 3）もあるので、本論文ではこのような図形も線図形と認め、線図形を以下のように定義して議論する。



（空白は1要素又は0要素）

図2. 2 2×2 の1要素の塊

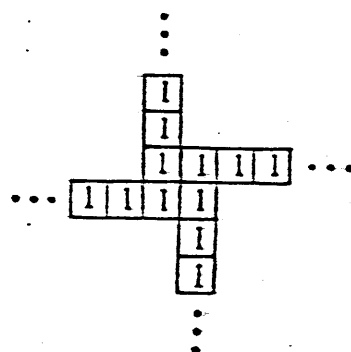


図2. 3 風車状図形

[定義 1] (4 連結の線図形) 図形に属するすべての 1 要素が次のどちらかの条件を満足しているときに, その図形を 4 連結の線図形であるという.

$$(1) \quad N c^{(4)}(x) \neq 1$$

$$(2) \quad S^{(8)}(x) \equiv \sum_{k \in s} b_{(x,k)} \leq 2$$

この定義は図 2. 3 のような図形を線図形として認めた定義であり, そのほかは前述した直観的定義とほぼ同一である (2. A. 1 参照).

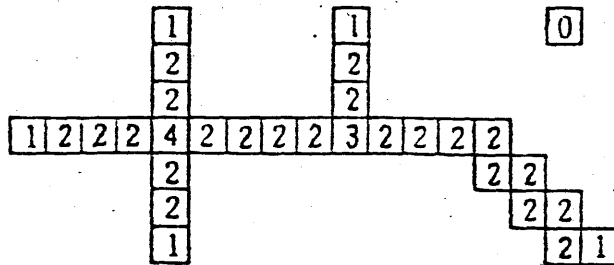
[定義 2] (8 連結の線図形) 図形に属するすべての 1 要素が次のどちらかの条件を満足しているときに, その図形を 8 連結の線図形であるという.

$$(1) \quad N c^{(8)}(x) \neq 1$$

$$(2) \quad S^{(8)}(x) \leq 1$$

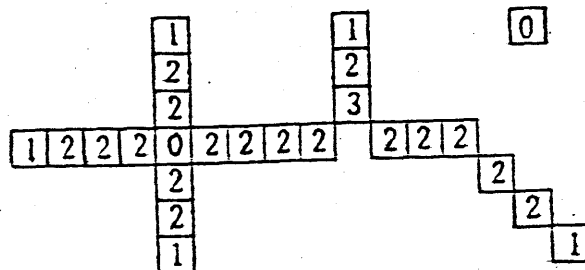
ここで定義 1, 定義 2 の条件 (1) は, その要素を除去すると図形の連結性が変化することを意味しており, 別のいいかたをすると, 境界線が 2 回以上通る点であることを示す条件⁽⁵⁾である. また, 条件 (2) は線図形の端点に対する条件である. それぞれの線図形の

例を図 2. 4, 図 2. 5 に示す.



(数字は連結数を示し、空白は 0 を示す。)

図 2. 4 4 連結の線図形の例



(数字は連結数を示し、空白は0を示す。)

図 2. 5 8 連結の線図形の例

2. 3. 2 2値図形に対する細線化アルゴリズム

① 逐次形細線化アルゴリズム⁽¹⁾

逐次形細線化アルゴリズムでは、走査による境界点除去処理の異方性を少なくするために、多数の除去条件を用いるのが普通であったが、ここでは、異方性のある除去判定マスクを導入して除去条件の整理を行う。

[アルゴリズム1] (逐次形細線化アルゴリズム: Sequential Thinning Algorithm for Binary Picture—STA)

処理回数 $m=1, 2, \dots$ について以下の処理を行い、処理図形が変化しなくなったとき、すなわち $B^{(m)} = B^{(m-1)}$ となったとき終了する。

図形を左上からラスタ走査し (図2. 6)、以下に示す点の除去条件 (1) ~ (3) のすべてを満足する要素を0とし、その他の要素の値は変化させない。

(1) 4連結の点の除去条件

(1) 境界点条件

$$S^{(4)}(x) \equiv \sum_{k \in s_1} b^{(m-1)}(x_k) < 4$$

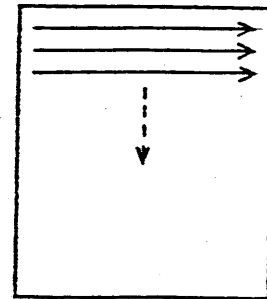


図2. 6 走査の方向

(2) 連結性保存条件

$$Nc^{(4)}(x) \equiv \sum_{k \in s_1} b^{(n)}(x_k) [1 - b^{(n)}(x_{k+1}) b^{(n)}(x_{k+2})]$$

$$= 1$$

但し, $n = \begin{cases} m & (j=2, 3, 4, 5 : \text{既走査点}) \\ m-1 & (j=1, 6, 7, 8 : \text{未走査点}) \end{cases}$

(3) 端点保存条件 (a)又は(b)を満たすこと.

$$(a) S^{(8)}(x) \equiv \sum_{k \in s} b^{(n)}(x_k) > 2$$

(b) 図 2. 7 (a) ~ (d) のいずれかの論理マスクを満たすこと.

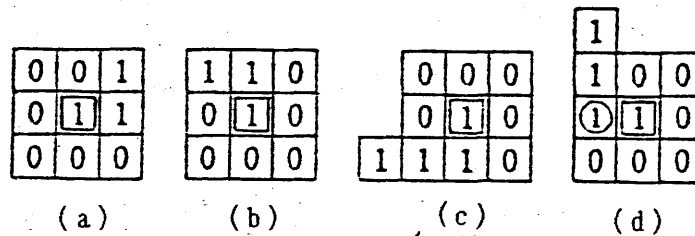


図 2. 7 除去判定マスク

(マスクと一致したときには注目画素を除去する.)

(d)では, 近傍画素①も除去する)

(2) 8 連結の点の除去条件

(1) 境界点条件

$$S^{(4)}(x) < 4$$

(2) 連結性保存条件

$$Nc^{(8)}(x) \equiv \sum_{k \in s_1} \overline{b^{(n)}(x_k)} [1 - \overline{b^{(n)}(x_{k+1})} \overline{b^{(n)}(x_{k+2})}] = 1$$

(3) 端点保存条件

(a)及び(b)を満たすこと.

(a) $S^{(8)}(x) > 1$

(b) 図 2. 8 (a), (b) の論理マスクを満たさない.

	0	0	0		0	0	0
	0	1	0		0	1	1
0	1	1	0		0	0	1

(a) (b)

図 2. 8 除去防止マスク

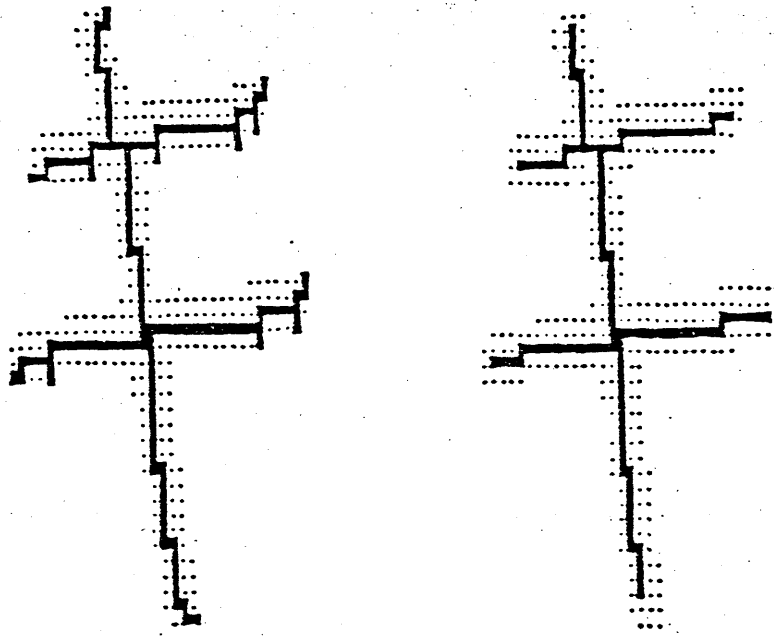
(マスクと一致したときには注目画素を除去しない)

ここで、条件(1)は線図形を元の図形の中心部に位置させるための条件であり、この処理では現走査を開始する前の状態を必要とする。これを1個の画像配列で行うには、除去した点の位置にその点が除去された走査回数を、例えば、負の値で記憶しておけばよい。

条件(2)は得られる図形の連結状態が元の図形の連結状態と一致するように、設定された条件であり、同時に定義1, 2の条件(1)の否定である。又、条件(3)は線分の縮退を防ぐための条件であり、定義1, 2の条件(2)の否定を基本とし、逐次形除去処理の異方性を防ぐために、方向異方性マスクを用いている。その効果の一例を図2. 9に示す。又、図2. 8のマスクは線幅2の線分の縮退防止用である。

② 並列形細線化アルゴリズム⁽¹²⁾

並列形細線化アルゴリズムは、図形の連結性を保存するために、すなわち、線幅2ビット以下の線分の消失を防ぐために、境界点(4近傍で0要素に隣接する1要素)を0要素の隣接する位置により、



(a) マスクを用いない場合 (b) マスクを用いた場合

図 2. 9 異方性マスクの効果

逐次型処理の 1 回分の処理を 4 つのサブサイクルに分けて除去判定を行う。すなわち、第 1, 2, 3, 4 サブサイクルの処理時には、それぞれ近傍 x_3, x_7, x_5, x_1 に 0 要素を持つ 1 要素のみについて点の除去判定を行い、条件 (1), (2), (3) をすべて満たす全 1 要素を 1 画面分の処理が終わったときに、同時に 0 とする。なお、1 要素の連結性を 4 (8) 連結で考えるときには、4 (8) 連結の点の除去条件を用いる。また、1 回目の除去判定は、すべて 1 - 1 回目の処理後の近傍の値を用いて行い、4 つのサブサイクルを通じて除去される要素がなくなるまで反復して処理を行う。

[アルゴリズム 2] (並列形細線化アルゴリズム: Parallel Thinning Algorithm for Binary Picture - PTA)

処理回数 $m=1, 2, \dots$ について以下の処理を行い, $B^{(m)} = B^{(m-1)}$ となったとき終了する.

図形を左上からラスタ走査し (図 2. 6), 以下に示す点の除去条件 (1) ~ (3) のすべてを満足する要素を 0 とし, その他の要素の値は変化させない.

(1) 4 連結の点の除去条件

(1) 境界点条件

$$S^{(4)}(x) \equiv \sum_{k \in s_1} b^{(m-1)}(x_k) < 4$$

(2) 連結性保存条件

$$\begin{aligned} Nc^{(4)}(x) &\equiv \sum_{k \in s_1} b^{(n)}(x_k) [1 - b^{(n)}(x_{k+1}) b^{(n)}(x_{k+2})] \\ &= 1 \end{aligned}$$

$$\text{但し, } n = \begin{cases} m & (j=2, 3, 4, 5 : \text{既走査点}) \\ m-1 & (j=1, 6, 7, 8 : \text{未走査点}) \end{cases}$$

$$(3) \text{端点保存条件} \quad S^{(8)}(x) \equiv \sum_{k \in s} b^{(n)}(x_k) > 2$$

(2) 8 連結の点の除去条件

(1) 境界点条件

$$S^{(4)}(x) < 4$$

(2) 連結性保存条件

$$Nc^{(8)}(x) \equiv \sum_{k \in s_1} \overline{b^{(n)}(x_k)} [1 - \overline{b^{(n)}(x_{k+1})} \overline{b^{(n)}(x_{k+2})}] \\ = 1$$

(3) 端点保存条件 $S^{(8)}(x) > 1$

ここで、条件(1)は線図形を元の図形の中心部に位置させるための条件であり、この処理では現走査を開始する前の状態を必要とする。これを1個の画像配列で行うには、除去した点の位置にその点が除去された走査回数を、例えば、負の値で記憶しておけばよい。

条件(2)は得られる図形の連結状態が元の図形の連結状態と一致するように、設定された条件であり、条件(3)は線分の端点を消失しないようにする条件である。また、条件(2)、(3)は、定義1、2の条件(1)、(2)の否定である。

2. 3. 3 濃淡図形に対する細線化アルゴリズム

2値図形の細線化法ではいずれの手法を用いても、分岐点・交点付近のひずみをなくすることが困難であるので、ここでは、図形の濃淡情報を用い、線図形が濃度値の高い所に位置するアルゴリズムを提案する。

[アルゴリズム3] (濃淡図形の逐次形細線化アルゴリズム: Se-

quential Thinning Algorithm for Graylevel Picture - STAG)

(STEP1) [準備] 入力濃淡図形 $F = \{f(x)\}$ からしきい値 T_1 により, 2値図形 $B = \{b(x)\}$ を作る.

この処理により”図”と”地(背景)”とを分離し, 以下の(STEP)では, 図の部分の細線化する.

(STEP2) [細線化処理] 次の処理(sub step)を $T=T_1, T_2, \dots, T_n$ (T_n は図形の最大濃度値)について順次行う.

(sub step) 次の処理を $m = 1, 2, \dots$ について行い, $B^{(m)} = B^{(m-1)}$ となったとき, 終了する.

図形を左上からラスタ走査し, $f(x) = T$ の要素を見つけ, それが前述した 2, 3, 2 の点の除去条件を満たすときに限り, $b^{(m)}(x) = 0$ とする.

(STEP3) [後処理] (STEP2)までの処理では, 原図形が濃度値に凹部を持つ場合, 線図形とならないこと(例えば, 図 2. 10)がある. そこで, 必要に応じて, 内点($S^{(8)}(x) = 8$ となる点), 極小点($N^{(8)}(x)$ に属するすべての x_k について $f(x_k) \geq f(x)$ が成立する点)について, 値を 0 にし, 2値図形の細線化法を通用すれば, 凹部に穴のあいた線図形が得られる.

3 3 3 3 3		1 1 1
3 2 2 2 3		1 1 1
3 2 2 2 3		1 1 1
3 2 2 2 3		1 1 1
3 3 3 3 3		

(a) 原図形 (b) (step2)

の結果

図 2. 10 線図形とならない例

このアルゴリズムでは、2値図形を入力図形とした場合に、2.
3. 2のアルゴリズムの処理結果と完全に一致する図形が得られる
ところに特色がある。

2. 3. 4 座標値表による高速化処理

文字図形のように“図”の部分の要素数の割合が小さい場合は、
(STEP2)の前に、しきい値以上の濃度値の点の座標値を濃度値別配
列に入れておくと、画面全部を走査する必要がなくなり、大幅な時
間短縮が可能となる。しかし、濃度値のレンジが広い場合は、しき
い値以上の点の座標を配列に入れておき、その配列から処理したい
濃度値を持つすべての点の座標値をそのつど別の配列に入れて、除
去判定する方法を用いたほうが、記憶場所をとらず有利である。

2. 4 他の細線化法との比較

2. 4. 1 2 値図形の細線化法

本節では、2. 3. 2で述べた細線化アルゴリズム (STA) を他の2 値図形の細線化アルゴリズムと比較する。

2 値図形の細線化法の比較を取り扱った文献 (9) によれば、4 連結の細線化法では横井の方法⁽⁵⁾ (YTA) と田村の方法⁽⁶⁾ (TTA) が良好な性質を持つものと考えられるので、本節ではこの2つの方法を比較の対象とする。YTA は連結性の保存が明確にされた逐次形のアルゴリズムであり、TTA は従来の種々の並列形のアルゴリズムを連結性の保存という観点から見直し、改良された並列形のアルゴリズムである。ここでの比較項目は、細線化法を文字認識に用いることを考え、文字線分の縮退・消失につながる線分の縮退点数、認識処理において余分な線分となるヒゲの数、処理時間、処理に必要な画像配列 (bit/pixel) の数、及び全境界点进行处理するのに必要な走査の回数 (サブサイクル数) とした (表 2. 1)。

(1) 線分の縮退

手書き文字データベースの文字線幅は4ビット程度であるため、人工的に作った線幅1～6ビットの十字形図形を用い (図 2. 11), 各細線化法について、入力図形の境界線から線図形の端点までの距離の上下左右4方向の和を調べた。その結果を図 2. 12 に示す。又、表 2. 1 の比較項目 (1) には、線幅1～6ビットについての和を示してあり、STA が最も良い結果となっている。

表 2. 1 4 連結細線化法の比較

細 線 化 法	Y T A	T T A	S T A
(1) 縮 退 点 数	26	66	22
(2) ヒ ゲ の 数	43	40	58
(3) 形 状 評 価	1118	2640	1276
(4) 処 理 形 態	逐次型	並列型	逐次型
(5) サブサイクル数	2	4	1
(6) 処理時間・[ms]	2056	(1857)*	769
(7) 画 像 配 列 数	3	3	2

* FACOM230-38S, FORTARN使用

+ 逐次型計算機での疑似実験

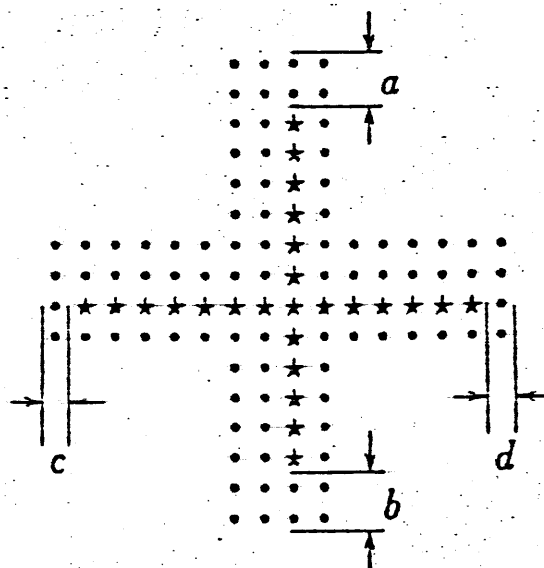


図 2. 1 1 線幅 4 ビットの
十字形図形

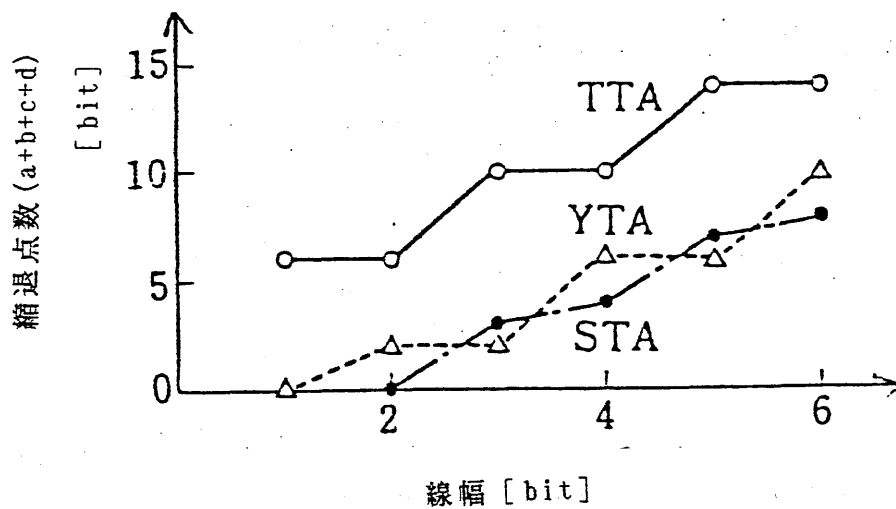


図2. 12 各細線化法における縮退点数

(2) ヒゲの発生及び処理時間

電子技術総合研究所・富士通作成の自由手書き文字データベース ETL1 のカタカナ文字（大きさ 60×60 メッシュ，各字種 5 文字計 230 文字）に対するヒゲの総数と 1 文字当りの平均処理時間を表 2. 1 に示す。ここで「ヒゲ」とは，正解の字種の標準パターンと 1 対 1 の対応が付かない線分のことであり，画像を観測した時の雑音，書き手のつき出し等により発生したものを含んでいる。ヒゲの数は TTA が最も少なく，処理時間は STA が非常に速く他の約 $1/3$ である。なお STA は 2. 3. 4 の高速化を行えば，更に速くなり，約 500 ms まで短縮できる。

(3) 形状評価

細線化法では，線分の縮退量とヒゲの発生数は共に小さいことが望ましいが両立しがたいものであるから，両者の積を評価基準とした。この結果 YTA と STA はほぼ同程度であり，TTA の約 2 分の 1 となった。

2. 4. 2 濃淡図形の細線化法

本節では、2. 3. 3で述べた細線化アルゴリズム (STAG) を他の濃淡図形用細線化アルゴリズムと比較する。他の細線化アルゴリズムとして、Hilditchの手法⁽⁴⁾ (以下、HIL)、文献 (10) の比較実験において良好な結果を示しているWPM⁽⁷⁾、横井らの手法⁽⁸⁾ (以下、YTAG)、及び2値図形の細線化法 (STA) を用いる。なおいずれの手法も後処理を行わない場合を示す。

(1) 抽出される線図形の基本的性質

まず上下方向に対しては無限に広がっている帯状図形について考察する。

(a) 濃度値に凹部点をもつ図形：図2. 13の図形を各細線化法で細線化すると抽出線的位置は図2. 14に示す矢印の通りとなり、STAG, YTAGでは、線図形とならず、凹部の内部 (YTAGは峰を含む) が抽出される。この性質により、文字認識では‘目つぶれ’領域もある程度抽出できる。

(b) 濃度値に平坦点を持つ図形：図2. 15の図形を細線化すると図2. 16のようになり、YTAGでは平坦点から最大濃度値までの全領域が抽出され、HIL, STAGでは最大濃度値の位置に抽出される。

(2) 手書き文字図形への適用

手書き文字データベース (16値) のカタカナ文字“エ”を各手法により細線化した結果を図2. 17に示す。

WPMでは、特に文字線分の端のところでノイズ枝が出やすく、連結性も4連結、8連結の混合形となり、分岐点付近でへこみが生じる

∴
 3 5 3 3 5 2
 3 5 3 3 5 2
 3 5 3 3 5 2
 3 5 3 3 5 2
 3 5 3 3 5 2
 3 5 3 3 5 2
 3 5 3 3 5 2
 ∴

(数字は濃度値,
 空白は0を示す)

図2. 13 濃度値に
 凹部を持
 つ帯状図
 形

∴
 3 6 5 4 3 3 3
 3 6 5 4 3 3 3
 3 6 5 4 3 3 3
 3 6 5 4 3 3 3
 3 6 5 4 3 3 3
 3 6 5 4 3 3 3
 3 6 5 4 3 3 3
 3 6 5 4 3 3 3
 ∴

図2. 15 濃度値に
 平坦点を
 持つ帯状
 図形

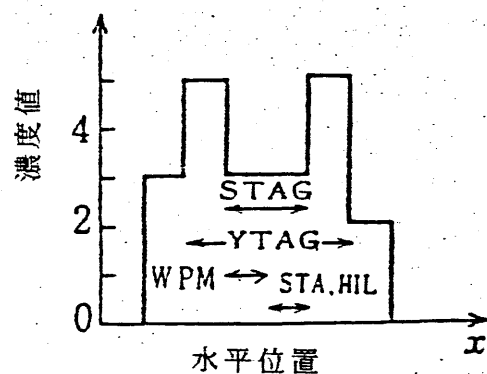


図2. 14 図2. 13の
 濃度値分布と
 抽出線の位置

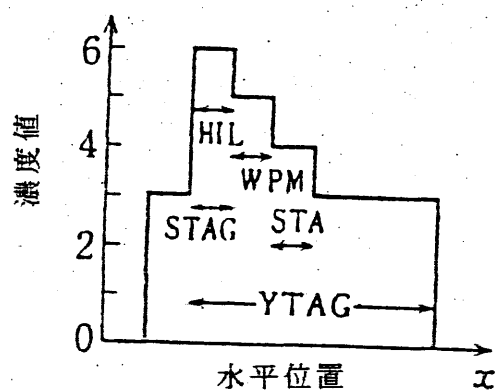
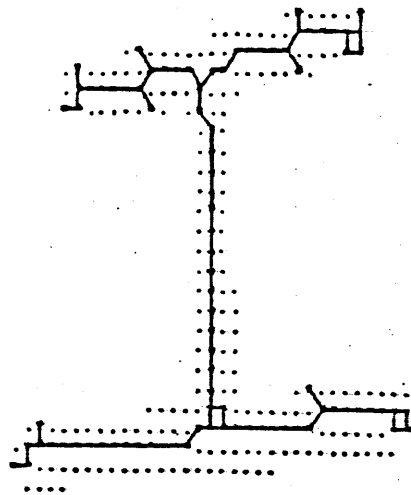
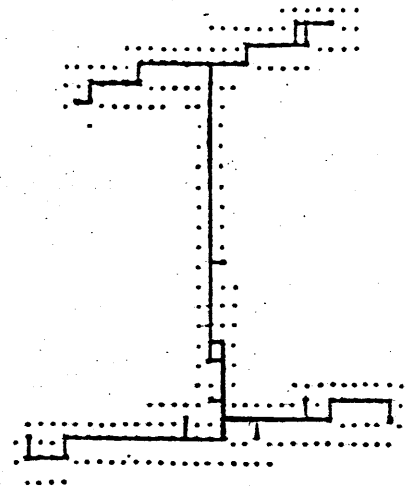


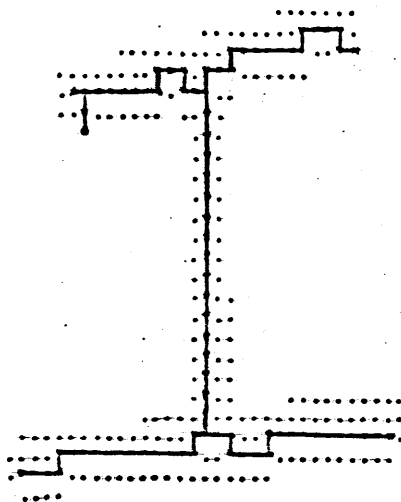
図2. 16 図2. 15の
 濃度値分布と
 抽出線の位置



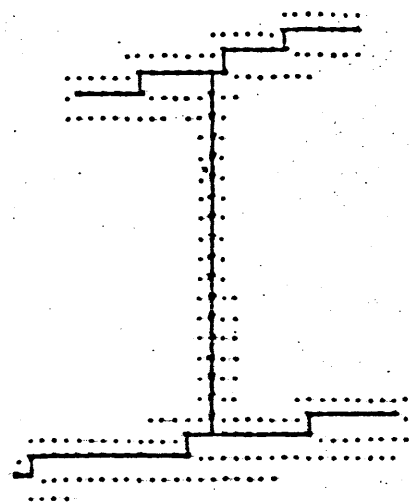
(a) WPM



(b) YTAG



(c) STA



(d) STAG

図 2. 17 手書き文字図形の実験例

Fig. 2. 17 Sample of handwritten character

場合がある (図 2. 17 (a)).

YTAGでは、平坦点が図形の所々で出現するために、線幅 2 となったり、ノイズ枝が発生したりする。しかし、分岐点付近のひずみはあまり生じない (図 2. 17 (b)).

STAでは、抽出される線の位置は境界線の位置により決まり、分岐点・交点付近のひずみは大きく、図 2. 17 (c) のようにへこみが生じる場合がある。しかし、ノイズ枝はWPM, YTAGより少ない。

STAGでは、分岐点・交点付近のひずみは、ほとんどなくなっている (図 2. 17 (d)). これは文字図形の濃度分布に筆記者の運筆状態が反映されているためである。又、WPM, YTAGに見られるノイズ枝は発生していないために、人間の考える中心線に非常に近い形状の線図形が得られる。

又、各手法の処理時間は、FACOM230-38Sを用いた場合、1454 ms (WPM), 1301 ms (YTAG), 758 ms (STA), 690 ms (STAG) である。STAGがSTAより速いのは、2. 3. 4 の高速化の手法を用いていることによる。

2. 5 文字認識システムへの応用

本節では、STAをストローク構造解析法⁽¹³⁾による手書きカタカナ文字認識システムに組み込んだ場合の認識実験について述べる。

このシステムは入力パターンを2値化、細線化、折線近似したうえ、5回以内の非決定的ストローク抽出を行い、特徴点間の角度と両特徴点間を結ぶ直線からの符号付最大偏位を特徴量とするマハラノビス距離を用いて字種判定する方式である（図2. 18, 2. A. 2参照）。

細線化操作にSTA, TTAを用いた場合の認識結果を表2. 2に示す。同表の認識率は共に電総研・富士通作成の自由手書き文字データベースETL1中の第6, 15, 18, 19, 20巻の計11, 500文字（未学習文字；各字種250文字）に対して、棄却率と誤読率を等しく設定したときの値であり、STAを用いた場合、TTAを用いた場合より正読率で1. 0%の上昇が認められる。

表2. 2 自由手書きカタカナ文字の認識結果

細 線 化 法	T T A	S T A
正 読 率 (%)	94. 6	95. 6
棄 却 率 (%)	2. 7	2. 2
誤 読 率 (%)	2. 7	2. 2

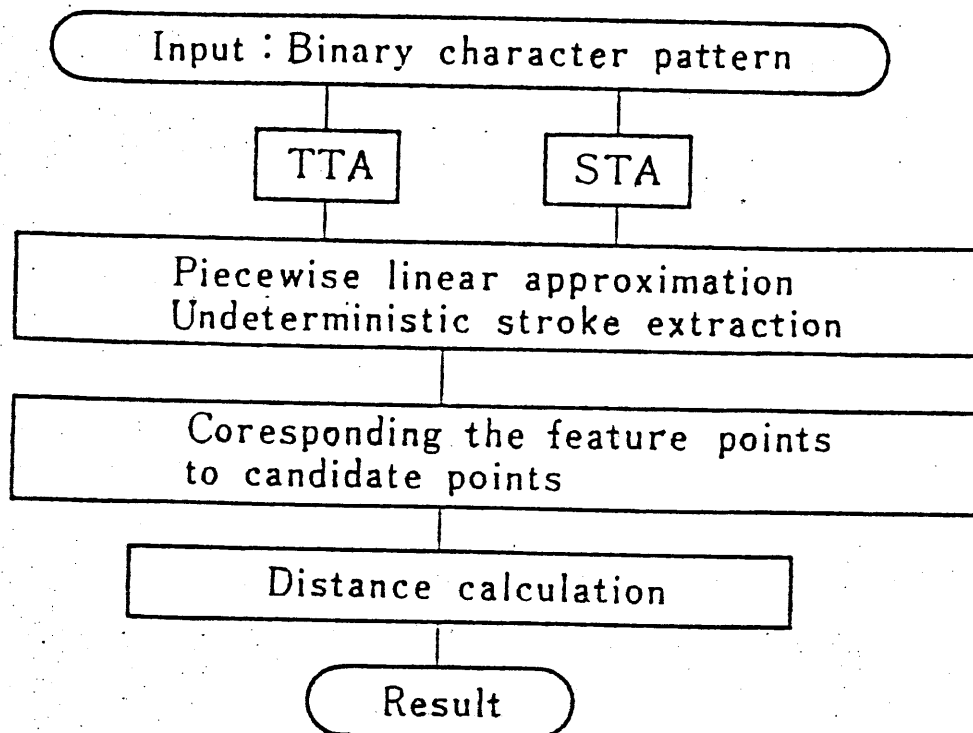


図 2. 18 文字認識実験の流れ図



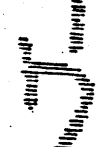


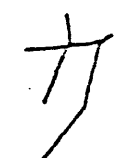
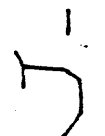



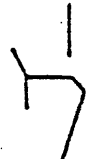
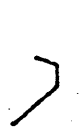
Original Pattern				
	(a)	(b)	(c)	(d)
TTA				
	'キ' → RJ	'+ ' → 'カ'	correct	correct
STA				
	correct	correct	'ウ' → RJ	'ノ' → RJ

図 2. 19 正読となった文字例 ((a), (b)) と
棄却となった文字例 ((c), (d))

これは図2. 19 (a), (b)に示すように細線化時の文字線分の縮退・消失が少なくなること起因するものがほとんどであり，新たに正読となった文字は約1.9%あった．又，新たに誤読・棄却となった文字には，書き手が不要な線分を書いたが，TTAでは縮退効果のために，それほど長い線分とはならなかった文字（図2. 19 (c)の第1ストローク），折点位置が偶発的に不適切に設定されたために棄却となった文字（図2. 19 (d)）などがあり，識別方法，折線近似の方法の改善により，更に本方法の有効性が示されることが考えられる．

更に，2.3.3で提案したSTAGを用いた場合には図2. 20のような文字が正読となり，分岐点付近のひずみの減少，折点位置の適正化という特徴が文字認識にとって有益であることが確認できた．






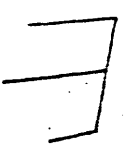






Binary Pattern				
	(a)	(b)	(c)	(d)
STA				
	'タ'→RJ	'ヲ'→'ヨ'	'ク'→'フ'	'ネ'→'ホ'
STAG				
	correct	correct	correct	correct

図2. 20 濃度値情報により改善された文字例

2. 6 む す び

本章では、まず従来明確に定義されていなかったデジタル図形における線図形を2つの近傍概念を用いて集合論的に定義し、その定義を満足する線図形が得られる2値図形と濃淡図形に対する細線化アルゴリズムを提案した。本アルゴリズムの特色は、異方性のある除去判定マスクを用いて走査方向による除去処理の異方性を少なくしていること、整理された除去条件により構成されていること、並列型のアルゴリズムを容易に構成できることなどである。又、他の2値図形の細線化法と比較した結果、本方式は線分の縮退が少なく、処理時間も少なく、総合的に評価して、有効な方法と考えられる。次に、濃度値情報を用いることにより、鉛筆等の筆記具を用いて描かれた文字図形では、交点・分岐点付近のひずみを少なくすることができ、折点位置が適切に設定されることを示した。

更に、自由手書きカタカナ文字の認識実験により、誤読・棄却の最も重大な原因であった文字線分の縮退・消失が本章の細線化アルゴリズムにより大幅に改善されることが分った。

今後の課題としては、細線化処理時に発生するヒゲの大局的情報による判別、及び変形交点の修復処理などが残されている。

なお、本章の実験には、名古屋大学工学部情報・自制計算機及び三重大学計算センターを利用した。

2. A. 1 線図形の直観的定義と定義1との関係

まず、デジタル2値図形における線図形の直観的定義の定式化を行う。

[直観的定義] 直観的に“線図形”とは、図形中に 2×2 の1要素の塊 (図2. 2) をひとつも持たない図形と考えられる。

2×2 の1要素の塊 (図2. 2) の個数 N_B は、式(2.A.1)で与えられる。

$$N_B = \sum_{x \in I \times J} b(x) \left\{ \sum_{k \in s_1} b(x_k) b(x_{k+1}) b(x_{k+2}) \right\} / 4 \quad (2.A.1)$$

ここで、 $I \times J = \{ (i, j) \mid 1 \leq i \leq M, 1 \leq j \leq N \}$ である。

2×2 の1要素の塊が図形中に存在しないことは、連結数の定義を用いて表すと次式となる。

$$\begin{aligned} N_B &= \sum_{x \in I \times J} b(x) \left\{ \sum_{k \in s_1} b(x_k) - N c^{(4)}(x) \right\} / 4 \\ &= 0 \end{aligned} \quad (2.A.2)$$

ここで、

$$\begin{aligned} \sum_{k \in s_1} b(x_k) - N c^{(4)}(x) &= \sum_{k \in s_1} b(x_k) b(x_{k+1}) b(x_{k+2}) \\ &\geq 0 \end{aligned} \quad (2.A.3)$$

なので、図形中のすべての1要素について式(2.A.4)

$$\sum_{k \in s_1} b(x_k) = N c^{(4)}(x) \quad (2.A.4)$$

が成立しなければならない。

次に、この定義と定義1との関係を説明する。

式(2.A.4)を満たす点の集合を L_1 、定義1を満たす点の集合を L_2 とし、各集合の $N c^{(4)}(x)$ と $S^{(4)}(x)$ のとりうる値について考察する。ここで L_2 は $S^{(8)}(x)$ で定義されているが、 L_1 と比較するために、8近傍は4近傍を含む関係を考慮して、 $S^{(4)}(x)$ を用いる。

L_1, L_2 のとりうる値は図2. A. 1の●, ○で示される。但し、×で示す組み合わせは、以下の理由によりありえない。

(1) 連結数の定義より、

$$\begin{aligned} S^{(4)}(x) - N c^{(4)}(x) &= \sum_{k \in s_1} b(x_k) b(x_{k+1}) b(x_{k+2}) \\ &\equiv C(x) \geq 0 \end{aligned}$$

なので、 $S^{(4)}(x) \geq N c^{(4)}(x)$. (2.A.5)

(2) $S^{(4)}(x) = 1$ のとき、 $C(x) = 0$ なので、

$$N c^{(4)}(x) = 1.$$

(3) $S^{(4)}(x) = 2$ のとき、 $C(x) \neq 2$ なので、

$$N c^{(4)}(x) \neq 0.$$

又、 $S^{(8)}(x) \leq 2$ より、 $C(x) = 0$ なので、

$$N c^{(4)}(x) \neq 1.$$

(4) $S^{(4)}(x) = 3$ のとき、 $C(x) \neq 3$ なので、

$$Nc^{(4)}(x) \neq 0.$$

したがって、L1とL2の差異は意外に少ないことが分かる。

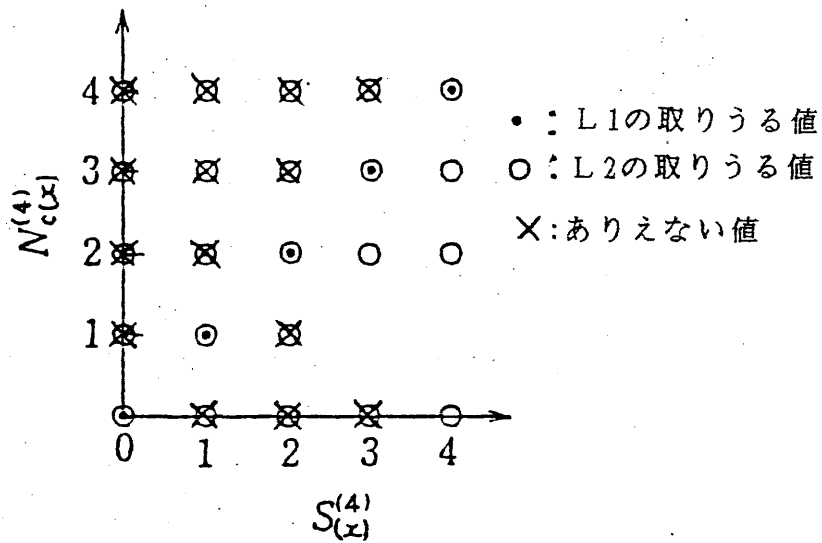


図2. A. 1 L1, L2の $Nc^{(4)}(x)$
と $S^{(4)}(x)$ の関係

2. A. 2 ストローク構造解析法の解説

ストローク構造解析法は、木村らによって開発された手書き文字の認識方法であり、細線化された図形（線図形）を非決定的にストローク抽出して文字を認識する方法である。その概略の説明は、第1章で述べたが、ここではもう少し詳しい解説をする⁽¹⁴⁾。

(1) 文字図形のグラフ表現

細線化処理までは、データの構造は、画像を表す2次元配列であるが、それを折線近似するときには、線図形の特徴点（端点、連結点、分岐点、交点、孤立点）をもとにしてグラフ表現にしている。文字図形をグラフ表現に変換する手順を以下に示す。

① 線図形の特異点の抽出

細線化図形における“図”の要素（1要素）は4近傍にある1要素の数によりつぎのように分類できる（“図”の部分では4連結の連結性を考えている）。

(1) 孤立点: $S^{(4)}(x) = 0$

(2) 端点 : $S^{(4)}(x) = 1$

(3) 連結点: $S^{(4)}(x) = 2$

(4) 分岐点: $S^{(4)}(x) = 3$

(5) 交点 : $S^{(4)}(x) = 4$

この分類を行った例を図2. A. 2に示す。この分類法は直観とよく一致するのが分かる。ここで、連結点以外の点は特異点と言わ

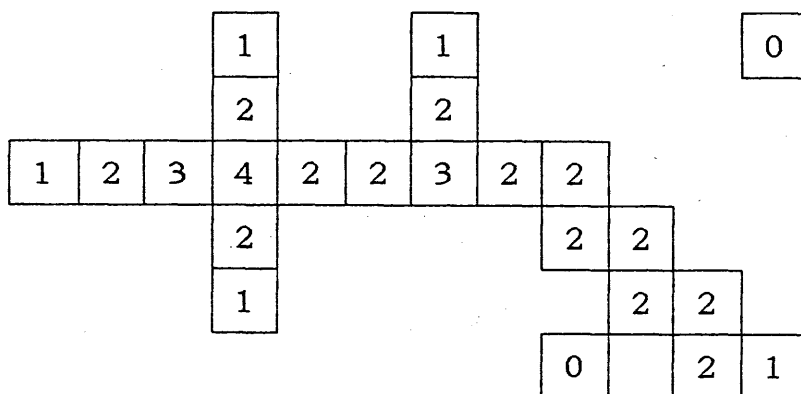


図 2. A. 2 細線化図形の 1 要素の $S^{(4)}(x)$ による分類

れており、線図形をグラフ表現するときに重要な意味を持つ。

② 線図形の要素列分解

要素列分解は、線図形のある特異点を出発して、連結点のみを通して他の特異点にいたるすべての要素列を重複なく求める処理である。要素列分解の例を図 2. A. 3 に示す。要素列の両端は、特異点であり、それらの間は要素列の長さが 2 以下の場合を除いて、連結点で結ばれている。また、特異点は、 $S^{(4)}(x)$ 個だけ分解結果に重複して含まれている。このアルゴリズムについては文献 (14) に詳しく書かれているので、ここでは省略する。

③ 要素列の折線近似法

要素列を構成している特徴的な点だけの x, y 座標を記憶しておけば、その要素列の大まかな位置、形状を知ることができる。特徴的な点としては、両端点と線分の途中にある折点が考えられる。折点を見つける手法は、折線近似の方法として知られており、種々の方法が開発されている。ここでは、両端点を直線で結ぶ線分から一番

遠い点を折点とする再帰的手法を使用している。

要素列 $y_0 y_1 \cdots y_M$ の両端点を結ぶ直線 $\overline{y_0 y_M}$ と要素列を構成している各要素の間の距離 h_i を求め、最大の距離 h となる要素 y_n を求める (図 2. A. 4)。

$$h \geq D \quad (D: \text{最大許容誤差}) \quad (2.A.5)$$

を満たしているならば、この要素列を 2 つに区切り、要素列 $y_0 \cdots y_n$ と要素列 $y_{n+1} \cdots y_M$ に分解する。

この処理を全要素列に対して繰り返して行くと、誤差が D 以下である折線近似パターン (ほぼ直線である要素列の集合) が得られる。

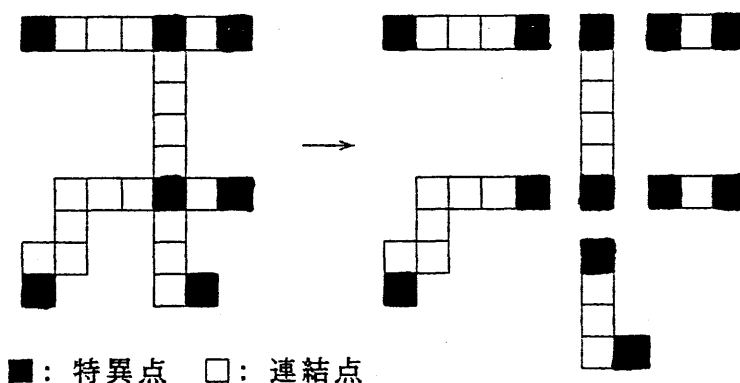


図 2. A. 3 要素列分解の例

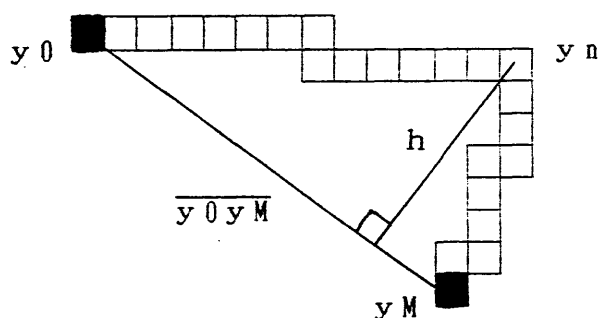


図 2. A. 4 折線近似法

④ 折線近似パターンの表現

折線近似パターンを表現するには、①折線を構成している線分の集合 $E = \{e_i\}$ と②線分の両端に位置する特徴点（特異点＋折点）の集合 $V = \{v_i\}$ 、③線分 e_i の両端の点の特徴点 v_i と v_j を示すための写像 $\Phi(e_k) = (v_i, v_j)$ 、④特徴点の画像上の位置に関する写像 $\Gamma(v_i) = (x_i, y_i)$ が必要であり、次のようなデータ構造（配列）によって、FORTRANでは表現している。なお、表中の数値は、図2. A. 5の文字パターンについて示している。

(i) 線分と両端の点の関係

[写像 $\Phi(e_k) = (v_i, v_j)$]

(線分の番号) ① ② ③ ④ ⑤ ⑥ ⑦

(始点の番号)

1	5	6	3	2	3	8
---	---	---	---	---	---	---

(終点の番号)

5	6	7	7	3	8	4
---	---	---	---	---	---	---

(ii) 点の画像上での位置

[写像 $\Gamma(v_i) = (x_i, y_i)$]

(点の番号) 1 2 3 4 5 6 7 8

(x 座 標)

x_1	x_2	x_3	...	x_8
-------	-------	-------	-----	-------

(y 座 標)

y_1	y_2	y_3	...	y_8
-------	-------	-------	-----	-------

(連結数)

1	1	3	2
---	---	---	---

特徴点の番号は、画像上の左上から横方向にスキャンし、特異点を見つけた順に付ける。その後、特異点で結ばれた線分を前述した方法で折線近似し、折点を表に追加する。また、線分の番号は文字を筆記した順とした。

(2) ストローク抽出と特徴点の対応づけ

折線近似したパターンについて、” 楷書において一筆で書かれる部分 (ストローク) ” に分解する方法を” ストローク抽出法 ” といっており、主に交点、分岐点における線分の交差角度の統計的確率をもとに、線分の統合を行う。この操作により、線分の集合は複数個の組に分解される。しかし、人間の筆記するストロークが必ずしももっとも確率の高い分解とはならないので、この分解を1つの文

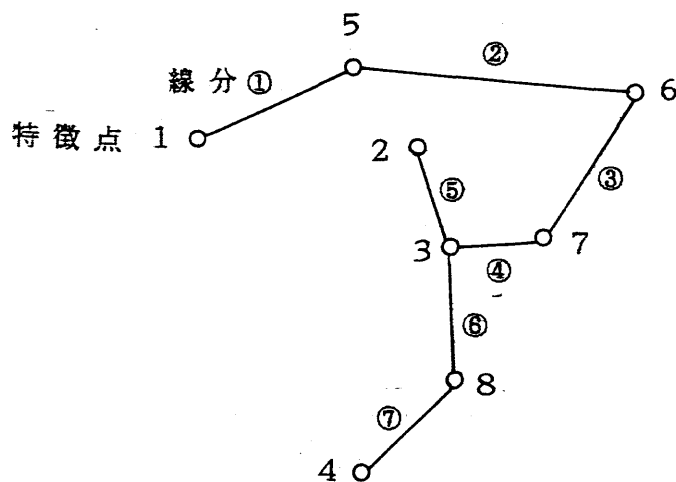


図 2. A. 5 折線近似パターンの例

字について複数回行い、複数個の分解方法を求める（非決定的ストローク抽出）。図2. A. 6は入力文字を2つのストロークに分解した例を3通り示す。

特徴点の対応づけは、標準パターンのストロークと入力パターンのストローク間で対応づけを行い、ストロークの対応づけのできた字種について、動的計画法により最小の相違度となるような特徴点の対応づけを行う（図2. A. 7）。図2. A. 6の例では(a)の場合には、“ア”と、(b)の場合には“タ”と対応づけができるが、(c)の場合には対応づけは困難となる。

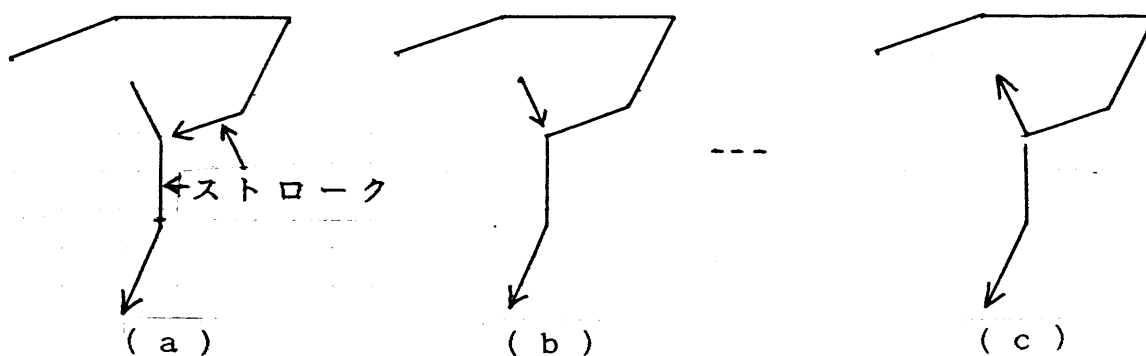


図2. A. 6 ストローク抽出

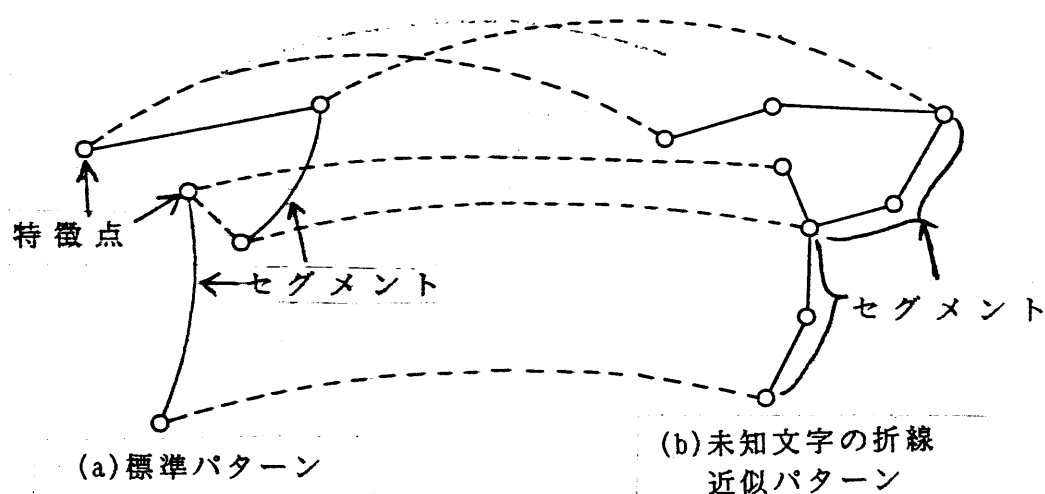


図2. A. 7 動的計画法による特徴点の対応付け

そして、次に述べる識別用の特徴量の計測は対応づけるのできた特徴点間（セグメント）で行う（下降的特徴抽出）。

（３）識別用相違度の計算と識別

特徴量としては、①特徴点の位置関係に関する特徴量 x_p （特徴点を結ぶ線分と水平線のなす角 θ_i を要素とするベクトル）と②セグメントの形状に関する特徴量 x_s （セグメントの両端点を結ぶ直線からの符号付き最大偏位 h_i を要素とするベクトル）を用いた（図 2. A. 8）。

また、識別用相違度としては、特徴量 x_p のマハラノビス距離と特徴量 x_s のユークリッド距離、標準パターンを構成するストロークと対応の付かない線分をノイズと見なしたときのペナルティを与える関数の和を用いた（式 (2. A. 6)）。

$$\begin{aligned}
 D^l(x) = & \frac{1}{M1^l} (x_{p^l} - \mu_{p^l})^t \Sigma^{-1} (x_{p^l} - \mu_{p^l}) \\
 & + \frac{1}{M2^l} \sum_{i=1}^{I-1} (x_{si} - \mu_{si})^2 \\
 & + \sum_{e \in Eu} (|e| / 2.0)^2 \{1 - P("e = \text{ヒゲ}"))\}
 \end{aligned}
 \tag{2. A. 6}$$

但し、 $M1^l$ ：字種 l の x_{p^l} の次元数

μ_{p^l} ：字種 l の x_p の平均ベクトル

Σ^{-1} ： x_{p^l} の共分散行列の逆行列

$M2^l$ ：字種 l の標準パターンのセグメントの数

μ_{si} : 字種 i の x_p の平均ベクトル

E_u : 字種 i の標準パターンと対応づけを行った場合の
未使用線分の集合

$|e|$: 線分 e の長さ

$$P("e=\text{ヒゲ}") = P(\theta_e, |e|) \quad (2.A.7)$$

ここで、 $P("e=\text{ヒゲ}")$ は線分 e がヒゲである確率で、実際に筆記された文字より計測されたヒゲの角度 θ_e と長さ $|e|$ (図 2. A. 9) の頻度分布から、与えられるものとする。

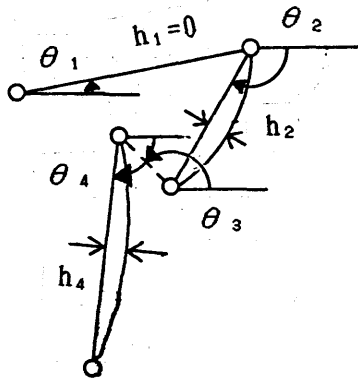


図 2. A. 8 識別用特徴量

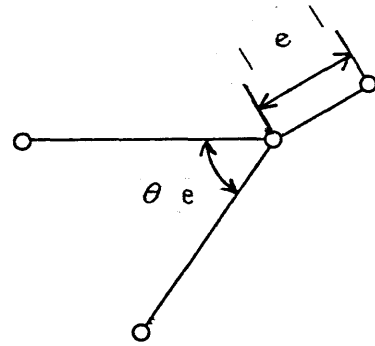


図 2. A. 9 "ヒゲ" の確率計算

第3章 人間の字種識別基準を用いた

自由手書きカタカナ文字認識

あらまし

従来の文字認識方式では、標準字体からの変形がわずかで、人間なら困難なく読める文字でも、正読とならない場合が散見される。その理由の1つとして筆者らは、手書き文字における「言行不一致」、すなわち実際に筆記される文字の集合が識別可能な文字の概念的分布（字種識別範囲）を正しく反映していないという仮説を立て、より人間に近い性能を示す認識方式を構成するために、この字種識別範囲の計測を試みた。

本章では、字種識別範囲の計測方法とその計測結果を示し、これを取り入れた識別関数（心理的距離）を提案する。本方法を用いて自由手書きカタカナ文字（ETL1）の認識実験を行った結果、統計的手法のみを用いる場合に比べ誤読率が半分以下となり、未学習文字4600文字に対する正読率が従来の最高値を上回る97.2%となった。又、誤読・棄却された文字も人間が識別しにくい文字に限られることが判明した。

3. 1 はじめに

文字認識方式の研究には、次の2つの方向が見られる。(1) 識別関数はユークリッド距離など簡単なものに固定し、特徴抽出の工夫によって性能向上を図る⁽¹⁾⁻⁽³⁾。(2) 特徴量の抽出方法は単純にして、より高次の統計量(分散)等を含む精密な識別関数を用いる^{(4),(5)}。(1)に属する研究成果はこれまでに多く発表されているが、(2)に属するものは少く、未解決の問題も多い。

まず第1の問題点として、2次識別関数で用いる標本分散と母集団の分散とのずれが考えられる。この対策として、現段階では学習文字数を増やしてなるべく2つの分散を近づける努力を行っている。しかし、現在研究されている文字認識方法の特徴ベクトルの次元数を考えると、現実的なレベルで学習文字数を増やしても、認識率に大きな変化はなく⁽⁶⁾、多大な労力にもかかわらず、その効果は少ない。

第2の問題点として、筆記文字においても、^{*}言行不一致^{*}の現象、すなわち^{*}行為の所産である筆記文字の分布と識別可能な文字の概念的分布は異なっている^{*}という仮説が考えられる。人間は筆記の際に文字の概念的分布の範囲(字種識別範囲)に入る文字だけを書くとは限らない。例えば、「ソ」、「ン」等のように、類似文字を持つ字種では、類似文字に近づくような変形については許容限界が狭く、不注意あるいは筆のすべりによって、意図と異なる字種の識別範囲に入る文字を書くことが考えられる。又類似文字を持たない字種では、実際に書かれた文字から許容限界付近の文字を取

集することは不可能であると考えられる。このことは第1の問題点より重要であり、すべての筆記文字を集め、母集団の統計量が正確に計測できても、この問題点を解決しないうちは、人間の字種判読感覚とは異なる特性を示すシステムになると考えられる。

そこで、筆者らはこの仮説が正しいかどうかを立証するために、各字種について文字の概念的分布（字種識別範囲）を計測する方法を考案し、従来明確にとらえられていなかった字種ごとの変形許容限界値（判読限界値）を初めて計測した。字種識別範囲は本来主観的なものと考えるのが自然であるが、人間の文字読取実験^{(7)・(8)}から、正常人での個人差は小さく、あいまいな文字はだれが見てもあいまいであり、そのような文字に対してだけ、読取り結果に個人差が認められる。このため、少人数の計測結果で多人数のそれを代表しても弊害はほとんど生じないと考えられる。

そこで、この計測結果を用いて識別関数（心理的距離）を定義し、ストローク構造解析法により自由手書きカタカナ文字（ETL1）を認識させた。その結果、統計的手法のみを用いる場合に比べ誤読率が半分以下となり、未学習文字に対して正読率97.2%、誤読率1.3%という好成績が得られた。また、誤読・棄却した文字も人間が識別しにくい文字に限られていることが判明した。

3. 2 変形文字の発生

本節では、人間の字種識別範囲の計測に用いる変形文字を計算機で発生させる方法について述べる。その方法の全体の流れ図を図3.1に示す。

3. 2. 1 前処理

入力データは、電総研・富士通作成の自由手書きカタカナ文字データベース（ETL1）の16値パターンに2値化、細線化⁽⁹⁾、折線近似等の処理を施して得られた折線近似文字とし、これから、非決定的ストローク抽出、および下降的特徴抽出⁽¹⁰⁾により主特徴点を抽出する。

3. 2. 2 主特徴点と副特徴点

入力文字の特徴点としては、文字の概形を表現する主特徴点（main feature point）（標準パターンの主特徴点と対応づけのできた点）のほかに、文字線分の微妙な形状を捕らえるための副特徴点（sub feature point）を考える。副特徴点の総数は主特徴点の数の約2倍とし、セグメント（主特徴点の間の線分）ごとの副特徴点の数 D_i は、平均パターンの各セグメントの長さに応じて式（3.1）により配分する。この数は主特徴点抽出用辞書に登録しておく。配分された副特徴点はセグメント上に等間隔に配置する。

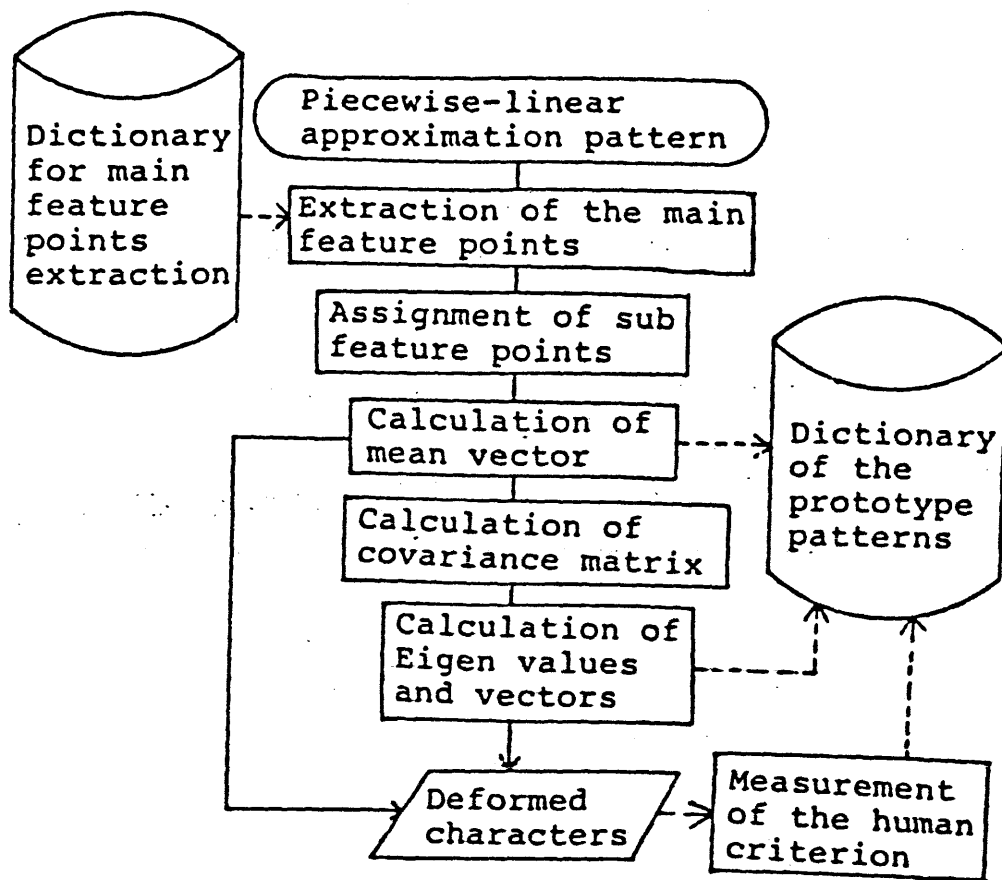


図 3. 1 変形文字の発生と辞書の作成

$$D_i = \left[\frac{a_i}{\sum_{i=1}^N a_i} \right] \cdot 2M + 1 \quad (3.1)$$

但し, a_i : 第*i*セグメントの長さ

[]: ガウス記号

N: セグメント数

M: 主特徴点の数

3. 2. 3 特徴ベクトル

学習段階あるいは認識段階では、各文字パターンから、特徴点（主特徴点と副特徴点）の x, y 座標の値を成分とするベクトル

$$X = (x_1, y_1, x_2, y_2, \dots, x_{n/2}, y_{n/2}) \quad (n\text{次元}) \quad (3.2)$$

を求める。但し、添字は特徴点の通し番号とし、特徴点の総数を $n/2$ とする。

このベクトル X に文字の位置・大きさの正規化に対応する操作を施し、特徴ベクトル x を作成する。

$$x = \frac{X - G}{\|X - G\|} \quad (3.3)$$

但し、 $G = (\bar{x}_1, \bar{y}_1, \dots, \bar{x}_{n/2}, \bar{y}_{n/2}) \quad (n\text{次元})$

\bar{x}_i : 特徴点の x_i 座標の平均値 ($i=1, 2, \dots, n/2$)

\bar{y}_i : 特徴点の y_i 座標の平均値 ($i=1, 2, \dots, n/2$)

3. 2. 4 変形文字の発生

与えられた文字サンプルの特徴ベクトルを x_1, x_2, \dots, x_N とする。 x_α ($\alpha=1, 2, \dots, N$) の平均ベクトルを \bar{x} , 共分散行列を Σ とし、 Σ の固有値を λ_i , 固有ベクトルを ψ_i ($i=1, 2, \dots, n$) とする。 N が十分に大きいならば、平均ベクトル \bar{x} が表している文字の形状をすべての筆記文字の平均的形状と考えることができる。

x がある文字を表しているとき、 $x - \bar{x}$ は x の \bar{x} からの字形変動すなわち変形を表し、 $(x - \bar{x}) \cdot \psi_i$ はその変形の第 i 主成分を表し

ている。従って任意の変形の主成分を持つ変形文字、すなわち変形の第*i*主成分が $k_i\sqrt{\lambda_i}$ (k_i : 変形指数) となる文字は

$$x = \bar{x} + \sum_{i=1}^n k_i \sqrt{\lambda_i} \psi_i \quad (3.4)$$

で与えられ、特に、変形の第*i*主成分が $k_i\sqrt{\lambda_i}$ 、その他の成分が0となる文字（主成分軸上の文字）は

$$x = \bar{x} + k\sqrt{\lambda_i} \psi_i \quad (3.5)$$

で与えられる。 k を $\pm 1, \pm 2, \dots$ などと変化させて得られる x を図表現（図3. 2）することによって、変形の第*i*主成分の意味づけを視覚的に行うことができる。又この図表現を被験者に呈示し、各主成分ごとにどの程度の変形まで文字として識別可能かどうかを調べることができる。

3. 2. 5 主成分の意味づけ

発生した文字をXYプロッタで出力し、低次の主成分によって変形させた文字と高次の主成分によって変形させた文字を比較すると、低次の主成分は各ストローク間の相対的位置変動（図3. 3（b））、ストロークの傾き変動（図3. 3（c））のような典型的な変形であり、高次の主成分は微妙な筆跡のゆらぎ（図3. 3（d）、（e））を表していることが分る。又、字種によっては、細線化処理

ア ア ア ア ア ア ア ア ア
平均パターン -15 -25 -35 -45 -55 -65 -75 -85 -95

ア ア ア ア ア ア ア ア ア
平均パターン 15 25 35 45 55 65 75 85 95

(a) 1st principal axis

ア ア ア ア ア ア ア ア ア
平均パターン -15 -25 -35 -45 -55 -65 -75 -85 -95

ア ア ア ア ア ア ア ア ア
平均パターン 15 25 35 45 55 65 75 85 95

(b) 2nd principal axis

ア ア ア ア ア ア ア ア ア
平均パターン -15 -25 -35 -45 -55 -65 -75 -85 -95

ア ア ア ア ア ア ア ア ア
平均パターン 15 25 35 45 55 65 75 85 95

(c) 3rd principal axis

ア ア ア ア ア ア ア ア ア
平均パターン -15 -25 -35 -45 -55 -65 -75 -85 -95

ア ア ア ア ア ア ア ア ア
平均パターン 15 25 35 45 55 65 75 85 95

(d) 4th principal axis

図3. 2 各主成分軸上の変形文字（上段：負方向，下段：正方向，数字は平均から分散の何倍離れているかを表す）

によって文字線分が鋭角で曲がっている場所に、ヒゲ状ノイズ枝が発生する文字がいくつか表れるため、それを反映している主成分が高次の主成分に表れる（図3. 4）。一般に、高次の主成分では、異字種に接近する因子は抽出できず、字種を識別する観点からは低次の主成分の情報が重要であると考えられる。そこで次節以下では高次の主成分を使用しないことにした。

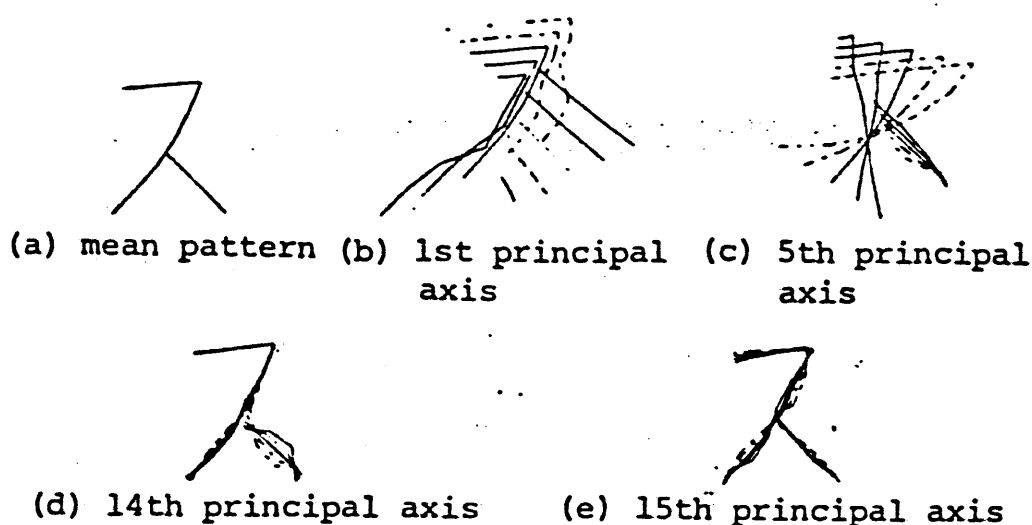


図3. 3 主成分の図表現
（実線は正方向、破線は負方向の変形）

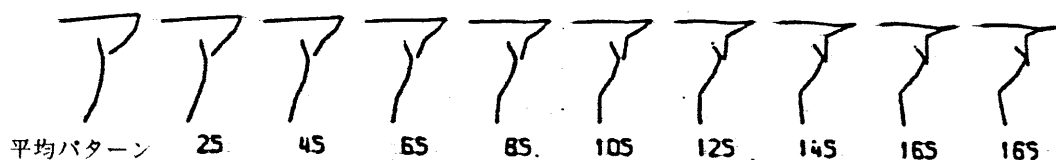


図3. 4 ノイズ枝を反映している主成分軸（第13軸）
（右上の突き出しはノイズ枝の影響）

3. 3 判読限界値

3. 3. 1 計測方法

前節の方法により発生させた変形文字を被験者に呈示し、どの変形文字までをその字種として識別できるかを k の値(判読限界値)で記入させた。 k の値としては整数値以外も許容した。被験者は文字認識の研究をしている大学院生1人とこの研究内容に習熟していない卒業研究生5人である。

3. 3. 2 計測結果

表3. 1に大学院生の判読限界値(U_{i+} , U_{i-})を示す。卒業研究生の平均値との比較を図3. 5に示す。ここで U_{i+} , U_{i-} はそれぞれ第 i 主成分における k を正の値とした時の判読限界値、 k を負の値とした時の判読限界値である。

卒業研究生は正読可能であっても、主観的に好ましくない変形文字を排除する傾向があり、ストローク全体のバランス、交差状態などの変形の許容度が小さい。大学院生は正読が可能か否かに重点を置くこと、実際に筆記された文字のバラツキ具合を把握していることなどの理由により、変形の許容度を大きくとると共に、類似文字を考慮して決定している。

次に字種ごとの判読限界値の全体の傾向に注目すると、高次の主成分ほど正負方向とも判読限界値の絶対値が大きくなる字種が最も

多い。すなわち、高次の変形の主成分は、文字の形を視覚的に大きく変化させていないことを意味する。しかし、中には‘ノ’のように判読限界値が中位の主成分で大きくへこむ字種もあり、細かく見ると字種ごとの判読限界値はかなり異なっている。

従来、文字認識を行う場合には、多数の学習文字に対して統計的処理を行い、字種内の文字の存在確率を求め、それを識別用辞書に使用していた。そして、その文字の存在確率は、主成分軸に対して正負対称であり、分散の同一定数倍のところでは軸番号に依存せず同一であった。しかし、この計測結果によれば、人間が把握している文字の概念的な識別範囲は主成分軸に対して正負非対称であり、軸番号により許容範囲は大きく異なり、実際の学習文字の分布範囲と人間の文字の識別範囲は重なっていない領域がかなり広いことが判明した。

表3. 1 各字種の判読限界値(大学院生)(その1)

主成分 軸番号		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
ア	-	6	6	5	6	4	5	8	8	8	8	6	10	10						
	+	5	5	4	3	5	4	5	8	6	5	8	16	8						
イ	-	5	6	5	5	6	6	7	5	7	18									
	+	10	5	12	5	5	12	8	5	8	12									
ウ	-	7	6	5	11	6	6	5	5	5	6	5	7	8	10	13	12			
	+	8	8	3	13	6	4	4	10	7	10	8	9	10	8	15	15			
エ	-	11	19	6	5	4	5	6	8	12	16	9	15	15						
	+	4	4	5	5	6	12	14	5	5	6	9	15	15						
オ	-	5	5	7	6	9	5	9	5	7	9	9	10	8	60	12				
	+	6	9	5	5	5	6	5	12	8	9	5	12	9	20	12				
カ	-	5	5	11	7	4	5	9	10	9	28	27	18							
	+	6	7	5	4	6	8	14	10	11	6	12	15							
キ	-	7	6	5	6	5	9	9	8	27	12	11	39	30						
	+	6	13	5	7	7	9	7	9	52	12	11	39	100						
ク	-	6	7	3	5	4	6	10	6	8	40	9	12							
	+	4	6	6	6	12	4	5	10	5	12	9	10							
ケ	-	5	8	5	5	30	5	7	5	7	9	7	10	10	13					
	+	5	4	9	7	9	6	5	6	8	6	7	9	10	15					
コ	-	9	6	6	5	9	7	6	7	6	100									
	+	5	4	14	9	9	5	9	11	7	35									
サ	-	4	6	11	6	6	6	4	10	6	10	10	13	50	12					
	+	6	12	3	7	4	6	17	6	6	10	10	11	18	18					
シ	-	11	5	5	6	4	5	6	4	7	30	5	10	8						
	+	3	4	5	14	5	5	6	6	6	12	6	7	10						
ス	-	4	5	6	14	10	9	11	10	4	9	12	9							
	+	6	10	6	9	8	7	8	12	10	5	12	9							

表3. 1 各字種の判読限界値（大学院生）（その2）

セ	－	5	2	6	5	5	5	4	10	6	5	6	10	10	12	14			
	＋	3	4	5	6	6	6	6	5	6	7	8	14	10	8	12			
ソ	－	7	6	6	5	7	5	6	8	9	24								
	＋	7	15	5	5	4	3	4	12	12	50								
タ	－	5	5	13	4	6	5	5	7	6	7	9	10	9	12	15	12		
	＋	4	4	12	6	6	8	7	5	7	8	9	12	10	9	50	12		
チ	－	4	4	3	15	6	6	5	5	4	4	6	10	15	14	7			
	＋	5	6	9	6	6	7	5	11	6	5	14	26	10	10	10			
ツ	－	6	10	12	15	6	4	5	6	8	9	7	7	10	15				
	＋	4	8	13	6	5	6	6	6	7	8	6	7	8	12				
テ	－	14	10	14	6	6	5	8	6	10	4	10	10	15					
	＋	6	14	6	5	6	7	9	8	6	12	12	12	12					
ト	－	5	4	10	15	5	5	7	15	10									
	＋	5	6	5	11	7	5	12	15	20									
ナ	－	6	4	5	5	7	7	5	20	12	12								
	＋	20	5	5	5	5	7	12	18	12	10								
ニ	－	6	12	6	5	18	10	10	40										
	＋	5	6	7	5	100	15	9	25										
ヌ	－	4	4	6	6	9	4	13	12	15	9	10	10						
	＋	7	4	7	7	6	7	7	9	8	12	12	15						
ネ	－	4	5	7	6	7	5	6	10	7	6	10	10	8	12	12	14	17	18
	＋	6	8	10	7	4	5	10	6	9	7	6	10	10	12	9	20	12	15
ノ	－	20	14	4	10	12	10												
	＋	15	5	13	10	10	15												
ハ	－	5	7	7	5	6	9	7	12	99									
	＋	6	5	5	6	5	7	8	10	18									
ヒ	－	4	5	5	6	7	9	5	6	7	10	9	12						
	＋	6	8	4	7	7	6	8	6	9	11	12	12						

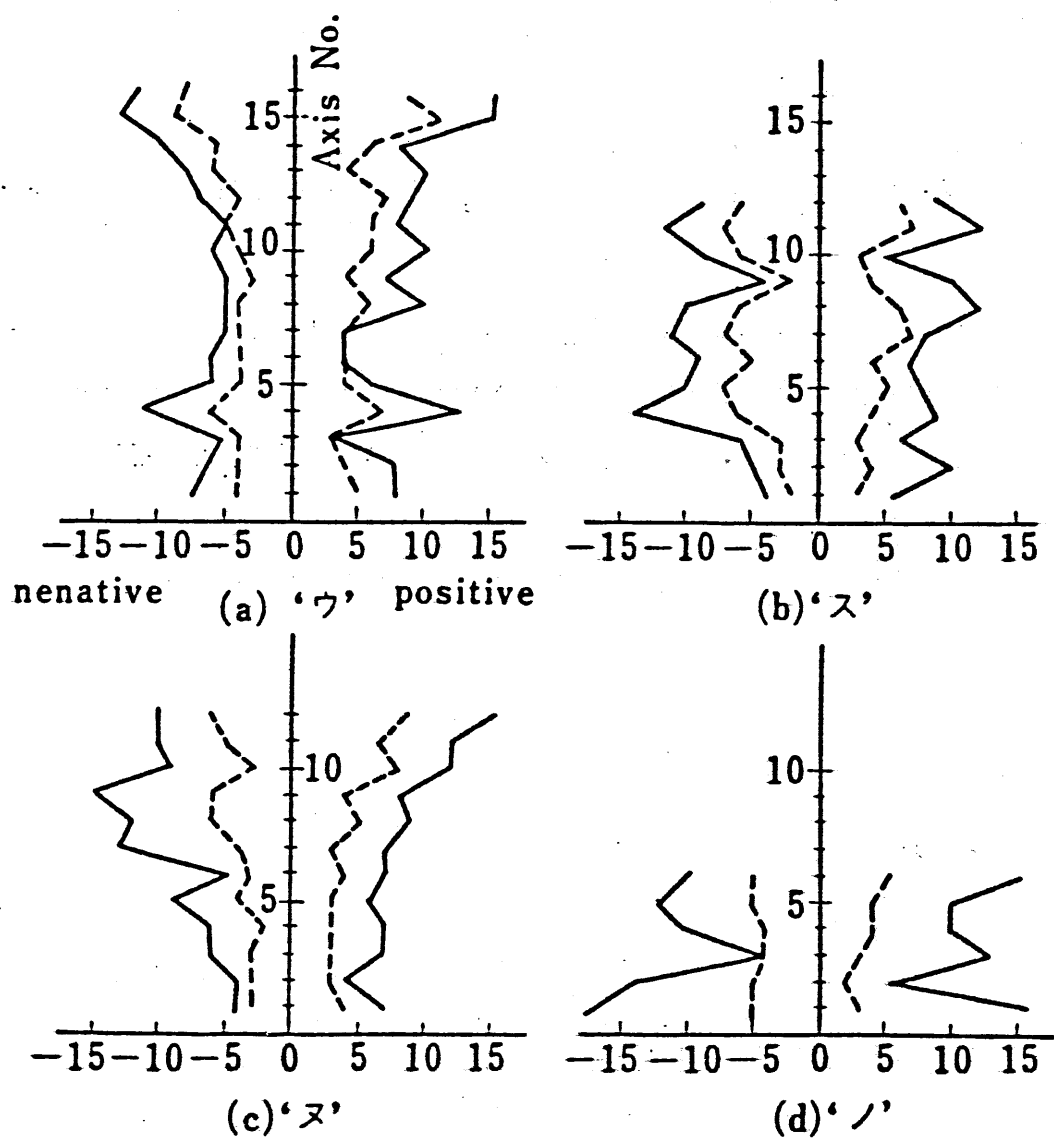
表3. 1 各字種の判読限界値（大学院生）（その3）

フ	-	6	5	14	5	5	8	12												
	+	3	6	5	6	6	5	10												
ヘ	-	7	4	4	6	10	8	11	7											
	+	5	6	5	5	14	8	15	10											
ホ	-	6	5	5	9	5	6	7	8	6	7	11	7	6	10	7	15	12	15	25
	+	5	5	6	7	5	6	6	6	6	10	7	9	7	9	9	15	15	15	35
マ	-	5	5	4	4	6	7	3	4	8	10	7	9							
	+	4	5	3	3	5	5	6	6	12	8	9	7							
ミ	-	12	4	5	5	10	6	6	6	9	7	30	36							
	+	4	12	5	6	10	6	8	7	8	7	25	30							
ム	-	6	6	4	6	7	5	5	6	5	15	12	12							
	+	7	6	4	5	5	8	5	5	5	100	15	9							
メ	-	4	9	4	5	5	7	12	8	10	12	9								
	+	4	4	5	4	4	6	4	100	12	10	12								
モ	-	9	10	8	5	5	10	5	5	7	10	6	10	9	25	18				
	+	8	4	5	6	10	6	5	5	7	6	9	12	10	100	15				
ヤ	-	4	5	5	5	4	7	14	5	10	7	60	8							
	+	6	5	3	3	5	6	11	9	9	8	15	9							
ユ	-	7	4	5	6	8	7	8	12	11	12									
	+	3	4	6	6	8	7	12	10	10	12									
ヨ	-	12	6	5	8	8	6	7	10	10	14	12	9	10	40					
	+	6	5	14	6	7	5	9	9	10	10	9	12	10	10					
ラ	-	5	6	5	7	6	10	8	6	12	10	20								
	+	6	12	6	6	6	10	8	7	16	10	10								
リ	-	4	15	10	5	6	6	7	5	25	15									
	+	7	10	5	5	6	10	26	6	8	12									
ル	-	5	10	6	6	8	7	7	11	7	12	8								
	+	5	5	7	5	7	7	6	7	11	12	8								

表3. 1 各字種の判読限界値（大学院生）（その4）

レ	-	4	6	4	6	8	15	8											
	+	4	5	6	5	11	100	9											
ロ	-	5	10	13	6	7	5	6	5	10	9	12	8	12	42				
	+	4	5	4	6	9	5	7	5	9	10	7	10	15	15				
ワ	-	5	5	10	5	5	4	3	13	9	4	14	8						
	+	7	7	4	5	6	7	6	5	4	6	8	10						
ヲ	-	15	6	4	6	6	7	7	8	10	20	7	7						
	+	5	6	7	7	9	11	8	8	8	8	6	8						
ン	-	10	6	4	6	3	4	4	7	9	6								
	+	5	3	5	7	7	6	5	4	5	20								
カ	-	5	4	4	5	4	8	4	7	9	7	9	9	8	30				
	+	6	5	8	4	5	8	8	8	5	5	12	6	8	15				
コ	-	7	7	8	9	7	8	10	6	7	9	8	10	15					
	+	5	9	8	10	6	7	8	8	8	8	9	10	15					
ケ	-	6	10	4	5	9	6	10	10	8									
	+	4	5	6	3	8	6	12	6	15									
ク	-	3	4	3	3	7	12	6	9										
	+	2	3	6	4	4	7	5	14										
ネ	-	5	4	5	9	6	7	6	10	9	7	9	18	15	15	20	20		
	+	7	5	4	8	8	11	8	6	10	8	9	12	15	10	15	20		
ネ	-	2	2	4	8	5	10	4	8	12									
	+	4	4	3	5	6	8	8	10	8									
ネ	-	2	2	3	5	5	5	7	10	10	10	30							
	+	4	4	4	4	6	5	9	12	14	15	15							
チ	-	2	3	3	4	6	5	4	8										
	+	4	5	10	3	6	6	6	8										

(注) - : U_i^- , + : U_i^+ ($i=1, 2, 3, \dots$)



solid line : postgraduate student
dotted line : undergraduate students
(average over five subjects)

図3.5 判読限界値

Fig.3.5 Permissible limit of reading

3. 4 識別方法

本節では、計測された判読限界値に基づく新たな識別関数を提案すると共に、棄却条件について考察する。

3. 4. 1 標準パターン

字種を識別するために用いる標準パターンは、学習文字を解析して得られた平均パターン \bar{x} (1: 字種番号), 固有ベクトル ψ_i ($i=1, 2, \dots, n$), 前章で求めた判読限界値 U_i^+, U_i^- ($i=1, 2, \dots, m$; +: 正方向, -: 負方向) である。但し, m は次式で定まる累積寄与率 AP を 99% 以上とする軸数 p の最小値であり, 一般に字種により異なる。

$$AP = \sum_{k=1}^p \lambda_k / \sum_{k=1}^n \lambda_k \quad (p < n) \quad (3.6)$$

図 3. 6 に「ア」, 「イ」の累積寄与率の変化を示す。一般に, m は総軸数 (特徴ベクトルの次元 n は最大 44 (「ホ」), 最小 12 (「ノ」), 平均 29. 2 である) の $1/2$ 以下であり, 最大 19 (「ホ」), 最小 6 (「ノ」), 平均 11. 9 である。

3. 4. 2 識別関数と棄却条件

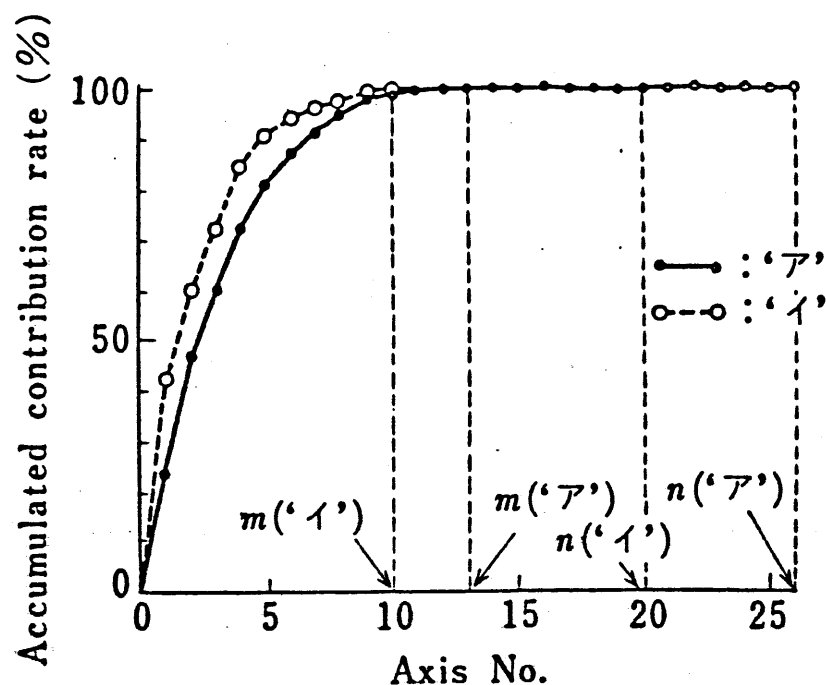


図 3. 6 累積寄与率

(1) 識別関数の定義

認識すべき文字パターンの特徴ベクトルを x とすると、その第 i 主成分 u_i は次式で求められる。

$$u_i = (x - \overline{x})^t \cdot \psi_i \quad (3.7)$$

但し、 i : 字種番号

この u_i と判読限界値 U_i ($i=1, 2, \dots, m$) を用いて新しい識別関数 [心理的距離: psychological distance (PD)] を次式により定義する。

$$d_{12}(x) = \sum_{i=1}^m \left(\frac{u_i}{s_i} \right)^2 \quad (3.8)$$

但し, ${}_i s_i = {}_i U_i / \sigma_i$

${}_i \sigma_i$: 第 i 主成分の標準偏差 ($= \sqrt{{}_i \lambda_i}$)

$${}_i U_i = \begin{cases} {}_i U_i^+ & ({}_i u_i \geq 0) \\ {}_i U_i^- & ({}_i u_i < 0) \end{cases}$$

${}_i U_i^+$: 第 1 字種, 第 i 軸の正方向判読限界値

${}_i U_i^-$: " " 負方向 "

この定義式で m 以上の高次の項を計算に入れていないのは, 分母の判読限界値が大きくなるため, 実在の文字では, 識別関数の値にほとんど影響がないからである. また, この式が 1 となる特徴空間での曲面は, 各軸の正負判読限界値を通る区分的超楕円面であり, 象限ごとにその長軸, 短軸は異なる.

(2) 従来の 2 次識別関数との比較

${}_i U_i$ が i に依存しない定数であれば, 式 (3. 8) は従来の 2 次識別関数 (マハラノビス距離) と同等となる. しかし, 3. 3. 2 の結果は ${}_i U_i$ が i 及び ${}_i U_i$ の正負により大きく異なることを示しており, 式 (3. 8) と従来の 2 次識別関数の差は意外に大きい. 又, 図 3. 7 が示すように, 従来の 2 次識別関数では等距離面が左右対称な超楕円体であるのに対して, この識別関数では非対称な区分的超楕円体となる. この性質により, 図 3. 7 の入力文字では, 2 つの識別関数で大小関係が反転し, 識別結果は異なる.

(3) 字種の決定

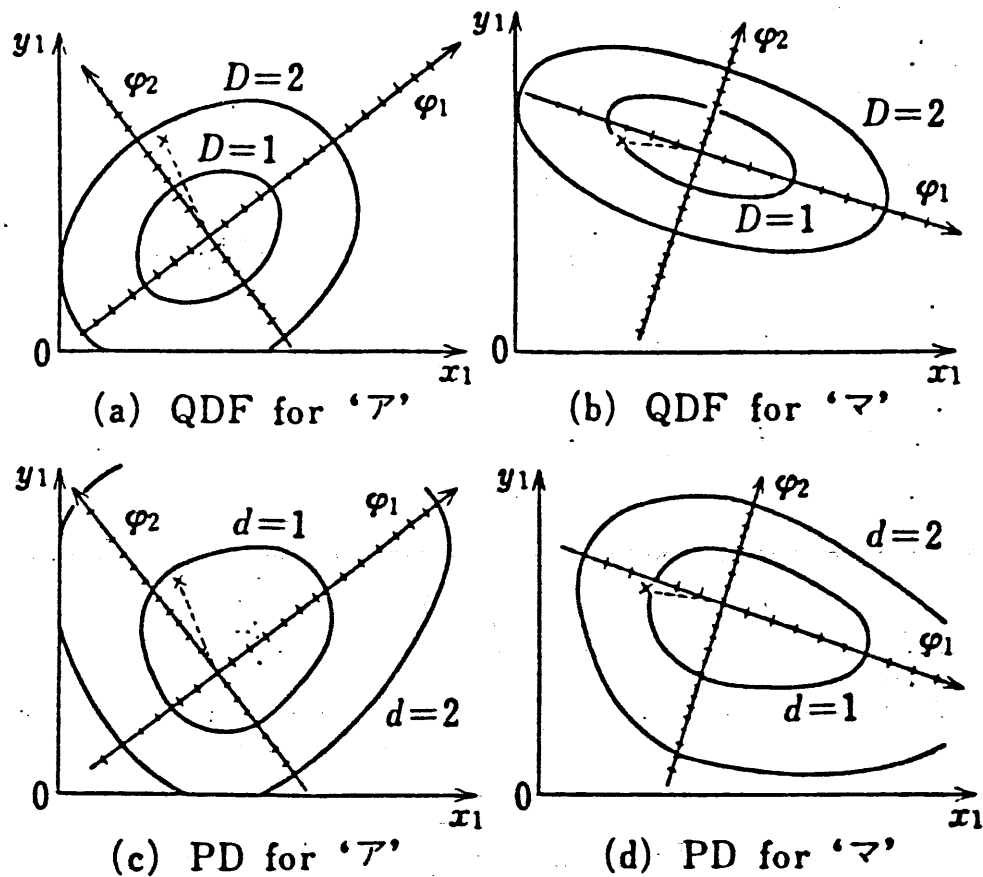


図3. 7 2次識別関数との比較

入力された文字 x に対して識別関数 $d_i^2(x)$ の値を字種ごとに計算し、最小値を与える字種 l ，すなわち次式を満たす字種 l^* を求める。

$$d_{l^*}^2(x) = \min_l \{ d_l^2(x) \} \quad (3.9)$$

この $d_l^2(x)$ の値により，(a)，(b)，(c) の場合に分けて，字種決定を行う。

(a) $d_l^2(x) \leq 1$ のとき

最小値を与える第 l^* 字種に属すると決定する。

(b) $1 < d_{1^2}(x) \leq 2$ のとき

(i) $d_{1^*}(x) - d_{1^*}(x) \geq T$ ならば,

第 1^* 字種に属すると決定する.

(ii) $d_{1^*}(x) - d_{1^*}(x) < T$ ならば,

棄却とする.

但し, $1'$: 第 2 候補となる字種

$$d_{1^2}(x) = \min_{l \neq 1^*} \{ d_{l^2}(x) \} \quad (\text{第 2 候補}) \quad (3.10)$$

T : しきい値

(c) $d_{1^2}(x) > 2$ のとき

棄却とする.

3. 5 認識結果

3. 5. 1 判読限界値の修正

予備実験の結果、3. 3. 2の計測結果をそのまま識別に用いるには不適切と感じられる場合があった。それは被験者が最初に変形文字を見て判読限界値を決めるとき、変形の小さな方から見ていくか、大きな方から見ていくかで判読限界値は異なり、判読限界値についての信頼性が落ちる場合があるからである。そこで信頼性を高めるために、計測された判読限界値を用いて、学習文字についても心理的距離値を計算し、大き過ぎる値のときには、その原因となっている主成分の変形文字を見直し、心理的抵抗を受けない場合に限って、判読限界値の修正を行った。修正した判読限界値は全体の1割以下であり、表3. 1は修正した結果である。

3. 5. 2 識別実験の結果

学習サンプルとして、自由手書きデータベース（ETL1のカタカナ文字部）の第2, 8, 11巻の各字種150文字計6900文字を、テストサンプルとして第18, 19巻計4600文字を使用して識別実験を行った。この識別率（しきい値 $T=0.15$ ：誤読率と棄却率がほぼ等しい時）を従来の2次識別関数の結果⁽⁹⁾と共に表3. 2に示す。この結果は大学院生の判読限界値を用いたものである。卒業研究生の判読限界値では棄却率が7%を超えたので略した。こ

表3. 2 識別率の比較 [%]

	正 読 率	誤 読 率	棄 却 率
心理的距離	97.2	1.3	1.5
二次識別関数	95.6	3.0	1.4
人間の識別率	99.1	0.9	0.0

の差は文字認識の研究を経験している程度のちがいであり、卒業研究生同士における個人差はあまり見られない。ここで人間の識別率はデータベースからランダムに取り出した約300文字ずつに分けて80人の被験者に識別させたときの平均識別率⁽⁷⁾である。

表3. 2から分るように心理的距離では従来の識別関数に比べ誤読率が半分以下になると共に、正読率が向上している。新たに正読となった文字を関数値と共に図3. 8に示す。従来の2次識別関数で誤読となった文字の中には、図3. 8のように人間が容易に判読できる文字もあったが、これらは心理的距離では正読となっており、2位以降も人間の識別結果と似た字種が順に選択されている。この結果は人間の判読限界値を文字認識系にかなり良く反映することができたことを示している。又、図3. 9のような文字も正読となっており、本方式は文字の変形に対して柔軟な読み取り方式であると考えられる。

3. 5. 3 誤読・棄却の原因

主な原因別に誤読・棄却になった文字数(4600文字中)と文

Binary pattern	Piece-wise linear pattern	METHOD	ODER			
			1st	2nd	3rd	4th
		QDF	マ (0.3)	ヌ (0.5)	ス (1.9)	サ (3.2)
		PD	ヌ (0.5)	マ (1.2)	ス (4.3)	ヤ (4.3)
		QDF	チ (0.3)	テ (0.5)	シ (1.0)	
		PD	テ (0.3)	ケ (1.0)	シ (2.2)	
		QDF	ケ (0.8)	シ (5.6)	ミ (6.4)	
		PD	ヨ (0.2)	ヲ (0.7)	シ (1.6)	
		QDF	ソ (0.3)	ン (0.4)	リ (0.4)	
		PD	リ (0.3)	ソ (0.6)	ン (1.5)	シ (1.9)
		QDF	チ (0.2)	ヲ (0.3)	ヌ (1.0)	シ (1.3)
		PD	ヲ (0.3)	キ (1.4)	シ (1.4)	チ (1.5)

QDF:quadratic discriminant function

PD :psychological distance

():the value of discriminant function

図3. 8 認識結果の比較

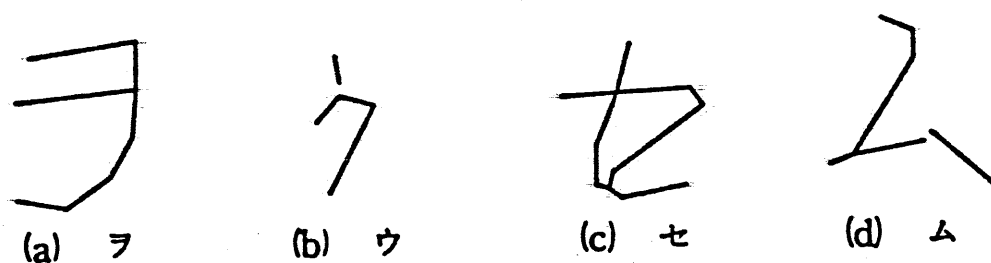


図3. 9 正読となった文字例

字例を示し、残された問題点を分析する。

(1) 2 値化の失敗

2 値化の失敗により、文字が寸断されたり、とぎれが生じて対応づけが正しく行われなかった文字〔16 文字, 図 3. 10 (a), (b)〕。

(2) 細線化・折線近似の失敗

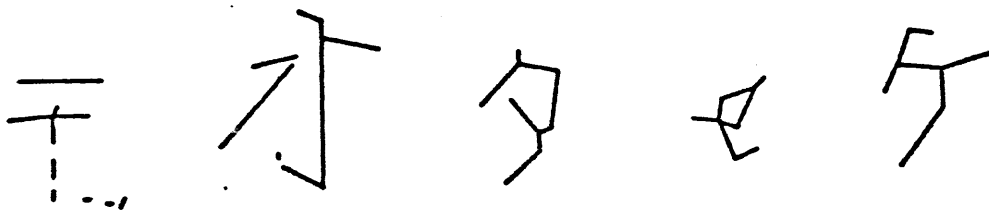
細線化処理における交点・分岐点付近の歪み、線分の鋭角部に発生するヒゲ状ノイズ枝、変形交点処理の誤り等のために対応づけが正しく行われなかった文字〔36 文字, 図 3. 10 (c), (d)〕。

(3) 特徴点抽出の失敗

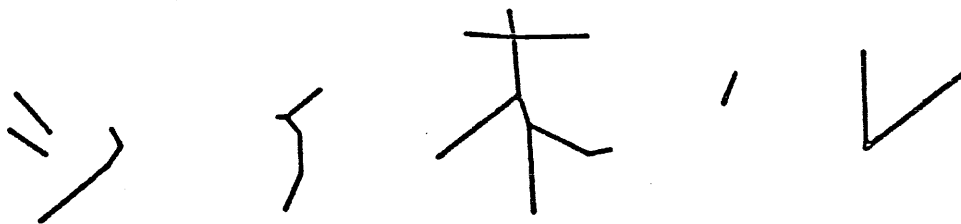
前処理は正常であるが、余分な線分や突き出しなどのために、特徴点の抽出が正しくできなかった文字〔10 文字, 図 3. 10 (e)〕。

(4) 類似文字の筆記上の不注意

‘ク’と‘ワ’、‘ソ’と‘リ’と‘ン’、‘ツ’と‘シ’のような類似文字群に属する文字をあいまいに書いた文字〔36 文字,



(a) ‘モ’(チ) (b) ‘オ’(RJ) (c) ‘タ’(RJ) (d) ‘セ’(RJ) (e) ‘ケ’(RJ)



(f) ‘ツ’(シ) (g) ‘イ’(RJ) (h) ‘ホ’(RJ) (i) ‘レ’(ル)

図 3. 10 誤読・棄却となった文字例

(‘ ’ : 正解, () : 認識結果)

図 3. 10 (f)〕.

(5) 書き方が乱雑

特徴点の対応づけは正しく行われたが、原文字パターンの変形が非常に大きい（伸ばすべき線分が伸びていなかったり、離すべきところが離れていない）うえに、細線化・折線近似による歪が加わって、棄却となった文字〔25文字, 図 3. 10 (g), (h)〕

(6) その他

副特徴点の数が不足となったもの〔3文字〕, 書き手の間違い（空白, 及び誤った字種の文字を書いたもの）〔2文字〕, 切り出し時に隣接文字の線分が枠内に入り込んだ文字〔4文字, 図 3. 10 (i)〕.

この分析結果において、明らかに識別部の不備に起因すると思われる文字は認められない。正しく特徴点が抽出できた文字では正読率が98.5%になることから、前処理部の改良により本方式の有効性はさらに確かなものになると思われる。

3. 6. むすび

本研究は人間の判読可能範囲を主成分軸上で直接計測し、文字認識系にその結果を組み入れ、人間と似た判読特性をもつシステムを作成するという構想の下に行われた研究であり、以下のことが明確になった。

- (1) 特徴点座標を特徴量として学習サンプルを主成分分析した結果、低次の主成分は各ストローク間の位置変動、回転変動などの典型的な字形変動を表し、高次の主成分は筆跡の微妙なゆらぎを表していることがわかった。
- (2) 一般には、各主成分軸上の正負方向判読限界値 $|U_i^+|$, $|U_i^-|$ は互いに大きく異なるうえに、各主成分ごとにもかなり異なるため、人間の心理的な識別関数は従来の2次識別関数と著しく異なる。この結果は従来の2次識別関数を機械的にパターン認識に適用する上で生じる重大な問題点を指摘すると共に、その一解決法を与える基礎資料となると考えられる。
- (3) 本章で提案した心理的距離は字形変動に対してより柔軟に対処でき、識別部が原因で誤読・棄却する文字は極めて少くなった。その結果、正しく特徴点抽出された文字に対しては人間の識別率 (99.1%) にかなり近い結果 (98.5%) を得た。
- (4) 本認識システムは2次判定を行っていないが、未学習文字による認識結果は正読率 97.2%, 誤読率 1.3%, 棄却率 1.5% であり、自由手書き文字データベース (ETL1)

に対する読取率としては最高値である。

又、今後の課題として、次のことが残されている。

(1) 類似文字の識別のように、特徴量という数値の微妙な比較の問題となると、学習文字を統計的に分析して行う判別分析的手法の方が人間よりもよい識別結果を与えるという実験結果⁽¹¹⁾が得られているので、これを2次判定に用いる。

(2) 特徴点の抽出精度を向上させるために、2値化、細線化、折線近似を改良する。特に細線化処理では、歪みの少ない処理方式を考えることのほかに、細線化処理後の図形を整形する試みもなされており⁽¹²⁾、その成果を組み込む。

(3) 提案した変形文字の発生法を発展させ、幅をもったストロークの表現を可能にすることなどがある。

なお、この研究の一部は文部省科研費（一般研究（B））の補助に基づくものであり、実験には三重大学計算センターを利用した。

第4章 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

あらまし

本章では、手書き漢字・ひらがな等の文字のオフライン認識に有効なボトムアップ式の特徴抽出方法とその特徴量に適した識別関数を開発し、その有効性を実験により検証する。本手法は輪郭線解析法をパターン類似度法との融合化の方向へ発展させ、高精度に読み取ることを目的としたものであり、その特徴は①単純な特徴抽出アルゴリズムであり、ハードウェア化しやすいこと、②相違度計算には従来のベイズ識別関数（2次識別関数）の問題点を解決した”疑似ベイズ識別関数”を用いていること、③ある意味ではもっとも認識が困難とされているひらがなの正読率が非常に高いこと、などである。本認識方式により文字画像を 4×4 の小領域に分割して、各小領域の方向指数のヒストグラム（64次元）を特徴ベクトルとして電総研作成の手書き教育漢字データベース（927字種；ETL8）の認識実験を行った結果、正読率99.5%（高品質文字）、96.3%（低品質文字）が得られた。

4. 1 はじめに

電子計算機による日本語情報処理に関する興味は近年急速に高まりつつあり、漢字入力的高速化・省力化は重要な問題となってきた。この問題を根本的に解決する方法として、OCRでの手書き漢字認識は有力な方法であり、この分野の研究が活発化し、多くの研究成果が発表されている⁽¹⁾。

従来の認識方法は、パターン類似度法⁽²⁾と構造解析法⁽³⁾に大別できる。パターン類似度法は図形の濃度値パターンを特徴量としており、生物の視覚系に類似している点や並列化により高速処理が可能な点などの長所を有している。他方、構造解析法は幾何学のおよび位相的特徴を抽出してから認識するために、局所的位置変動などのような字形変動に強い特徴を持っている。手書き漢字のように字種数も多く、複雑な線分構成となっている字種集合を認識する手法が、備えるべき条件としては、(1)高速認識処理の可能性を有する並列処理が可能であること、(2)文字線分の局所的位置変動に対して鈍感な特徴抽出機構を持つこと、(3)特徴抽出に要する計算量が少ないこと、(4)標準パターンの自動作成が可能であることなどが考えられる。

これらの条件は実用的な文字認識装置を作る観点からだけでなく、人間の視覚情報処理を探究する観点からも興味深い。例えば、人間が文章を斜め読みするときのように高速で文字を読む場合を考えると、特徴抽出の方法は極めて高速であり、かなり単純な方法であると考えられ、標準パターンとの複雑な対応づけを丁寧に行っている

とは考えにくい。特徴抽出において対応づけを行っているのはあいまいな文字、不明確な文字などに対してだけであると思われる。このことは人間がこれらの文字を読み取る時に、その速度を落して文字をみつめている経験から推察される。このような経験から筆者らは、視覚神経系の特徴抽出法を参考にし、文字領域の輪郭線に関する情報が、文字認識にとっても有効であると考え、ここに示す特徴抽出法を構成した。

本認識方式は、輪郭線解析法をパターン類似度法との融合化の方向へ発展させ、高精度に読み取ることを目的としたものであり、その特徴は①単純な特徴抽出アルゴリズムであり、ハードウェア化しやすいこと。②相違度計算には従来のベイズ識別関数（2次識別関数）の問題点を解決した”疑似ベイズ識別関数”を用いていること。③従来の方法では意外に認識が困難なひらがなの正読率が漢字の正読率以上に高いこと。などである。本認識方式により文字画像を4×4の小領域に分割して、各小領域の方向指数のヒストグラム（64次元）を特徴ベクトルとして電総研作成の手書き教育漢字データベース（927字種；ETL8）の認識実験を行った結果、正読率99.5%（高品質文字）、96.3%（低品質文字）が得られた。

4. 2 加重方向指数ヒストグラム法

4. 2. 1 加重方向指数ヒストグラム法

人間の視覚特性からヒントを得た文字認識方式は従来から数多く提案されている。例えば、飯島のボケ変換は網膜にとらえられる図形から考え出されたものであり⁽²⁾、文字線分の輪郭線の特徴抽出する認識方法^{(3)・(4)}は網膜内の視神経の生理現象に対応している。

筆者らは、上の2つの例は従来別々の方向に発展してきているが、これらは統合されるのが望ましいと考えている。統合された情報処理系が生体の生理的現象に即しており、そういう情報処理機構が高精度な文字認識処理系を構成すると考えている。そこでこれらの2つの潮流を合流させたモデルを次に提案する。この情報処理系の特色は、(1)空間的方向的ボケ変換を含むことにより文字線分の局所的な位置変動を吸収できること、(2)文字線分の輪郭線を抽出することにより線素構造を把握できること、(3)人間の視覚情報処理系を数理的に模倣していること、すなわち、情報変換・圧縮が並列処理で行われており、装置化した時に高速処理が可能であること、である。

[加重方向指数ヒストグラム]

文字の書かれている2値画像を $F = \{ f_{ij} \mid 1 \leq i \leq M, 1 \leq j \leq N \}$ とし、それから作成された8連結輪郭線方向指数画像(3次元2値画像)を $\{ G_l : \text{方向番号} (1 \leq l \leq 16 \text{ (最終的には、4方向にする)}) \}$ 、方向指数ヒストグラム画像を H とするとそれらの関係は次の

ようになる.

$$G' = \{ g'_{ij} \mid 1 \leq i \leq M, 1 \leq j \leq N \} \quad (4.1)$$

$$\text{但し, } g'_{ij} = \begin{cases} 1 \cdots f_{ij}=1 \text{かつ } f_{ij} \text{の方向指数} \\ \quad \text{が } l \text{ のとき} \\ 0 \cdots \text{それ以外の場合} \end{cases} \quad (4.2)$$

$$H = \{ h'_q \mid 1 \leq l \leq 16, 1 \leq q \leq M \times N \} \quad (4.3)$$

$$\text{但し, } h'_q = \sum_{(i,j) \in R_q} a_{ij} g'_{ij} \quad (4.4)$$

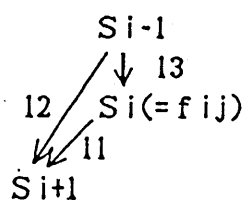
ここで R_q は全画像を P 個 ($P < M \times N$) に区分した時の q 番目の小領域を意味し, $\{ a_{ij} \}$ は空間フィルタの重み係数である. 従来の研究の多くは, この重み係数がすべて 1 である場合であるが, 本研究ではボケ変換理論⁽²⁾を適用し, 重み係数をガウス形分布関数とした. このボケ変換により空間的な情報圧縮を行い, 標本点数を少なくすると同時に文字線分の位置変動に対する吸収作用を実現させている.

画素 $f_{ij}(=S_i)$ の方向指数は文字輪郭部を左回り (穴部は右回り) に 8 連結の輪郭追跡を行い, 得られた輪郭点系列の 1 つ後の要素 (S_{i+1}) の位置する場所の番号 (図 4.1(a)) によって 8 方向の方向指数を得る. この方向 l_i と f_{ij} の 1 つ前の要素 (S_{i-1}) の方向 l_{i-1} を図 4.1(b) のように平均化して f_{ij} の 16 方向の方向指数を得る.

16 方向の方向指数を 8 方向に圧縮するには, 式 (4.5) を用い, 8 方向の方向指数画像群を得る. ここでの重み係数 (方向フィルタ) は各方向の成分が等しい重みで伝達される値が望ましい. そこで本研究では, $l = l^*$ のときには $b_l = 1.0$ とし, $l = l^* - 1, l^* + 1$ のときには $b_l = 0.5$ とした.

7	5	3
9	f_{ij}	1
11	13	15

(a)



(b)

図4. 1 画素 f_{ij} の方向指数

$$x^{l*}q = \sum_{l=1*-1, 1*, 1*+1} b_l h^{l*}q \quad (4.5)$$

但し, b_l : 重み係数

$$1* = \{1, 3, 5, 7, 9, 11, 13, 15\}$$

輪郭線の方法指数において, 互に反対方向の向きを同一視しても文字画像のような線状図形の場合には大きな差は生じないことが予備実験の結果から明らかとなったので⁽⁵⁾, 本研究では $1* = \{1, 3, 5, 7\}$ の方向に集約された4次元のベクトル $X = \{x^{l*}q\}$ を以後特徴ベクトルとする.

4. 2. 2 3次元的解釈

4. 2. 1 で述べた加重方向ヒストグラム法を3次元の特徴抽出法として解釈すると以下のようなになる(図4.2).

2値画像 $F = \{f_{ij}\}$ を左端に位置させ, 3次元の方向指数画像 $\{G^l \mid l=1, 2, \dots, 16\}$ をその右側に考える. すなわち, 1つの方向指数につき1枚の2次元画像 G^l を方向指数の順に考える. 方向指数

画像 $\{G'\}$ は3次元の2値画像であり、画像 F の輪郭点 f_{ij} の方向指数 l に対応する画素 g'_{ij} が1で、他は0である。別の言い方をすれば、輪郭線が通る場所においては、輪郭点 f_{ij} に対応する1個の方向指数画素 $\{g'_{ij}\}$ のうち1つの画素が1となり、他は0となる。また、輪郭線が通らない場所では、その画素に対応する方向指数画素 $\{g'_{ij}\}$ はすべて0となる。

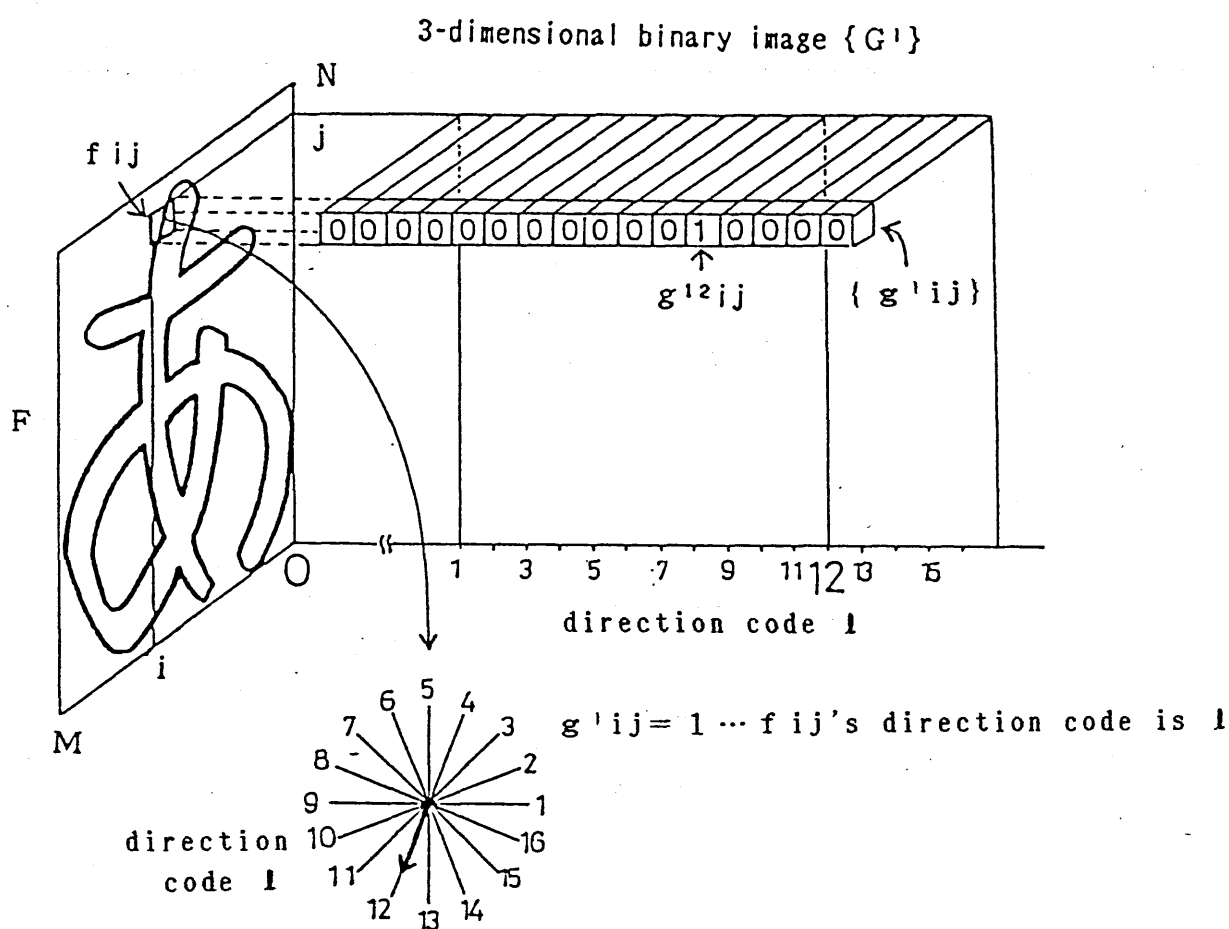


図4. 2 方向指数画像 $\{G'\}$

式 (4.4) は文字線分の空間的拡がりに対して処理される空間情報圧縮であり、式 (4.5) は線分の方方向的拡がりに対して処理される方

向情報圧縮である。この2つの圧縮操作は式(4.4), 式(4.5)では2段階で行っているが, 方向指数画像 $\{G^1\} = \{g^1_{ij}\}$ より1段階で特徴ベクトル $X = \{x^1 \cdot q\}$ を求めることができ, その場合に用いられる重み係数は3次元的拡がりを持つ3次元フィルタとなる(式(4.6))。

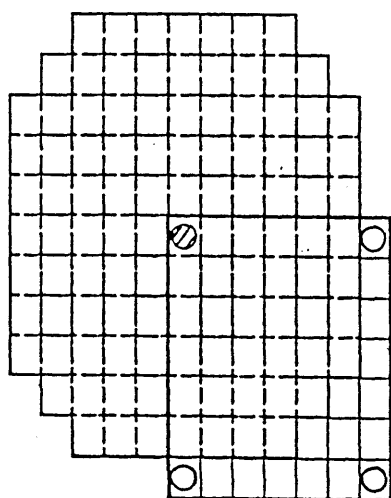
$$x^1 \cdot q = \sum_{l, i, j} c^1_{lij} g^1_{lij} \quad (4.6)$$

但し, c^1_{lij} : 重み係数(3次元フィルタ)

4. 2. 3 実現化

4. 2. 2で述べたように, 文字画像を一気に少数の標本点数に圧縮することも可能であるが, 重み係数の乗算回数が多くなるため, ここでは2段階で情報の圧縮を行った。まず第1段階では 7×7 の小領域に分け, その小領域内では重み係数はすべて1として, 方向指数のヒストグラムを作成する。そして第2段階で, 標本点数を 2×2 , 4×4 個としてガウス形重み係数により情報圧縮を行う。このときのそれぞれの空間フィルタの中心位置とフィルタの重み係数を図4.3, 図4.4に示す。これらの重み係数は, 2次元ガウス分布に基づいており, その分散は各小領域のヒストグラムが特徴ベクトルに対して寄与率がほぼ一定となるように定めた。このためサンプリング間隔が大きくなるにつれ分散は大きくなる(2×2 標本点で分散7.38, 4×4 標本点で分散0.81)。このようなフィルタの重なりを行うことにより, 文字線分の位置が近隣の小領域に”位置ずれ”しても, 特徴量の変動は, 重なりのない場合に比べ, 少なくなるように配慮した。

Weighted filter



Histogram of
direction code

○: Center of filter

(a) Center position of filter

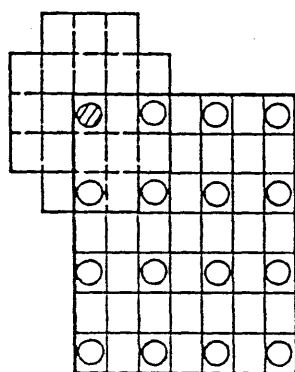
○.093	.087	.071	.051	.031	.017	.0
.087	.081	.066	.047	.029	.016	.0
.071	.066	.054	.039	.024	.013	.0
.051	.047	.039	.028	.017	.009	.0
.031	.029	.024	.017	.011	.0	.0
.017	.016	.013	.009	.0	.0	.0
.0	.0	.0	.0	.0	.0	.0

○: Center of filter

(b) Weight function

図4. 3 標本点数 2×2 の場合の空間的加重フィルタ

Weighted filter



Histogram of
direction code

○: Center of filter

(a) Center position of filter

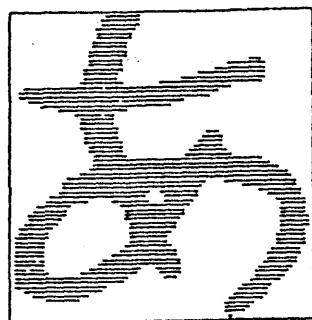
.0	.009	.017	.009	.0
.009	.057	.105	.057	.009
.017	.105	○.194	.105	.017
.009	.057	.105	.057	.009
.0	.009	.017	.009	.0

○: Center of filter

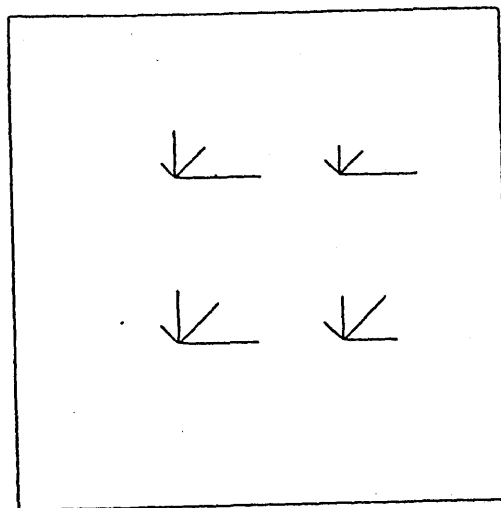
(b) Weight function

図4. 4 標本点数 4×4 の場合の空間的加重フィルタ

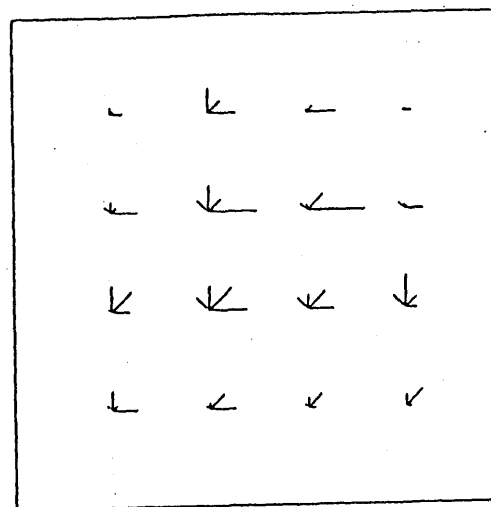
実際の手書き漢字にこの特徴抽出法を用いて抽出した特徴ベクトルを図4.5に示す。(a)が観測した2値図形であり、(b)が2x2で特徴抽出した場合、(c)が4x4で特徴抽出した例であり、各フィルタの位置に方向指数ヒストグラムを4方向の線分の長さで表現した図形である。



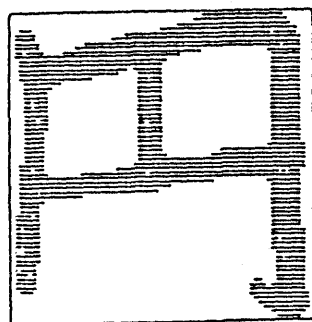
(a) Binary pattern



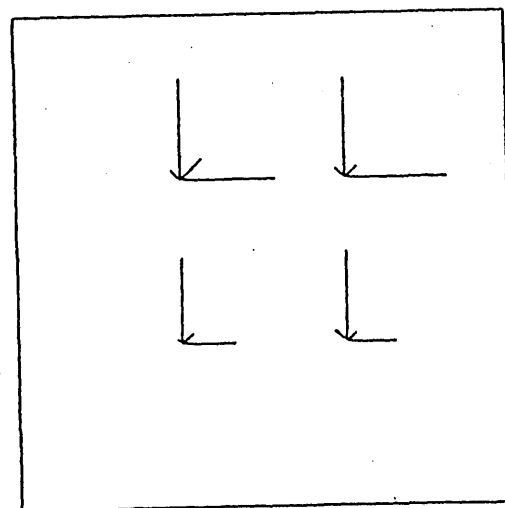
(b) 2x2x4 vector



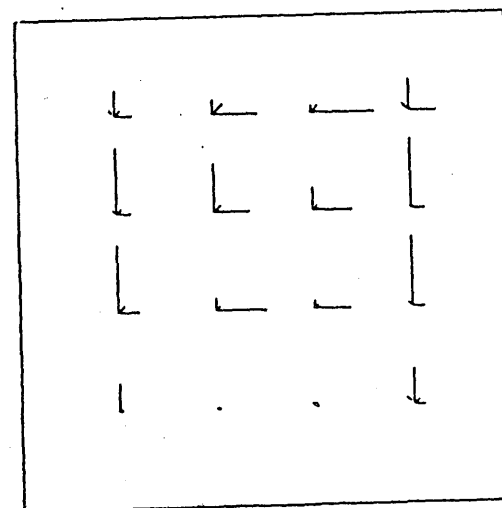
(c) 4x4x4 vector



(d) Binary pattern



(e) 2x2x4 vector



(f) 4x4x4 vector

図 4. 5 特徴ベクトルの図表現

4. 3 識別関数

4. 3. 1 従来のベイズ識別関数の問題点

従来から種々のパターン認識の分野で使われているベイズ識別関数（2次識別関数）は式（4.7）で示され，対象が正規分布に従い，特徴ベクトルの平均，共分散行列が既知の場合，最適識別関数となる。

$$g^1(x) = (x - {}_1\mu)^t {}_1\Sigma^{-1} (x - {}_1\mu) + \ln |{}_1\Sigma| \quad (4.7)$$

但し， x ：入力文字の特徴ベクトル（ n 次元）

${}_1\mu$ ：字種1の平均特徴ベクトル

${}_1\Sigma$ ：字種1の共分散行列

上式を手書き漢字認識に適用した場合の問題点を以下に述べる。

（1）漢字は文字構造が複雑で類似文字が多いので，英数字，カタカナ，ひらがなを認識する場合に比べ，特徴ベクトルの次元数 n を大きくしなければ，高精度認識が期待できない。 $g^1(x)$ の計算には， $O(n^2)$ の計算量と共分散行列のための記憶容量が必要となる。

（2） $g^1(x)$ を計算するのに，共分散行列の逆行列を求める必要がある。しかし，学習サンプル数 m が特徴ベクトルの次元数 n に比べ少ない時には，共分散行列は一次独立でなくなり，行列式の値が求められない場合が生じる。このため，共分散行列の固有値・固有ベクトル $\{\lambda_i, \phi_i \mid i=1, \dots, n\}$ を求め，式（4.8）の $g_{i1}(x)$ により相違度を計算している⁽⁶⁾。

$$g_{i1}(x) = \sum_{i=1}^k \frac{\{\Phi_{i1}(x - \mu)\}^2}{\lambda_i} + \ln \prod_{i=1}^k \lambda_i \quad (4.8)$$

但し, λ_i : 字種 1 の共分散行列 Σ の i 番目の固有値

Φ_{i1} : 字種 1 の共分散行列 Σ の i 番目の

固有ベクトル

k : 学習サンプル数により適当に定められる

整数 ($1 \leq k \leq m, n$)

式 (4.8) の固有ベクトルは高次になるほど推定誤差が大きくなることは知られており⁽⁷⁾, 認識率の最高値を示す整数 k は通常実験的に決められている。そして, 第 k 固有ベクトル以上の高次の固有ベクトルは無視されている。しかし, 漢字の認識には類似文字が存在するためかなり高次の固有ベクトルまで認識に必要と考えられる。

(3) 高次の固有ベクトルと入力文字の特徴ベクトルは一般に直交性が高いので, 内積値は結果として 0 に近い値となる。しかし, 浮動小数点で演算する場合には演算過程で, 各ベクトル要素間の積の絶対値が大きくなることがあるので, 桁落ちが生じやすい。そのため, 内積値を計算機で精度よく求めるのが難しい。

4. 3. 2 疑似ベイズ識別関数

前節で述べた問題点 (2), (3) を解決するために, 先に筆者らが提案した "疑似ベイズ識別関数 (Modified Bayes Discriminant Function : MBDF)" を適用した (式 (4.9))⁽⁸⁾。

$$\begin{aligned}
g_2^{-1}(x) = & \sum_{i=1}^k \frac{\{\Phi_i^t(x - \mu)\}^2}{\lambda_i} \\
& + \sum_{i=k+1}^n \frac{\{\Phi_i^t(x - \mu)\}^2}{\lambda_{k+1}} \\
& + \ln \left(\prod_{i=1}^k \lambda_i \cdot \prod_{i=k+1}^n \lambda_{k+1} \right) \quad (4.9)
\end{aligned}$$

ここで,

$$\sum_{i=1}^n \{\Phi_i^t(x - \mu)\}^2 = \|x - \mu\|^2 \quad (4.10)$$

を用いると、次のようになり、内積計算はk回ですむことがわかる。

$$\begin{aligned}
g_2^{-1}(x) = & \frac{1}{\lambda_{k+1}} \{ \|x - \mu\|^2 \\
& - \sum_{i=1}^k \left(1 - \frac{\lambda_{k+1}}{\lambda_i} \right) (\Phi_i^t(x - \mu))^2 \} \\
& + \ln \left(\prod_{i=1}^k \lambda_i \cdot \prod_{i=k+1}^n \lambda_{k+1} \right) \quad (4.11)
\end{aligned}$$

式(4.9)は式(4.8)において無視した高次の固有ベクトルを考慮に入れたものであり、高次の固有値はその絶対値が小さく、推定誤差・計算誤差が相対的に大きくなり、真の値を求めることは困難なので、 λ_{k+1} を $\lambda_i (i=k+1, \dots, n)$ の近似値として用いている。すなわち、固有ベクトルの分散を観測値よりも大きめに推定し、評価値を計算している。

高次の固有ベクトルを考慮した利点は、手書き漢字は複雑な変動をしており、固有値の誤差が少なくなるような低次の固有ベクトルだけでは識別に必要な情報は把握できないと考えられるからである。

すなわち、字種 1 の k 番目までの固有ベクトルで作られる部分空間に入力文字の特徴ベクトルが射影されない成分の大きさについても考慮しているために、 $g_1'(x)$ より $g_2'(x)$ の方が $g'(x)$ に近い値が求まると考えられる。

実際の手書き文字における自カテゴリー（事）の標準パターンと他カテゴリー（尊）の標準パターンの各固有ベクトルごとの相違度成分の分布を図 4.6 に示す。第 1, 4 固有ベクトルでは自カテゴリーの相違度の方が大きいですが、高次の固有ベクトルではそのようなことは無い。このような分布はほとんどの文字で見られ、手書き漢字では高次の固有ベクトルの相違度成分が識別に関して重要な情報であることを示している。

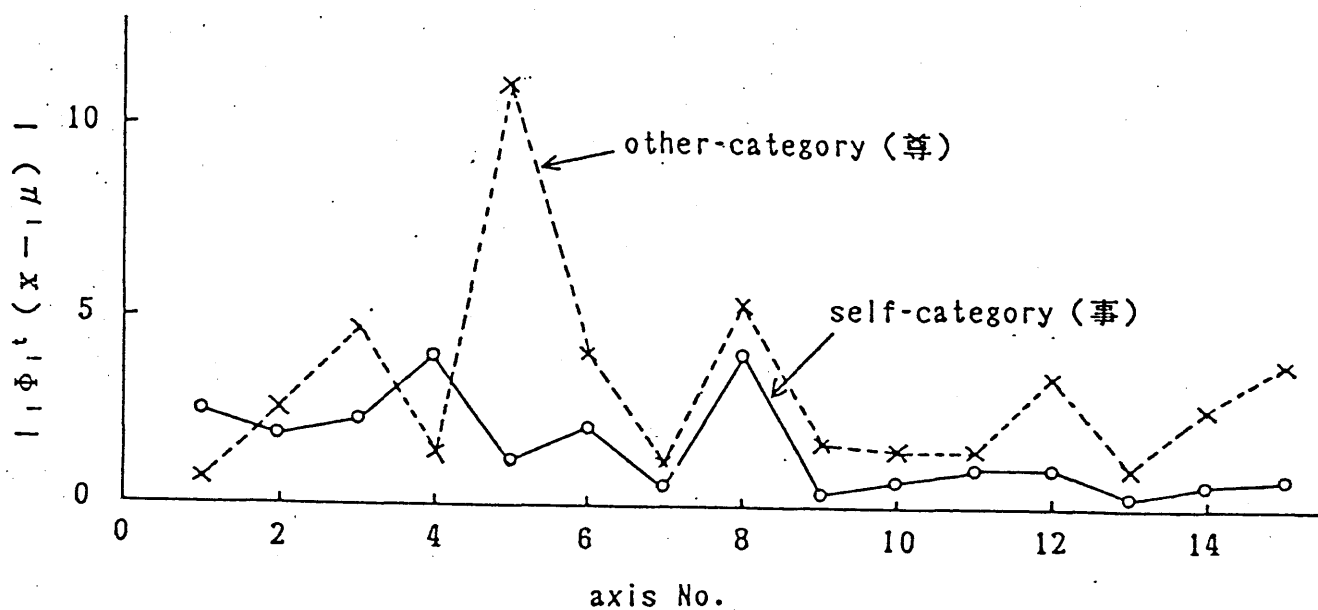


図 4. 6 自カテゴリーと他カテゴリーの相違度成分

4. 4 手書き漢字・ひらがなの識別実験

4. 4. 1 実験に使用したデータ

電総研作成の手書き教育漢字データベースETL8の第7～32巻（130文字／字種）を学習文字，第1巻と第6巻（各5文字／字種）を未学習文字として認識実験に用いた文献（8），（9）などより，第1巻は高品質文字の集合，第6巻は低品質文字の集合と考えられる。また，認識対象字種数は，漢字881字種とひらがな46字種の混在で927字種である。

特徴量として， 2×2 の標本点において4方向のヒストグラムを用いた場合（16次元）と 4×4 の標本点において4方向のヒストグラムを用いた場合（64次元）の2組を用意した。

4. 4. 2 識別関数の比較

文字認識の分野でよく使用されている距離関数として，マハラノビス距離がある。この距離に関して3. 2の式(4.11)に対応する式を求めると⁽⁸⁾，

$$g_3^{-1}(x) = \frac{1}{\lambda_{k+1}} \{ \|x - \mu\|^2 - \sum_{i=1}^k \left(1 - \frac{\lambda_{k+1}}{\lambda_i}\right) (x - \mu, \Phi_i)^2 \} \quad (4.12)$$

となる。この式を”疑似マハラノビス距離 (Modified Mahalanobis Distance:MMD)”と呼ぶこととすると、次式の関係式が成立する。

$$g_2^1(x) = g_3^1(x) + \ln\left(\prod_{i=1}^k \lambda_i \cdot \prod_{i=k+1}^n \lambda_{k+1}\right) \quad (4.13)$$

MBDFとMMDの累積分類率の比較を図4.7に示す。MBDFの方が常にMMDより高い分類率を示しており、固有値の積の項を含むMBDFの方が最適識別関数のよい近似となっている。

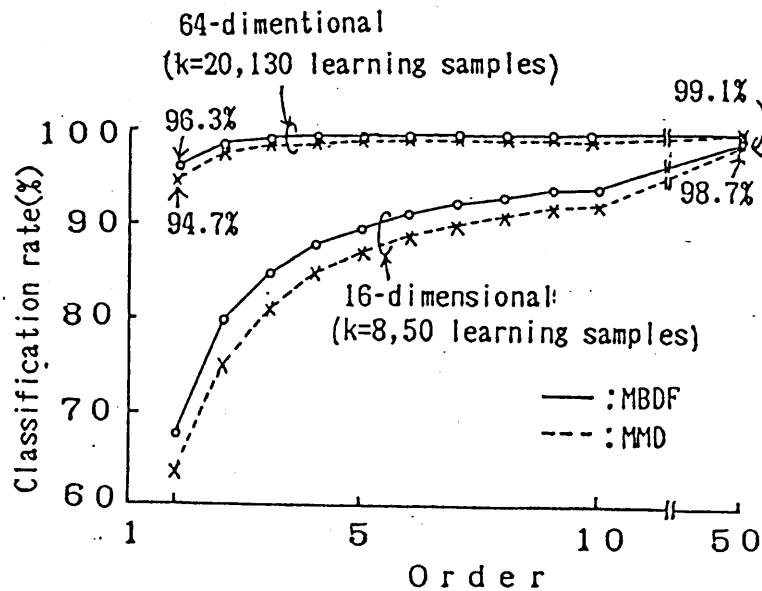


図4.7 MBDFとMMDの比較 (第6巻)

4.4.3 学習の効果

学習文字数を増加させると、平均、分散の推定誤差が少なくなり、字形変動が吸収できるので、分類率は上昇すると考えられる。図4.8に50文字/字種と130文字/字種学習の場合の累積分類率を示

す。本識別方法は学習の効果が明白であり、学習文字数をさらに増加させれば、16次元の特徴量でより高い正読率が得られると思われる。

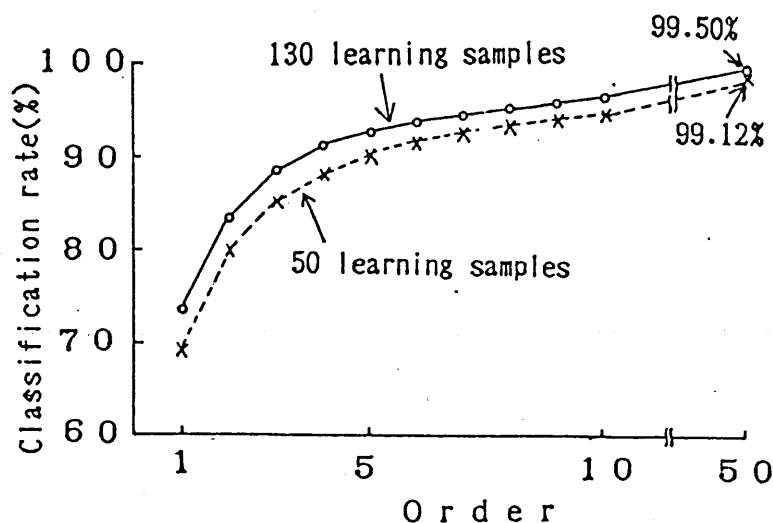


図4. 8 学習の効果 (16次元, MBDF, 第6巻)

4. 4. 4 累積分類率

16次元と64次元の特徴ベクトルについて、高品質文字と低品質文字に対する累積分類率を図4.9、図4.10に示す。また、テスト文字集合ごとの累積分類率を表4.1、4.2に示す。

64次元の特徴ベクトルでは、高品質文字については正解が1位にくる正読率は、漢字・ひらがなの混在状態で99.46%である。この場合のひらがなの各字種の平均正読率は100.0%であり、漢字の平均では99.43%とひらがなよりやや低い正読率である。また低品質の文字についてもこの傾向は同じであり、2.6%ひらがなの正読率がよく、従来の認識方法では意外に認識が困難なひらがなの正読率が漢字の正読率以上に高い。

次に、16次元の特徴ベクトルでは、高品質文字については50位累積分類率で99.95%となり、低品質文字でも99.48%となり、大分類の手法として十分な能力であると考えられる。

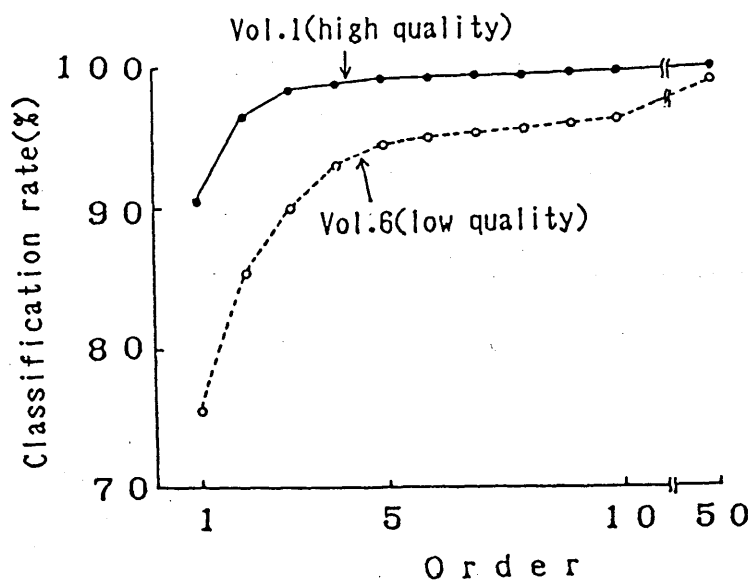


図4. 9 16次元特徴ベクトルの累積分類率 (MBDF, k=10)

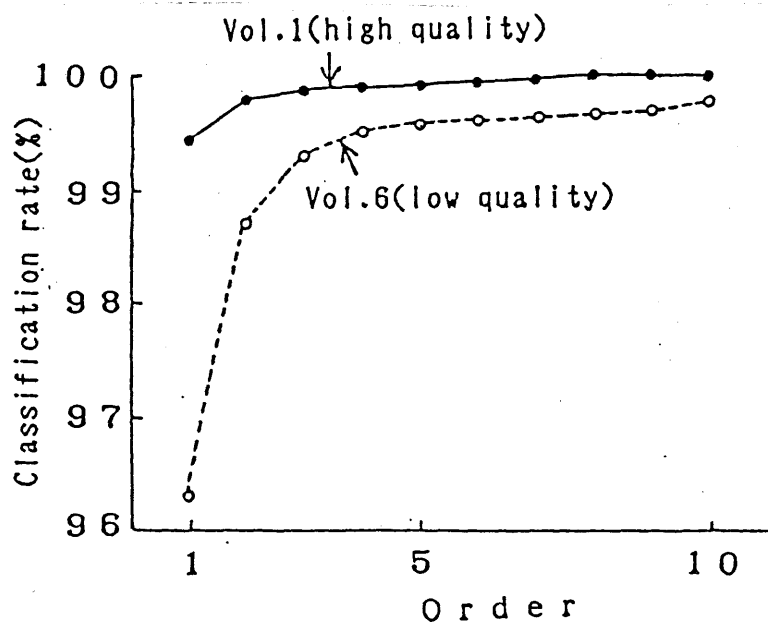


図4. 10 64次元特徴ベクトルの累積分類率 (MBDF, k=10)

表 4. 1 64次元特徴ベクトルにおける累積分類率(%)

テスト 文字集合	第1巻(高品質文字)		第6巻(低品質文字)	
	正読率	5位累積分類率	正読率	5位累積分類率
漢字				
ひらがな	99.46	99.91	96.25	99.55
ひらがな	100.00	100.00	98.70	99.57
漢字	99.43	99.91	96.12	99.55

表 4. 2 16次元特徴ベクトルにおける累積分類率(%)

テスト 文字集合	第1巻(高品質文字)		第6巻(低品質文字)	
	正読率	50位累積分類率	正読率	50位累積分類率
漢字				
ひらがな	91.37	99.95	75.68	99.48
ひらがな	99.13	100.00	92.61	99.57
漢字	90.96	99.95	74.81	99.48

4. 4. 5 誤読文字

64次元の特徴ベクトルを用いた時に10位までに正解の字種が入らなかった文字の例を図4.11に示す。光電変換時雑音により文字部分の切り出しが、正しく出来なかった文字の例を同図(a)に、また2値化のしきい値が不適当だったために文字線分のかすれや切れが原因と考えられる文字例を同図(b)に、文字の特定の線分が異常に長い(過長ストローク)文字の例を同図(c)に、紙面上の汚れを含む文

字の例を同図(d)に、筆記した文字のプロポーシヨンが著しく悪い文字の例を同図(e)に、続け文字の例を同図(f)に示す。

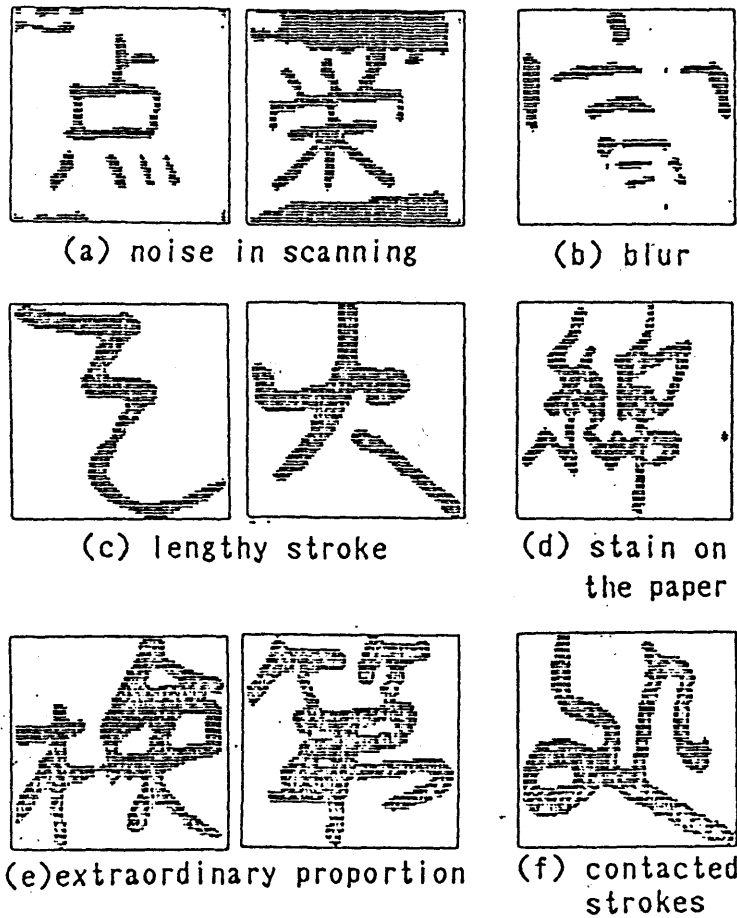


図4. 1 1 誤読文字例

4. 4 . 6 処理時間

低分解能と高分解能における汎用計算機 (FACOM M200) での1文字当りの特徴抽出・認識に要する計算時間は217.3ms/文字 (16次元), 734.5ms/文字 (64次元)であり, その内訳を表4. 3に示す。

16次元, 64次元の特徴ベクトルにおける全体の処理時間における相違度計算の割合は, それぞれ約70%, 約90%である. 相違度計算は内積計算を主体としているので, 専用プロセッサまたは並列処理による処理時間の高速化は容易に実現できる. またこの表では64次元の特徴ベクトルを使用する場合に大分類を行っていないが, 例えば候補字種数を50字種に絞れば, 約20分の1の処理時間となる.

表4. 3 処理時間 (mS/文字)

特徴ベクトル の次元 計算内容	16次元 (大分類向き) k=10	64次元 (詳細識別向き) k=15
特徴抽出	66.0	66.0
相違度計算	143.6	660.8
相違度のソート	7.7	7.7
合 計	217.3	734.5

4. 5 むすび

本研究では、手書き漢字・ひらがな認識に適した認識方式として、加重方向指数ヒストグラム法を提案し、その有効性を手書き教育漢字データベース E T L 8 (956 字種) を用いて大型計算機によるシミュレーション実験を行い、実用的な認識方式であることを検証した。

本認識方式の成果を要約すると次のようになる。

- (1) 直線を主体とする漢字と、曲線を主体とするひらがなを同一の認識アルゴリズム (64 次元の特徴ベクトル) で認識させた結果、高品質のひらがなで 100.0%, 漢字で 99.4%, 低品質のひらがなで 98.7%, 漢字で 96.1% の正読率が得られた。
- (2) 領域の分割数を 2×2 に低下させた時にも、混在認識において、ひらがな 99.1%, 漢字 91.0% の正読率と、ひらがな 100.0%, 99.95% の 50 位累積分類率が得られており、大分類の手法としても有効な認識方式であることが確認できた。
- (3) 処理時間は汎用計算機 (FACOM M-200) によるシミュレーションで 17.3 ms / 秒 (16 次元), 734.5 ms / 秒 (64 次元) であり、比較的高速な認識が行える。

又、今後の課題として、(1) クラスタリングなどの手法を利用してより高速な大分類の手法を検討すること、(2) 輪郭部の雑音に対して安定して特徴抽出する機構を開発すること、(3) 本認識

方式に適した棄却条件を見いだすことなどが残されている。

4・A. 1 輪郭追跡の方法について

文字画像の輪郭線の方角指数を求める場合に使用している輪郭追跡の方法について述べる。

輪郭追跡法は、種々の画像処理で用いられている方法であり、多数の方法が提案されている。本論文で使用している方法は横井らによって提案された方法⁽¹⁾であり、その追跡手順を第2章で定義した用語を用いて以下に説明する。

[輪郭追跡アルゴリズム (8 連結)]

まず、図形の任意の境界点 (0 要素に 4 隣接する 1 要素) を y_0 とし、その要素に 4 隣接する 0 要素を z_0 とする。つぎに、 y_0 の 8 近傍において、 z_0 から反時計回りに 1 要素を捜し、 y_0 の 8 近傍にある最初の 1 要素を y_1 とする。つぎに、 y_0 から反時計回りに y_1 の 8 近傍を捜し、最初の 1 要素を y_2 とする (図 4. A. 1)。以後同様にして、 y_3, y_4, \dots の順に決めてゆき、最初に、 $y_{m-1} = y_0, y_m = y_1$ となったとき、一本の輪郭線の追跡を終える。このとき、系列 $y_0, y_1, y_2, \dots, y_{m-1}$ を一本の輪郭線と呼び、 $y_0, y_1, y_2, \dots, y_{m-1}$ を輪郭線要素と呼ぶ。

[性質 1] 得られた 1 本の輪郭線要素はすべて 8 連結で連結しており、図形の外側では反時計回りに追跡され、図形の内側では時計回りに追跡される。

[性質 2] 図形のすべての境界線の中に図形内の任意の要素 x_0 が現

れる回数はその要素の連結数に等しい。すなわち、端点、分岐点、交点を除いた線幅1ビットの部分（連結点）は2回境界線の中に数えられる。

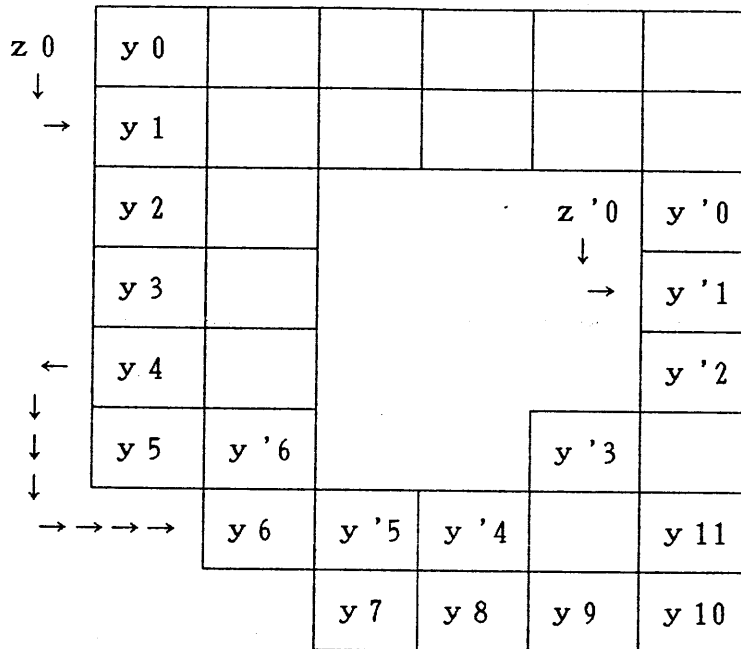


図4. A. 1 輪郭追跡の例

図4. A. 1の例では、左上からの横方法の走査（ラスタ走査）により、2つの輪郭線の出発点（ y_0 と y'_0 ）を見つけた例である。図形の外側では、ラスタ走査により初めて見つかる1要素 y_0 を出発点として輪郭線が追跡され、 y_1, y_2, y_3, \dots の順に輪郭要素が抽出され、図形を反時計回りに追跡する輪郭線が抽出される。そして、左下のコーナー（ y_5, y_6, y_7 ）では右下へ延びる8連結の連結性となり、右下のコーナー（ y_9, y_{10}, y_{11} ）では4連結の連結性となる。

一方、図形の内側も外側と同様に、 y'_0 を出発点として輪郭線が追跡され、 y'_1, y'_2, y'_3, \dots の順に輪郭要素が抽出され、穴を

時計回りに追跡する輪郭線が抽出される。この場合には、穴の角が斜め ($y'2$, $y'3$, $y'4$) であろうと、直角 ($y'5$, $y'6$) であろうと 8 連結の線が得られる。

また、要素 $y'0$, $y'1$, $y'2$ は外側の輪郭線にも内側の輪郭線にも含まれるので、2 回数えられ、その点の連結数と等しくなる。

4・A. 2 バイズ識別関数の固有値・固有ベクトル 表現（式（4.8））について

バイズ識別関数は一般に式（4.7）で与えられているが、

$$g_1(x) = (x - {}_1\mu)^t {}_1\Sigma^{-1} (x - {}_1\mu) + \ln | {}_1\Sigma | \quad (4.7)$$

但し、 x ：入力文字の特徴ベクトル（ n 次元）

${}_1\mu$ ：字種 1 の平均特徴ベクトル

${}_1\Sigma$ ：字種 1 の共分散行列

本研究では、4.3.1で述べた理由により、これを固有値・固有ベクトル表現した $g_1(x)$ （式（4.8））により、相違度を計算している。

$$g_1(x) = \sum_{i=1}^k \frac{\{ {}_1\Phi_i^t (x - {}_1\mu) \}^2}{{}_1\lambda_i} + \ln \prod_{i=1}^k {}_1\lambda_i \quad (4.8)$$

但し、 ${}_1\lambda_i$ ：共分散行列 ${}_1\Sigma$ の固有値

${}_1\Phi_i$ ：共分散行列 ${}_1\Sigma$ の固有ベクトル

k ：整数（ $1 \leq k \leq m, n$ ）

この導出過程を以下に示す。

(1) 共分散行列 Σ の固有値・固有ベクトル表現

行列の固有値の定義により,

$$\Sigma \Phi_i = \lambda_i \Phi_i \quad (4.A.1)$$

であるので, 共分散行列 Σ の n 個の固有ベクトル $\{\Phi_i \mid i=1, \dots, n\}$ をつぎの $n \times n$ 行列 Φ で表し,

$$\Phi = [\Phi_1, \Phi_2, \dots, \Phi_n]^t \quad (4.A.2)$$

共分散行列 Σ の n 個の固有値 $\{\lambda_i \mid i=1, \dots, n\}$ を対角要素に持った $n \times n$ 対角行列 Λ を

$$\Lambda = \begin{bmatrix} \lambda_1 & & 0 \\ & \lambda_2 & \\ & & \dots \\ 0 & & \lambda_n \end{bmatrix} \quad (4.A.3)$$

とすると, 式 (4.A.1) の関係は,

$$\Sigma \Phi = \Lambda \Phi \quad (4.A.4)$$

となる. ここで, 固有ベクトルの直交性より,

$$\Phi \Phi^t = I = \begin{bmatrix} 1 & & 0 \\ & 1 & \\ & & \dots \\ 0 & & 1 \end{bmatrix} \quad (4. A. 5)$$

という関係がある。

式 (4. A. 4) を書き直すと,

$${}_1\Sigma \begin{bmatrix} {}_1\Phi_1 \\ {}_1\Phi_2 \\ \vdots \\ {}_1\Phi_n \end{bmatrix} = \begin{bmatrix} {}_1\lambda_1 & & 0 \\ & {}_1\lambda_2 & \\ & & \dots \\ 0 & & {}_1\lambda_n \end{bmatrix} \begin{bmatrix} {}_1\Phi_1 \\ {}_1\Phi_2 \\ \vdots \\ {}_1\Phi_n \end{bmatrix} \quad (4. A. 6)$$

となり, Φ^t を両辺に右からかけると

$${}_1\Sigma \Phi \Phi^t = \Lambda \Phi \Phi^t \quad (4. A. 7)$$

となり, 式 (4. A. 5) を左辺に適用すると, ${}_1\Sigma I = {}_1\Sigma$ なので, 共分散行列 ${}_1\Sigma$ は次のように固有値・固有ベクトル表現できる。

$${}_1\Sigma = \begin{bmatrix} {}_1\lambda_1 & & 0 \\ & {}_1\lambda_2 & \\ & & \dots \\ 0 & & {}_1\lambda_n \end{bmatrix} \begin{bmatrix} {}_1\Phi_1 \\ {}_1\Phi_2 \\ \vdots \\ {}_1\Phi_n \end{bmatrix} [{}_1\Phi_1^t \dots {}_1\Phi_n^t] \quad (4. A. 8)$$

$$= \begin{bmatrix} {}_1\lambda_1 {}_1\Phi_1 \\ {}_1\lambda_2 {}_1\Phi_2 \\ \dots \\ {}_1\lambda_n {}_1\Phi_n \end{bmatrix} [{}_1\Phi_1^t \dots {}_1\Phi_n^t] \quad (4. A. 9)$$

$$= \sum_{i=1}^n {}_1\lambda_i {}_1\Phi_i {}_1\Phi_i^t \quad (4. A. 10)$$

(2) 共分散行列 ${}_1\Sigma$ の n 個の固有値を $\{{}_1\lambda_i \mid i=1, \dots, n\}$ とした
ときの逆行列 ${}_1\Sigma^{-1}$ の固有値

行列の固有値の定義により,

$${}_1\Sigma {}_1\Phi_i = {}_1\lambda_i {}_1\Phi_i$$

両辺を ${}_1\lambda_i$ で割って

$$\frac{1}{{}_1\lambda_i} {}_1\Sigma {}_1\Phi_i = {}_1\Phi_i \quad (4. A. 11)$$

両辺に左から, 逆行列 ${}_1\Sigma^{-1}$ をかけると

$$\frac{1}{{}_1\lambda_i} {}_1\Sigma^{-1} {}_1\Sigma {}_1\Phi_i = {}_1\Sigma^{-1} {}_1\Phi_i \quad (4. A. 12)$$

${}_1\Sigma^{-1} {}_1\Sigma = I$, $I {}_1\Phi_i = {}_1\Phi_i$ より

$$\frac{1}{{}_1\lambda_i} {}_1\Phi_i = {}_1\Sigma^{-1} {}_1\Phi_i \quad (4. A. 13)$$

したがって、 ${}_{1}\Sigma^{-1}$ の固有値は $1/{}_{1}\lambda_i$ であり、行列表現すると

$${}_{1}\Sigma^{-1}\Phi = \begin{bmatrix} 1/{}_{1}\lambda_1 & & 0 \\ & 1/{}_{1}\lambda_2 & \\ & & \dots \\ 0 & & & 1/{}_{1}\lambda_n \end{bmatrix} \Phi \quad (4.A.14)$$

となる。この式に式(4.A.10)を用いると、逆行列 ${}_{1}\Sigma^{-1}$ の固有値・固有ベクトル表現は

$${}_{1}\Sigma^{-1} = \sum_{i=1}^n \frac{1}{{}_{1}\lambda_i} {}_{1}\Phi_i {}_{1}\Phi_i^t \quad (4.A.15)$$

となる。

(3) ベイズ識別関数の固有値・固有ベクトル表現

式(4.A.15)をベイズ識別関数(式(4.7))に代入すると

$$g^1(\mathbf{x}) = (\mathbf{x} - {}_{1}\boldsymbol{\mu})^t \left(\sum_{i=1}^n \frac{1}{{}_{1}\lambda_i} {}_{1}\Phi_i {}_{1}\Phi_i^t \right) (\mathbf{x} - {}_{1}\boldsymbol{\mu}) + \ln |{}_{1}\Sigma| \quad (4.A.16)$$

ここで、式(4.A.8)より共分散行列の行列式 $|{}_{1}\Sigma|$ は

$$|{}_{1}\Sigma| = \begin{vmatrix} {}_{1}\lambda_1 & & 0 \\ & {}_{1}\lambda_2 & \\ & & \dots \\ 0 & & & {}_{1}\lambda_n \end{vmatrix} |{}_{1}\Phi\Phi^t| \quad (4.A.17)$$

であり、また、式(4.A.5)より、 $|\Phi\Phi^t| = 1$ なので、

$$\ln |\Sigma| = \ln \prod_{i=1}^n \lambda_i \quad (4.A.18)$$

したがって、

$$g^1(x) = \sum_{i=1}^n \frac{1}{\lambda_i} (x - \mu)^t \Phi_i \Phi_i^t (x - \mu) + \ln \prod_{i=1}^n \lambda_i \quad (4.A.19)$$

$$= \sum_{i=1}^n \frac{\{\Phi_i^t(x - \mu)\}^2}{\lambda_i} + \ln \prod_{i=1}^n \lambda_i \quad (4.A.20)$$

$$= \sum_{i=1}^k \frac{\{\Phi_i^t(x - \mu)\}^2}{\lambda_i} + \ln \prod_{i=1}^k \lambda_i$$

$$+ \sum_{i=k+1}^n \frac{\{\Phi_i^t(x - \mu)\}^2}{\lambda_i} + \ln \prod_{i=k+1}^n \lambda_i \quad (4.A.21)$$

ここで、 $i=k+1$ 以降の項の固有値 λ_i ($\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$) を0とすると式(4.8)となる。

第5章 筆記者に対して適応機能を持った自由手書き文字認識

あらまし

本章では、筆記者に対して適応機能を持った自由手書き文字認識方式について述べる。この方式は、入力される自由手書き文字をあらかじめ用意された汎用辞書で認識して、その文字を学習に使用し、筆記者の字形に適応する辞書を逐次作成していく方法であり、多人数の標準的な字形と異なる字形を書く人にとって、特に有効となる認識方式である。この方法を用いたOCRでは、筆記者に特有の書き癖や続け・省略を含んだ文字でも読み取れる可能性があり、実用性の高い方法である。本章では、平均と固有値・固有ベクトルを用いる認識方法における適応方法を3種類示し、加重方向指数ヒストグラム法を用いて、全ひらがな文字（46字種）を対象とした5人の特定筆記者で認識実験を行った。その結果、汎用辞書（電総研作成のETL4の100文字／字種で学習）で96.8%の平均正読率の文字データが、99.5%に向上し、この方式が効果的であることが実証された。

5. 1 はじめに

従来の多くの手書き文字認識の研究では、光学的文字読取装置（OCR）に不特定多数の筆記者の文字を読み取らせることを目的として、汎用性のある辞書を用いて筆記者の個性（書きぐせ）を吸収する種々の試みがなされてきた。しかし、筆記者の個性を完全に取り除くことは相当難しく、まだ多くの問題点を解決する必要があると考えられる。

一方、半導体技術の急激な進歩により、OCRが小型・安価となり、分散型オフィス機器として用いられることを考えると、使用者は特定の筆記者に限定され、パーソナル機器としての役割が重要になると考えられる。このような使用方法を考えると、不特定多数の筆記者の文字から作成された汎用性のある辞書より、特定の使用者の字形に合わせた辞書を用いたほうが利点は多いと考えられる。

この利点とは、筆記者の個性（書きぐせ）を学習できることであり、これにより続け字や省略のある字、傾斜字、異体字等も認識対象字体とすることが期待できる。また、従来認識の難しかった類似字種対^{(1)・(2)}（カタカナの”ソ”と”ン”など）についても、特定の筆記者の文字では、多数の筆記者の文字を集めた場合に比べ、字形のばらつきは少ないこと⁽³⁾等の事実から、高精度で認識できる可能性もあり、文字認識システムが普及するには重要な技術と考えられる。

別の見方をすれば、これらの研究分野は、人間の生理系で行われている適応学習機構のモデルを提案し、モデルの性能を実際の手書

き文字データで評価しているとも考えられ、認知科学の立場からも興味深い研究分野であると思われる。

そこで、近年筆記者の字形の個人差を積極的に辞書作成に利用して認識率を向上させようとする動きが、見られるようになった。

吉村ら⁽⁴⁾は、パターン整合法を用いて、特定筆記者の文字だけで作成した専用辞書を用い、他の筆記者の文字より作成した辞書による認識結果より正読率が良くなることを示した。また、内藤ら⁽⁵⁾は、ストローク密度特徴を用いて平均値だけを使用した距離関数により、個人性情報の利用方法に関する手書き漢字認識の研究を行った。そして、特定筆記者の平均特徴と不特定多数の筆記者の文字の平均特徴の線形和を認識に使うことを提案し、学習文字数が少ないとき、効果的であることを示した。

本章では、従来ほとんど研究されていなかった、固有値・固有ベクトルを用いた認識方法における個人性情報の活用方法を検討するために、筆者らが開発した加重方向指数ヒストグラム法を用いて、基礎的実験を行った。始めに、特定筆記者の文字だけを学習に用いた筆記者専用の辞書の効果とその問題点を述べ、次に、不特定多数の筆記者から作成した汎用辞書（平均ベクトル、固有値、固有ベクトル）を逐次、特定の筆記者の専用辞書に近づけていく3種類の方法（更新型辞書、修正型辞書、混合型辞書）を提案する。そして、各々の適応方法の特徴を実験的に比較検討するために、全ひらがな字種（46字種）を対象とした5人の特定筆記者で認識実験を行った。その結果、汎用辞書（電総研作成の文字データベースETL4の100文字／字種で学習）で96.8%の平均正読率の文字データが、適応型辞書で99.5%に向上し、この方式が効果的である

ことが実証された。

5. 2 専用辞書と汎用辞書

辞書を筆記者別に作成すると汎用辞書を使用した場合に比べ、認識率が向上することが知られている^{(4)・(5)}。ここでは、加重方向指数ヒストグラム法においても同様の性質があるかどうかを調査した。

5. 2. 1 特徴抽出と識別関数

(1) 特徴抽出

特徴抽出は、第4章で述べた加重方向指数ヒストグラム法⁽⁶⁾を用いた。すなわち、観測されたデジタル文字図形を2値化し、文字部分に外接する長方形を一定の大きさに拡大縮小するという位置・大きさの正規化（外接枠合わせ）を行い、文字部分の輪郭線を抽出し、各輪郭線素の方向指数を求める。次に、文字枠を7（ヨコ）×7（タテ）の小領域に区切り、小領域内の輪郭線素を方向指数ごとに計数し、4方向のヒストグラムを作成する。そして、そのヒストグラムをガウス型の重み付きフィルタを用いて、縦4点、横4点の標本点（4方向）で標本化し、これを字種識別のための特徴量（6次元の特徴ベクトル）として用いる。

この特徴量を用いた認識システムは、手書き漢字認識で、現在トップレベルの正読率を示しており、電総研作成の手書き教育漢字データベース（927字種；ETL8）で正読率99.5%（高品質文字）、96.3%（低品質文字）が得られている⁽⁷⁾。

(2) 識別関数

識別関数も、特徴抽出と同様に、第4章で述べた”疑似ベイズ識別関数 (Modified Bayesian Discriminant Function : MBDF)” (式 (5.1))⁽⁶⁾を使用した。

$$\begin{aligned}
 g^l(x) &= \sum_{i=1}^k \frac{(x - \mu_l, \Phi_i)^2}{\lambda_i} \\
 &\quad + \sum_{i=k+1}^n \frac{(x - \mu_l, \Phi_i)^2}{\lambda_{k+1}} \\
 &\quad + \ln \left(\prod_{i=1}^k \lambda_i \cdot \prod_{i=k+1}^n \lambda_{k+1} \right) \\
 &= \frac{1}{\lambda_{k+1}} \{ \|x - \mu_l\|^2 \\
 &\quad - \sum_{i=1}^k \left(1 - \frac{\lambda_{k+1}}{\lambda_i} \right) (x - \mu_l, \Phi_i)^2 \\
 &\quad + \ln \left(\prod_{i=1}^k \lambda_i \cdot \prod_{i=k+1}^n \lambda_{k+1} \right) \} \quad (5.1)
 \end{aligned}$$

但し、 x : 入力文字の特徴ベクトル (n 次元)

μ_l : 字種 l の平均特徴ベクトル

λ_i : 字種 l の共分散行列 Σ の i 番目の固有値

Φ_i : 字種 l の共分散行列 Σ の i 番目の固有ベクトル

k : 学習サンプル数 m と特徴ベクトルの次元

n により適当に定められる整数

($1 \leq k \leq m, n$)

この識別関数は、従来からパターン認識の分野で使われているベイズ識別関数の問題点⁽⁷⁾を解決した識別関数であり、第 k 主成分より高い主成分について固有値を一定としていることに特色があり、ベイズ識別関数に比べて計算量が少ないことや計算誤差が生じにくいという利点がある。

5. 2. 2 認識実験

(1) 使用したデータ

特定筆記者用の文字データとして、自由手書きひらがな46字種を対象に、三重大大学の学生・院生が1字種当り26文字、OCR用紙にシャープペンシルで筆記された文字を、16階調で光電変換して5人分用意した。観測された原データの大きさは、ほぼ 40×50 ドットの大きさであった。

不特定多数の筆記者の文字データとして、電子技術総合研究所作成の文字データベースETL4(16値原データ)を利用した。ETL4は自由手書きひらがな文字51字種を対象に作成されており、100人の筆記者(名古屋大学の学生)が1人、1文字/字種ずつ筆記した文字画像を収集したデータベースである。

特定筆記者用のデータのうち、偶数番目の文字10文字をテスト文字として用い、残る16文字を学習文字として、各筆記者について専用辞書を作成した。また、汎用辞書は、不特定多数の筆記者用のデータを100文字/字種用いて作成した。

(2) 認識結果

① 認識率の比較

専用辞書と汎用辞書により、5人の筆記者の文字をそれぞれ1字種当たり10文字、計460文字認識させた結果を図5.1に示す。

汎用辞書では、筆記者による正読率のバラツキが大きく、分散は専用辞書の約6倍以上となっている。また、平均正読率は96.8%で、専用辞書(99.0%)より2.2%低い値であり、全筆記者で専用辞書より低い正読率である。汎用辞書の誤読文字は、一部の字種で多くの文字が誤読されている場合が多く、筆記者の概念パターンと不特定多数の筆記者の概念パターンの平均が、一部の字種では異なっていると考えられる。この傾向は、筆者らが先に実験を行った濃度値の空間的分布状態を特徴量とする複合類似度法でも、確かめられている⁽⁸⁾。

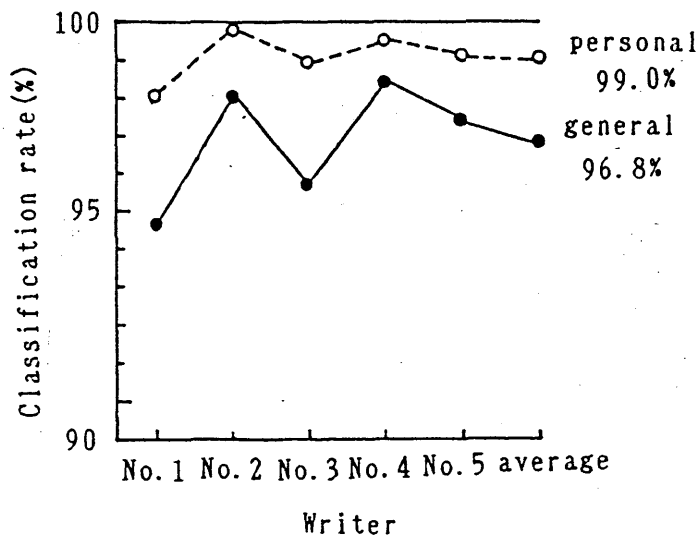


図5. 1 個人専用辞書と汎用辞書の比較

② 学習文字数と正読率

専用辞書の学習文字数と正読率の関係を図5.2に示す。学習文字数が少ない場合（5文字／字種）には、共分散行列の推定が正しく行えないために、92.8%と汎用辞書より低い正読率となるが、学習文字数が8文字以上となると、汎用辞書より高い正読率が得られる。

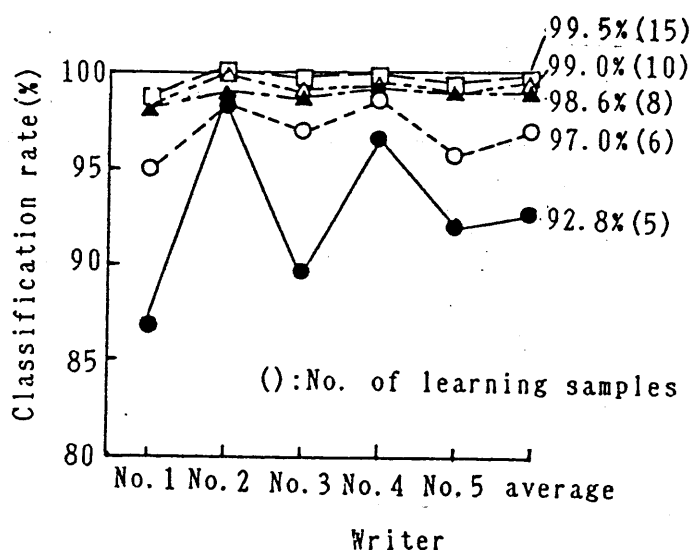


図5. 2 個人専用辞書における学習の効果

③ 誤読文字

汎用辞書では正読となる文字の中で、専用辞書で誤読となる文字の例を図5.3に示す。同一筆記者の他の文字と比較すると、筆記者が急に普段と異なる字形で筆記した文字が、誤読となっている。この原因は、専用辞書は各筆記者の少数の文字だけで学習しているため、少ない種類の字形変動しか学習されておらず、吸収できる変形が限定されているためと考えられる。










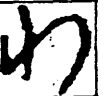
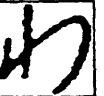
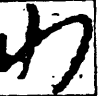







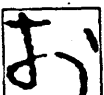




misrec- ognition	Examples of learning samples
	    
	    
	    
	    

図 5. 3 個人専用辞書の誤読文字例

5. 3 筆記者に適應する文字認識

筆記者に適應する文字認識とは、初めに用意されている辞書は、筆記者の字形とは異なる字形が用意されているが、筆記者が文字認識装置に文字を入力するにしたがい、次第に筆記者の字形を学習し、辞書が筆記者の字形に適應していく機能を持った文字認識方式をいう。

5. 3. 1 2種類の適應型辞書

(1) 辞書作成方法

適應型文字認識における辞書作成方法として、まず、次の2つを比較検討する。

①更新型辞書：汎用辞書作成用の文字と筆記して得られた特定筆記者の文字から、新たに辞書（平均特徴ベクトル、固有値・固有ベクトル）を作成する。

②修正型辞書：辞書の中の平均特徴ベクトルについては、特定筆記者の既に得られている文字より作成し、固有値・固有ベクトルについては、汎用辞書の固有値・固有ベクトルをそのまま利用する。

更新型辞書は、辞書を更新するのに、不特定多数の筆記者の字形

と特定筆記者の字形より作られた平均特徴ベクトルと、固有値・固有ベクトルを作成するもととなる特徴ベクトルの共分散行列を記憶しておく必要があるが、統計学的には自然な平均と固有値・固有ベクトルを利用する方法であり、一つの基準となる方法である。なお、吉村ら⁽⁴⁾は、この方法で学習してパターン整合法では、総学習文字数が同じときに、専用辞書が汎用辞書より高い正読率を示すことを報告した。

修正型辞書は、汎用辞書の平均特徴ベクトルの代わりに、既に得られている特定筆記者の文字から作った平均特徴ベクトルを用いる方法であり、平均だけが特定筆記者の字形を反映しているという不自然さがある。しかし、ここで、固有値・固有ベクトルを計算し直さないのは、学習文字数が少ない場合には、固有値・固有ベクトルが安定して計算されないからであり、そのようなものを使用するより、安定した多人数の文字より計算されたものを用いた方が、字種内の字形の変動を的確に吸収できると考えたからである。

(2) 認識実験

① 更新型辞書

汎用辞書に加えた特定筆記者の文字数と正読率の関係を図5.4に示す。加えた文字数が0文字の場合は、汎用辞書による正読率(96.8%)を示しており、文字を追加するにしたがって、正読率が向上していくことが確かめられた。

また、5人中4人の筆記者では、専用辞書(10文字/字種)より高い正読率が得られ、5人の平均正読率は99.3%(専用辞書で99.0%)となる。これは、特定筆記者の文字以外に、多数の

筆記者の文字も学習しているので、字種内における多様な変形が吸収できたためであると考えられる。

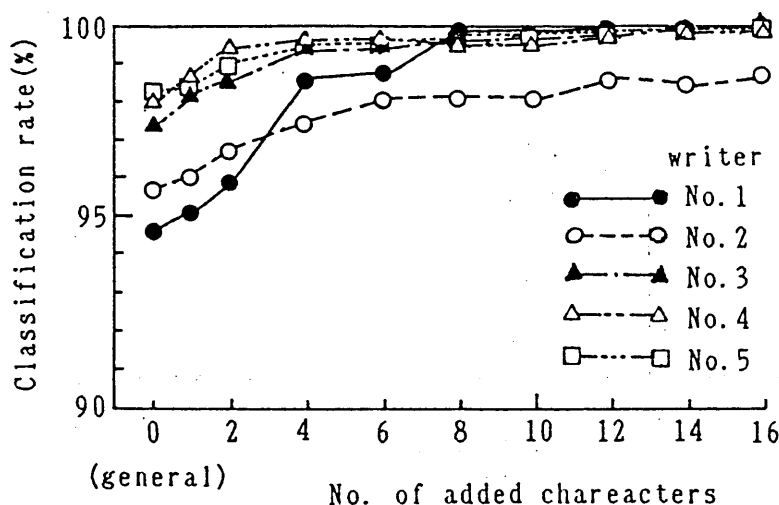


図5. 4 更新型辞書における特定筆記者の文字数と正読率

また、筆記者以外の文字を含めて学習させた辞書の方が、筆記者だけで学習した専用辞書より、高い正読率を示すことは、筆者らが複合類似度法で行った実験でも検証されており⁽⁸⁾、特徴抽出法に依存しない現象であると思われる。

② 修正型辞書

平均特徴ベクトルの計算に用いた特定筆記者の文字数と正読率の関係を図5.5に示す。

文字数が1文字の場合には、汎用辞書による正読率より、悪い正読率となる筆記者の方が多い。このことは、文字を一文字学習しただけでは、その筆記者の字種ごとの字形を適切に代表できないことを示しており、一部の字種では、10文字認識して10文字誤読となる場合もあった。

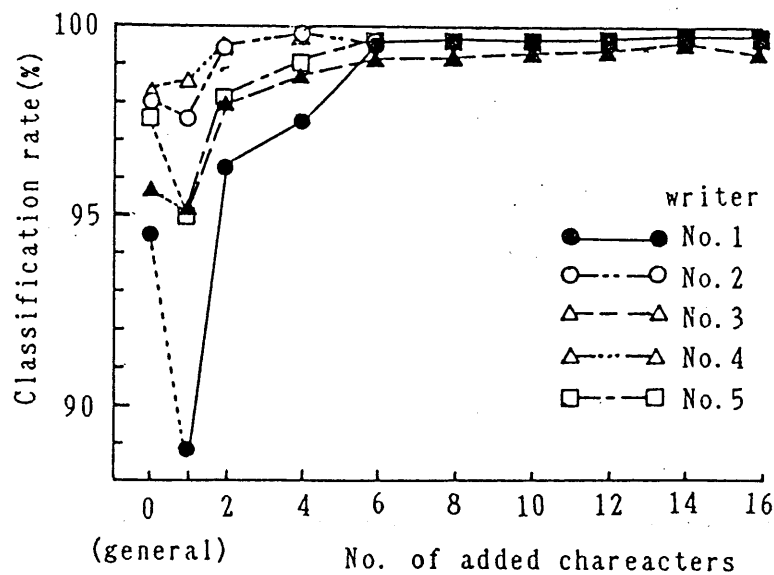


図5. 5 修正型辞書における特定筆記者の文字数と正読率

しかし、学習文字数を増やしていくと、そのような誤読はなくなり、8文字程度の学習で、正読率はほぼ飽和に達し、10文字学習した専用辞書と同程度かそれ以上の正読率が得られる。

③ 更新型辞書と修正型辞書の正読率

2つの辞書における特定筆記者の文字数と正読率の比較を図5. 6に示す。

いずれの方法も専用辞書より、かなり良い正読率が得られている。また、特定筆記者の文字を1文字だけ学習したときには、更新型辞書の方がよい正読率であるが、2文字以上の場合には、修正型辞書の方が良い正読率となる。そして、学習文字数が8文字程度で飽和に達し、両手法の差異はほとんどなくなる。

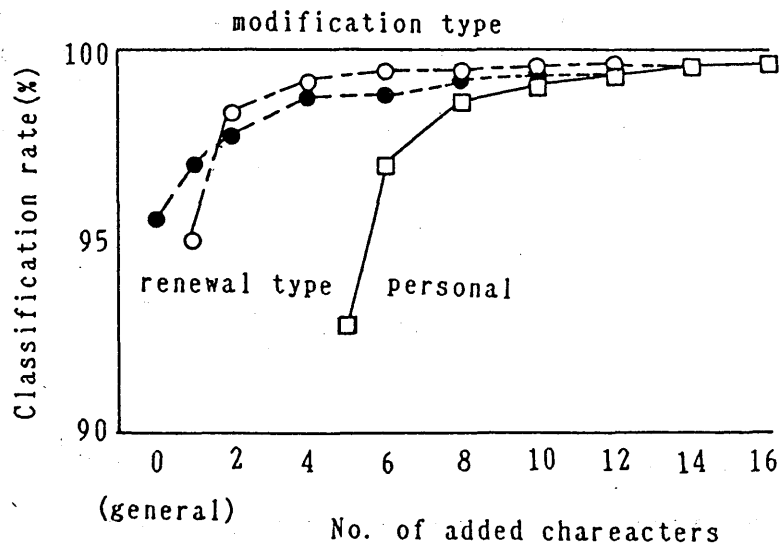


図 5. 6 更新型辞書と修正型辞書の比較

5. 3. 2 混合した適応型辞書

筆記者適応型の文字認識システムを実際に使用する場合を考えると、特定の筆記者の文字が少数しか得られていない段階でも、精度の良い認識を行える方法が望ましい。この観点から特定筆記者の文字が1文字しか得られていない場合を除けば、修正型辞書の方が、望ましい方法であると考えられる。

そこで、修正型辞書の平均ベクトルの作成に、汎用辞書の平均ベクトルを考慮する方法を次に提案する。

(1) 重み付き平均ベクトルによる認識

特定筆記者の平均ベクトル \bar{x} と汎用辞書の平均ベクトル \bar{y} より、式(5.2)で定義される重み付き平均ベクトルを考え、これを修正型辞書の平均ベクトルに用いた認識実験を行った。

$$w_T' = (1 - m) \cdot w_T' + m \cdot w_T' \quad (5.2)$$

但し, m : 重み係数 ($0 \leq m \leq 1$)

この式は $m = 0$ のときには, 汎用辞書による認識となり, $m = 1$ のときには, 修正型辞書で認識を行うこととなる. そして, $0 < m < 1$ のときには, 多数の筆記者の文字の平均ベクトルと特定筆記者の文字の平均ベクトルの混合したベクトルを使用することを意味する.

特定筆記者の文字数 N を 1, 2 及び 10 とした場合における, 重み係数 m と正読率の関係を図 5.7 に示す.

1 文字学習の場合には, $m = 0.25$ で最高の正読率となり, 95.0% ($m = 1.0$) から 97.6% まで向上した. 2 文字学習の場合には, $m = 0.75$ で最高となり, 10 文字学習のときには, $m = 0.75$ と 1.0 で最高の正読率となった.

この結果より, 重み係数 m の値は一定ではなく, 学習文字数が増加するにしたがって, 減少させるのが望ましいと考えられる.

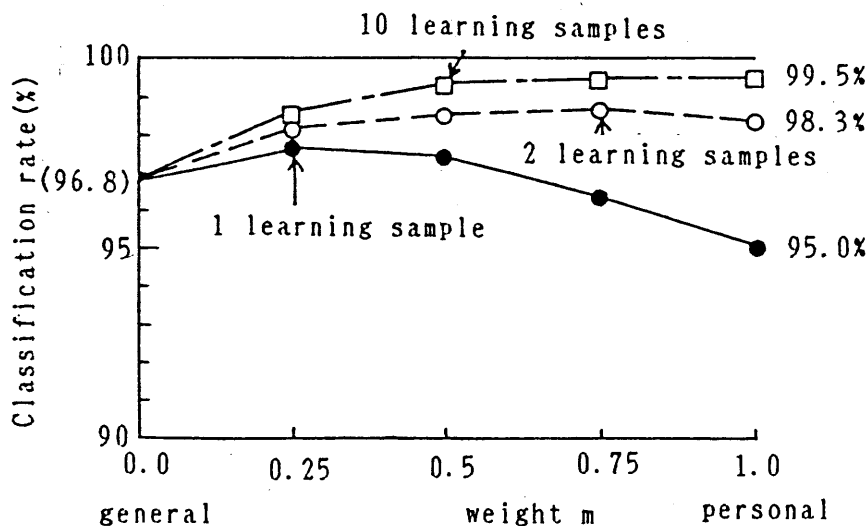


図 5. 7 重み係数と正読率

内藤ら⁽²⁾もストローク密度関数法で、専用平均と汎用平均で同様な認識実験を行い、同様な傾向を示しているが、固有値・固有ベクトルを使用していないため、学習文字数が少ないときに、特定筆記者の平均ベクトルの重みを重くすると、図5.7に比べ、正読率の低下が著しい。

(2) 混合型辞書

前節の考察により、修正型辞書の平均ベクトルとして、特定筆記者の既に筆記された文字の平均ベクトルを使う代わりに、式(5.3)を用い、固有値・固有ベクトルは、汎用辞書と同じものを使用した辞書(混合型辞書)を提案する。

$${}_0\bar{f}^l = \frac{1}{{}_0N + 1} ({}_0\bar{f}^l + \sum_{i=1}^{}_0N {}_0f^l{}_i) \quad (5.3)$$

但し、 ${}_0f^l{}_i$: 特定筆記者の字種 l の i 番目の学習

文字の特徴ベクトル

${}_0\bar{f}^l$: 汎用辞書の字種 l の平均ベクトル

${}_0N$: 特定筆記者の学習文字数

式(5.3)は、汎用辞書の平均ベクトルを特定筆記者の1文字と同じ重みで学習に使用していると解釈できる。この式の ${}_0N$ を1, 2, 10と変化させることは、式(5.2)の重み係数 m を0.5, 0.7, 0.9と変化させることに対応しており、前節の結果を実現する一つの方法である。

この方法による特定筆記者の文字の正読率を図5.8に示す。筆記者No.1以外のすべての筆記者で、1文字学習しただけで、汎用辞書よりも高い正読率が得られており、この方法が効果的であることが確

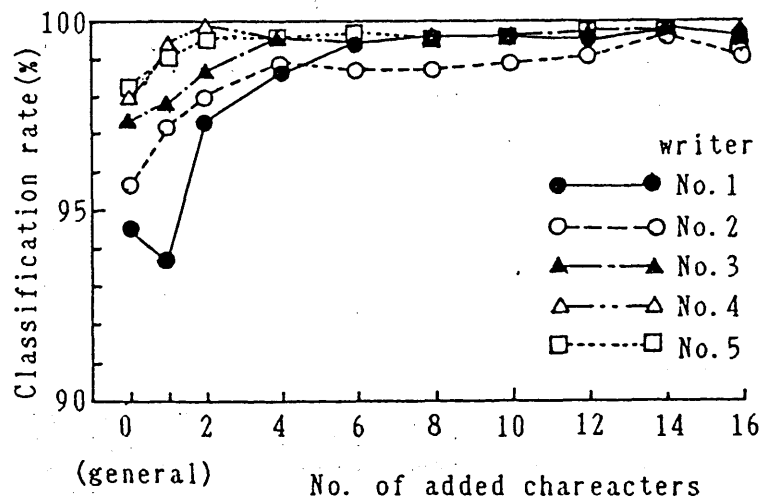


図5. 8 混合型辞書における特定筆記者の文字数と正読率

認できる。

専用辞書と前に述べた2つの方法、この方法との平均正読率における比較を、図5.9に示す。式(5.3)を用いた方法は、他のいずれの方法より、高い正読率か、同程度の正読率であり、1文字しか学習していないときに汎用辞書より低い正読率となる修正型辞書の欠点が改善されたと考えられる。

5. 3. 3 3種類の辞書の比較

本章で提案した3種類の辞書の認識性能、計算コスト、必要な記憶容量について比較検討する(表5.1)。

(1) 認識性能

更新型辞書は、1文字でも特定筆記者の文字を学習すれば、汎用辞書より高い正読率を示すようになる。それに比べ、修正型辞書は、

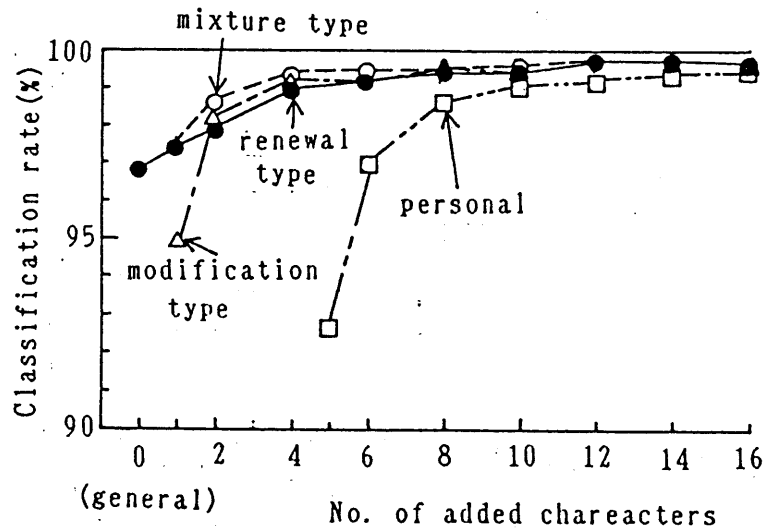


図5. 9 更新型辞書と修正型辞書, 混合型辞書の比較

- 1 文字の学習では, 汎用辞書より, 低い正読率しか得られないが,
- 2 文字以上学習すれば, 更新型辞書より, 高い正読率が得られる.

修正型辞書を改良した混合型辞書では, 特定筆記者の学習文字数が, いくつかの場合でも, 更新型辞書や修正型辞書より, 高い正読率を示し, 有効な適応方法であると考えられる.

(2) 計算量

更新型辞書は, 特定筆記者の文字を一文字学習することにより, 平均, 共分散行列, 固有値・固有ベクトルを計算し直す方式であり, 計算量は非常に多い. それに対して, 修正型辞書や混合型辞書は, 平均ベクトルだけを計算し直す方法であり, 非常に少ない計算量で計算できる. また, 更に計算量を減らしたい場合には, 変形がわすかであれば, 漸化式⁽⁹⁾も利用できる.

実際の計算量を名古屋大学大型計算機センターのFACOM M 382で計測した結果, 更新型辞書で140.7ms/文字, 修正型辞書, 混合型辞書で0.1ms/文字であり, 前者が後者の約1

400倍の計算量であることが分かった。そして、その大部分は、共分散行列の固有値解析にかかる時間であった。

(3) 記憶容量

3種類の辞書を作成し直すための記憶容量について考察する。修正型辞書、混合型辞書では、平均特徴ベクトルだけを保存しておけば良いのに対して、更新型辞書では、平均特徴ベクトルと共分散行列を必要とする。特徴ベクトルを64次元とすると、更新型辞書の記憶容量は修正型辞書、混合型辞書に比べ、65倍となる。

4バイト実数のCPUを用い、記憶容量を計算すると、修正型辞書、混合型辞書の記憶容量は、256バイト/字種であり、3000字種を読み取り対象字種としても、おおよそ800キロバイトとなり、市販の1メガバイトのフロッピーディスク1枚に記憶させる

表5. 1 3種類の適応型辞書

			更新型辞書	修正型辞書	混合型辞書
辞書	平均		専用	専用	専用
	固有値 固有ベクトル		専用	汎用	汎用
正読率	使用した 特定筆記者の文字数	1文字	汎用辞書より高い	入力文字により低下	汎用辞書より高い
		2文字以上	汎用辞書より高い		
	専用辞書との比較		特定筆記者の文字数が同数ならば、専用辞書より高い		
再計算量			1400	1	
記憶容量			65	1	

ことができる。したがって、筆記者が自分専用の辞書を安価な媒体に作成しておき、文字認識システムを使用するときには、その辞書を使用するようにすれば、実用性の高いシステムが構成できる。

これらの考察より、本章で提案した3種類の適応型辞書の中では、混合型辞書が、認識性能、計算量、記憶容量の面で総合的に最も実用化に適していると考えられる。

5. 4 むすび

本研究では、手書き文字の個人性に、文字認識の立場から注目して、筆記者に対して適応機能を持つ文字認識システムの3種類の方法（更新型、修正型、混合型）を提案した。そして、その有効性を調べるため、加重方向ヒストグラム法を用いて、5人の筆記者のひらがな全字種（46字種）で基礎実験を行った。その結果、次の知見が得られた。

- （1）認識対象者の文字だけで作成した専用辞書を用いて認識実験をすると、学習文字数が少なくても、不特定多数の筆記者の文字（ETL4）を使用して作成した汎用辞書（100文字／字種学習で96.8%）より、良い正読率（10文字／字種学習で99.0%）が得られることが確認できた。
- （2）筆記者に対して適応機能を持つ3方法の文字認識方式を示し、認識実験の結果どの方法でも、汎用辞書より、よい正読率（筆記者の文字10文字／字種使用で更新型辞書：99.3%，修正型辞書：99.5%，混合型辞書：99.5%）が得られることが明確化した。また、認識対象者の学習文字数が同数のときには、専用辞書（10文字／字種学習で99.0%）より、高い正読率が得られた。
- （3）混合型辞書は、認識性能が高く、辞書修正に必要な計算量、記憶容量（800キロバイト）が少ないので、実用に適した適応方法であると考えられる。

今後の課題として、（１）本章で提案した筆記者の文字に適應する方法を、手書き漢字を対象として検討すること、（２）認識対象としている字種の一部の字種より、字種に依存しない筆記者の個性（書き癖）を抽出して、筆記されていない他の字種の認識に、活用する方法^{（１０）}を確立することなどが残されており、特に（２）の課題の有効な解決策が研究されれば、本章で議論した方法は広く認識装置で利用されると考えられる。

第6章 文字認識における筆記者の個性ベクトルの利用法

あらまし

特定筆記者の文字を学習した適応型辞書を使用すれば、汎用辞書の正読率よりかなり良い結果が得られることが第5章で述べられた。しかし、この方法をそのまま手書き漢字認識に適用すると、漢字の字種数が多いので、学習文字数が限られている場合には、筆記者が事前に筆記した字種を書く頻度はかなり低くなる。そのため、適応型辞書を有効に利用できる場合はかなり少ないと考えられる。そこで、本章では、事前に筆記していない字種においても適応型辞書が有効に機能するように、少数の字種の文字より筆記者の字種に依存しない個性ベクトル（書きぐせ）を抽出する方法を提案し、その個性ベクトルから未学習の字種における筆記者の個人平均ベクトルを推定する方法を述べる。この手法を加重方向指数ヒストグラム法において適応し、推定した個人平均ベクトルを用いてひらがなの全字種で認識実験した結果、5人の筆記者中3人で良い正読率が得られ、平均正読率は96.8%（汎用辞書）から97.1%に向上し、その有効性が確かめられた。

6. 1 はじめに

従来の光学式文字認識装置（OCR）は集中型の処理装置と位置付けされており、大量のデータを高速に処理する集中型のオフィス機器であった。しかし、近年の半導体技術の進歩は、目ざましく、情報機器のハードウェアの低価格化は急速に進んできており、OCRの構成要素の価格も安価になりつつある。一方、社会の情報化も高度化してきており、数値情報だけでなく、文字情報の処理も活発化してきている。そして、それにともなって、キーボード以外の情報入力機器への期待も高まっている。

このような状況において、OCRは将来、安価で小型のものとなり、小規模な分散型オフィスなどでも使用され、パーソナル機器にもなりうると考えられる。パーソナル機器となったときの利点は、①筆記者の字形を記憶でき、認識性能が向上すること、②筆記者の使用する単語の頻度や文脈情報によって後処理による自動訂正が利用できること、③誤って入力されたときの訂正が容易となることなどがあり、不特定多数の筆記者に対する装置に比べ、使用者にとっては使いやすくなる要因が多い。

この章の研究はOCRをパーソナル機器として利用する形態を想定したものであり、筆記者を限定することによって筆記者の字形を記憶し、認識性能が向上することを目的としている。具体的な利点としては、筆記者に特有な字形の文字も読めるようになることであり、続け字や省略のある文字、傾斜字、異体字なども読めるようになると考えられる。

このような主旨に沿った研究のひとつに、第5章で述べた研究⁽¹⁾がある。その研究では、特定筆記者の文字を学習した適応型辞書を3種類（更新型辞書、修正型辞書、混合型辞書）比較し、混合型辞書（平均ベクトルは、特定筆記者の文字と汎用平均の加重和とし、固有値・固有ベクトルは汎用辞書のものを用いる方法）を使用すれば、汎用辞書の正読率よりかなり良い結果が得られることが確かめられた。しかし、この方法をそのまま手書き漢字認識に適用すると、漢字の字種数が多いので、学習文字数が限られている場合、筆記者が事前に筆記した字種を書く頻度はかなり低くなり、適応型辞書を有効に利用できる場合はかなり少なくなる。

そこで、本章では、事前に筆記していない字種においても適応型辞書が有効に機能するように、少数の字種の文字より筆記者の字種に依存しない個性ベクトル（書きぐせ）を抽出する方法を提案し、その個性ベクトルから未学習の字種における筆記者の個人平均ベクトルを推定する方法を述べる。この手法を加重方向指数ヒストグラム法⁽²⁾によって得られた特徴ベクトルに適応し、ひらがなの全字種（46字種）を用いて求めた5人の筆記者の個性ベクトルを比較した結果、筆記者ごとの差異は明確に抽出でき、それを用いて汎用計算機で認識実験した結果、5人の筆記者中3人でよい正読率が得られ、平均正読率は96.8%（汎用辞書）から97.1%に向上し、その有効性が明確となった。

6. 2 筆記者の個性ベクトルの抽出と個人平均ベクトルの推定

本節では、手書き文字における筆記者の個性ベクトルを抽出する一方法を提案する。筆記者の個性が反映される辞書の構成要素としては、平均ベクトルと固有値・固有ベクトルがあるが、ここでの議論は、第5章で述べた平均ベクトルだけを特定の筆記者に適応させて、認識率を向上させるタイプの適応型辞書を使用する場合を想定し、平均ベクトルに関するものに限定する。また、その個性ベクトルを用いて、筆記されていない字種の筆記者の個人平均ベクトルを推定する方法についても述べる。

6. 2. 1 個性ベクトルの抽出手順

手書き文字を特徴抽出し、得られた特徴ベクトル（多次元ベクトル）から、筆記者の個性ベクトルを抽出する手順を以下に示す。

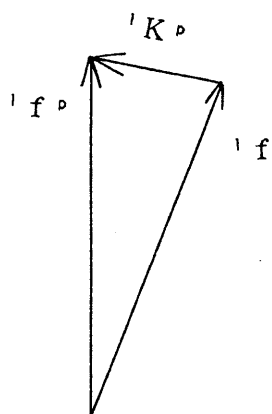
(1) 字種 l ，筆記者 p の個人平均ベクトル ${}^l f^p$ と不特定多数の筆記者の文字から汎用平均ベクトル ${}^l f$ を求める。

(2) 字種 l ，筆記者 p の個性ベクトル ${}^l K^p$ を個人平均ベクトル ${}^l f^p$ と汎用平均ベクトル ${}^l f$ の差ベクトルで定義する（式(6.1)，図6.1）。

$${}^l K^p = {}^l f^p - {}^l f \quad (6.1)$$

(3) 筆記者 p の字種に依存しない個性ベクトル K^p (以下, "個性ベクトル" という) を L 個の字種における平均個性ベクトルで定義する (式 (6.2), 図 6.2) .

$$K^p = \frac{1}{L} \sum_{l=1}^L {}^l K^p \quad (6.2)$$



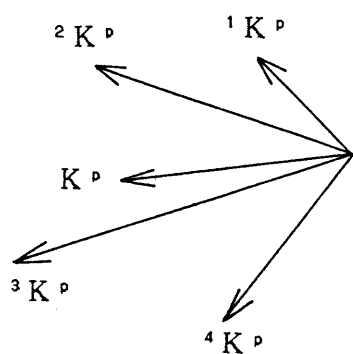
${}^1 f^p$: 字種 l , 筆記者 p の個人平均ベクトル

${}^1 f$: 不特定多数の筆記者の文字から求めた汎用平均ベクトル

${}^1 K^p$: 字種 l , 筆記者 p の個性ベクトル (差ベクトル)

$${}^1 K^p = {}^1 f^p - {}^1 f$$

図 6. 1 差ベクトル



L 個の字種における平均個性ベクトル

$$K^p = \frac{1}{L} \sum_{l=1}^L {}^l K^p$$

図 6. 2 個性ベクトル

このように定義した個性ベクトルが筆記者の書き癖を有効に表現している場合は、字種ごとの個性ベクトルがほぼ同じ方向を向いている場合である。しかし、字種ごとに個性ベクトルが全く違う方向を示している場合は、互いに打ち消し合い、個性ベクトルの大きさはほとんど零となり、その有効性は低くなる。

6. 2. 2 個人平均ベクトルの推定

6. 2. 1 の手順 (3) で求めた個性ベクトルと汎用平均ベクトルから、次に示す方法により、個人平均ベクトルを推定する。

(4) 未学習の字種 m の汎用平均ベクトル \mathbf{u}_f に (3) で求めた個性ベクトルを加え、推定した筆記者 p の個人平均ベクトル (以下, "推定ベクトル" という) $\hat{\mathbf{u}}_f^p$ を求める (式 (6.3), 図 6.3)。

$$\hat{\mathbf{u}}_f^p = \mathbf{K}_p + \mathbf{u}_f \quad (6.3)$$

この方法で正しく推定が行える場合は、汎用平均ベクトルと個人平均ベクトルの位置関係の差異が、多次元空間上で全字種とも同じ場合であり、ちょうど向きと大きさが個性ベクトルと同じ場合には、推定誤差は零となる (図 6.4)。

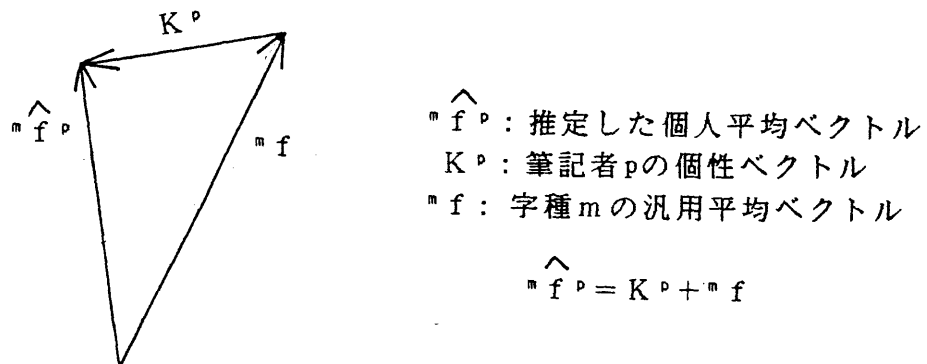


図 6. 3 推定したベクトル

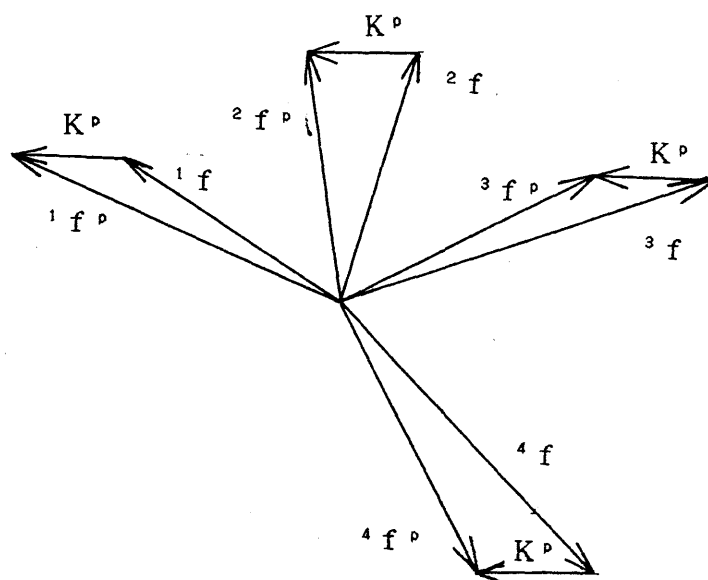


図 6. 4 正しく推定が行える場合
 (各字種の汎用平均ベクトルと個人平均ベクトルの相対的位置が同じ場合には正しく推定ができる)

6. 3 抽出した個性ベクトルの図表現と

推定した個人平均ベクトルの性質

前節で述べた個性ベクトルを、第4章で提案した加重方向指数ヒストグラム法により特徴抽出を行い、“ひらがな”の全字種（46字種）を使用して実際に求め、その性質について考察する。また、この個性ベクトルと汎用平均ベクトルを用いて個人平均ベクトルも推定したので、その性質についても述べる。

6. 3. 1 抽出した個性ベクトルの図表現

（1）実験データ

個性ベクトル抽出に使用した文字データは三重大大学の学部生・院生が筆記したひらがなの全字種46字種であり、各筆記者当りの文字数は、16文字／字種（5人分）である。また、汎用平均ベクトルの作成には、電総研作成の自由手書き文字データベースETL4（名古屋大学の学部生が筆記）の100人分の文字（100文字／字種）を用いた（表6.1）。

各筆記者の位置・大きさの正規化を行った文字例を図6.5に示す。

（2）抽出した個性ベクトルの図表現

加重方向指数ヒストグラム法（ $4 \times 4 \times 4$ 方向）により求めた6次元の特徴ベクトルから作成した、個人平均ベクトルと汎用平均

表 6. 1 使用した文字データベースの概要

	字 種	1字種当りの の文字数	筆記者数
特定筆記者の文字	ひらがな 4 6	26文字/人 学習 16 認識 10	5 人
不特定多数の筆記者の文字 ETL4B	ひらがな 4 6	1文字/人	100人

OCR用紙にHBの鉛筆で書いた文字

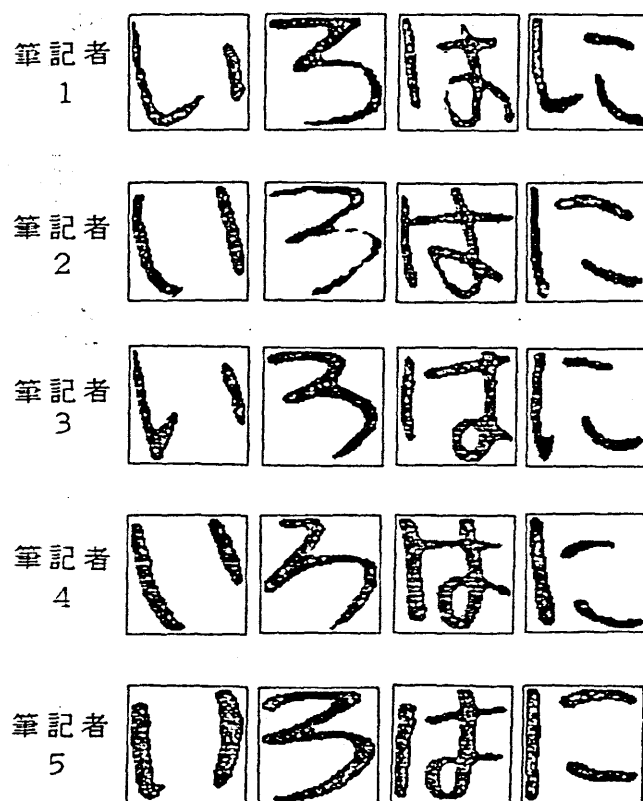


図 6. 5 各筆記者の文字例

ベクトル，それらの差ベクトル（字種ごとの個性ベクトル）の図表現を図6.6に示す。また，これらの差ベクトルの全ひらがな字種における筆記者別の平均ベクトル（字種に依存しない個性ベクトル，以下，“個性ベクトル”という）を図6.7に示す。以下に，抽出した筆記者ごとの個性ベクトルの特徴を述べる。

- ① 筆記者1では，文字の左上部分において水平方向の線分を長めに書く個性が抽出できている。
- ② 筆記者2では，文字の中心から上の中央部において水平方向の成分を長く書く個性と中心部を縦長に書く個性が抽出されている。
- ③ 筆記者3では，全体的に零に近く字種ごとに個性が違っているようである。
- ④ 筆記者4では，文字の中心よりやや上部を水平方向に長く書く個性が抽出されている。
- ⑤ 筆記者5では文字の下半分において右上がりの線分を長く書く個性が抽出できている。

これらのことから，個性ベクトルは筆記者ごとによりかなり異なり，この方法である程度，手書き文字の個性が抽出できることが分かる。

6. 3. 2 推定した個人平均ベクトルの性質

5人の筆記者において，同一字種における，各筆記者の個人平均ベクトル \hat{f}_p ，推定ベクトル \hat{f} ，汎用平均ベクトル \bar{f} の3者間のユークリッド距離値を求め，その度数分布を調べた（図6.8）。その結果，推定ベクトルは多くの字種で，汎用ベクトルより個人平均ベ

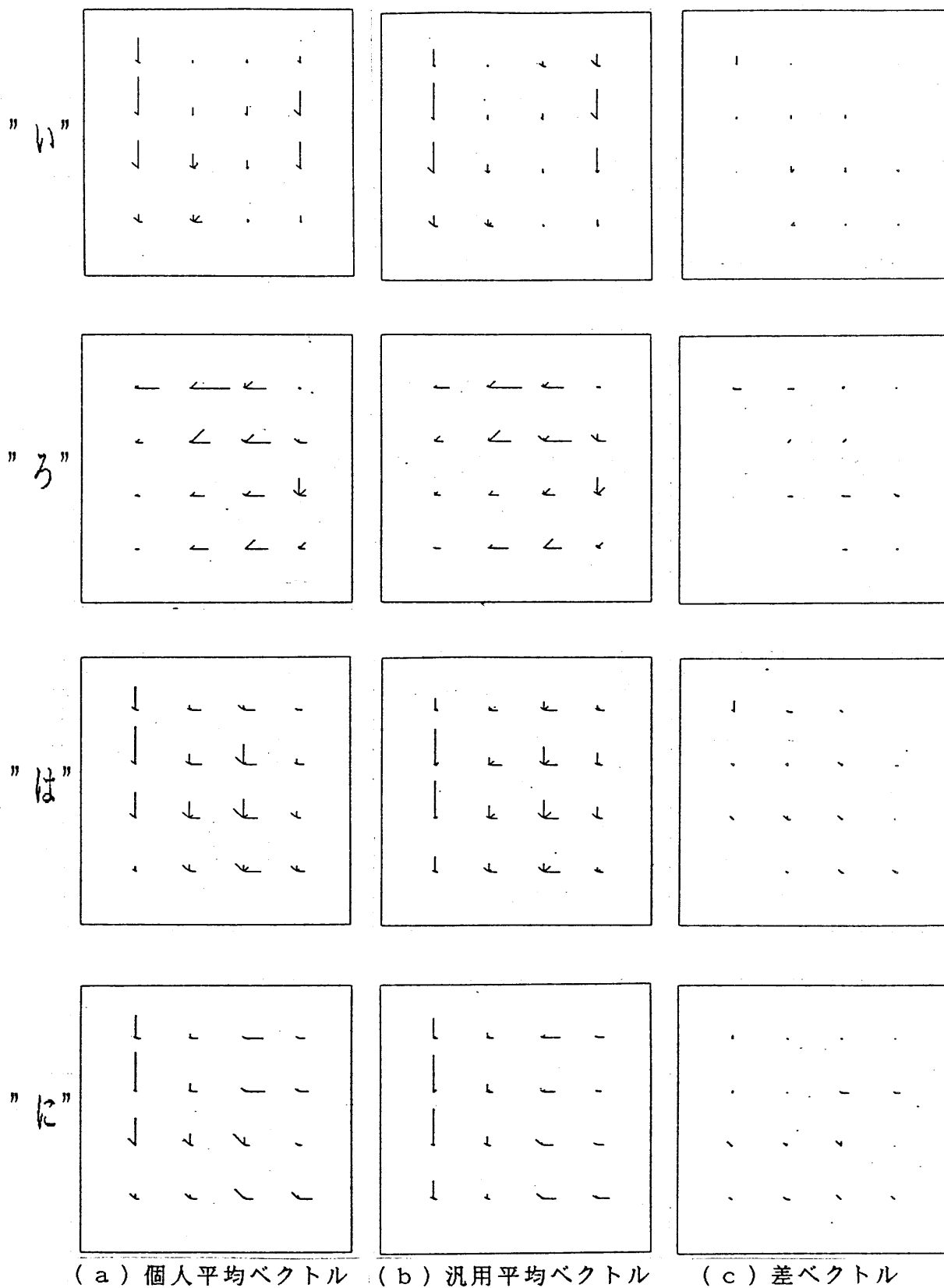
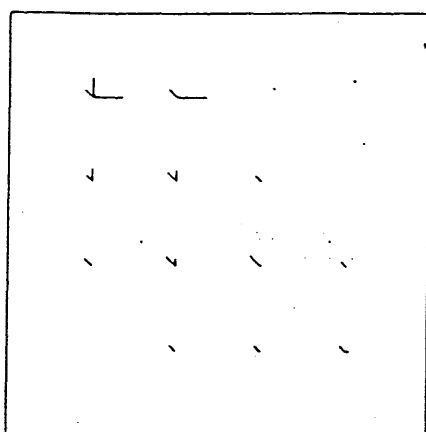
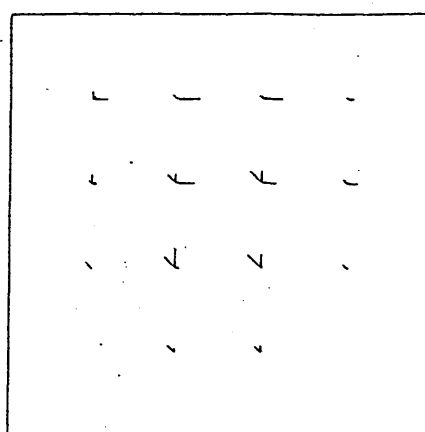


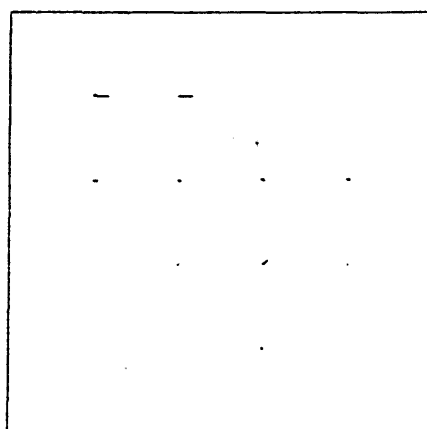
図 6. 6 個人平均ベクトル，汎用平均ベクトル，
差ベクトルの図表現



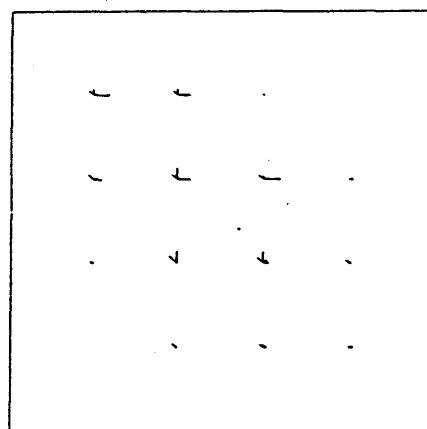
筆記者 1



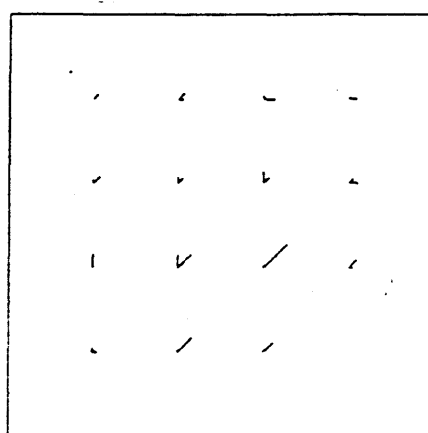
筆記者 2



筆記者 3



筆記者 4



筆記者 5

図 6. 7 各筆記者の個性ベクトルの図表現

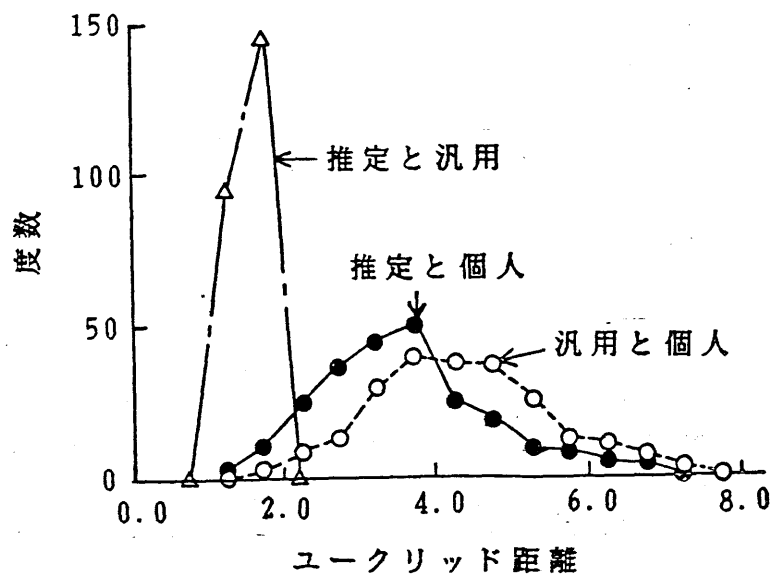


図6. 8 距離値の分布状況 (46字種×5人)

クトルに近くなっており、特に距離値の大きい字種でその効果が大
きいことが明確となった。この結果、6.2で述べた方法で、個人
平均ベクトルの推定はある程度正しく行われることが分かった。

6. 4 認識実験

識別関数として、疑似ベイズ識別関数⁽²⁾を使用し、個性ベクトルの作成に使用していない各特定筆記者の全字種の文字（未学習文字）を対象とした、認識精度に関する基礎実験を汎用計算機で行った。ここで、識別関数に使用した固有値・固有ベクトルは、汎用平均ベクトル作成に用いた電総研作成の自由手書き文字データベースETL4の100人分の文字から計算した固有値・固有ベクトルである。

認識実験の結果、推定ベクトルを用いた場合には、5人の筆記者中3人で汎用辞書による認識結果より、良い正読率が得られ、5人の平均正読率は96.8%（汎用辞書）から97.1%に向上した（図6.9）。図では専用辞書の認識結果⁽³⁾も参考のために示したが、これは推定誤差が零で個人平均ベクトルが推定できた場合の結果であり、今後の研究は、少ない個性ベクトルの数で（字種ごとに個性ベクトルを作製すれば、推定誤差は最小となるが、未学習の字種に対しては効果はない）この差を縮める方法を考案することである。

次に、個性ベクトルの重みを変えて、個性ベクトルと汎用平均ベクトルの比率を変える認識実験を行った。5人の筆記者の平均正読率を図6.10に示す。個性ベクトルの重みは0.5と1.0で最高の正読率（97.1%）となり、6.2で述べた理論値（1.0）とほぼ一致した結果が得られた。

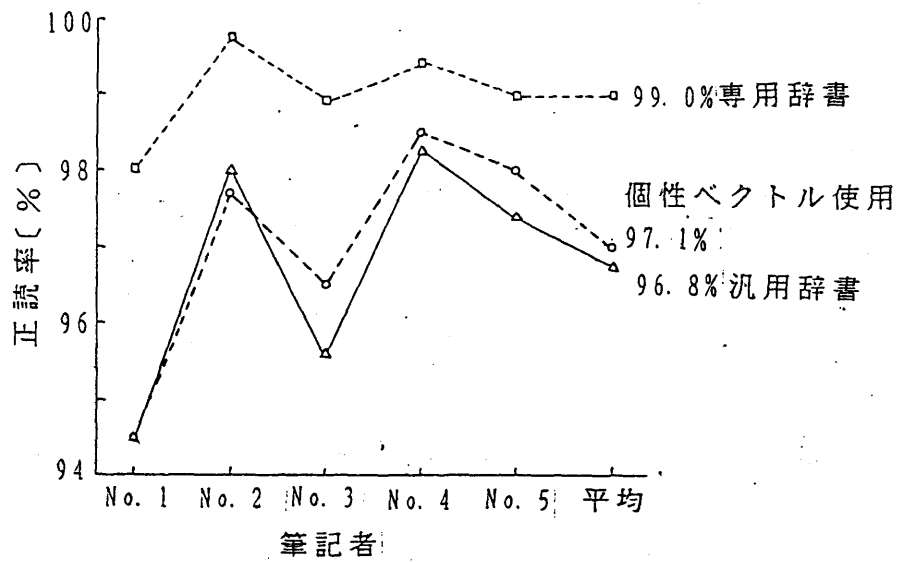


図 6. 9 個性ベクトルを用いた効果

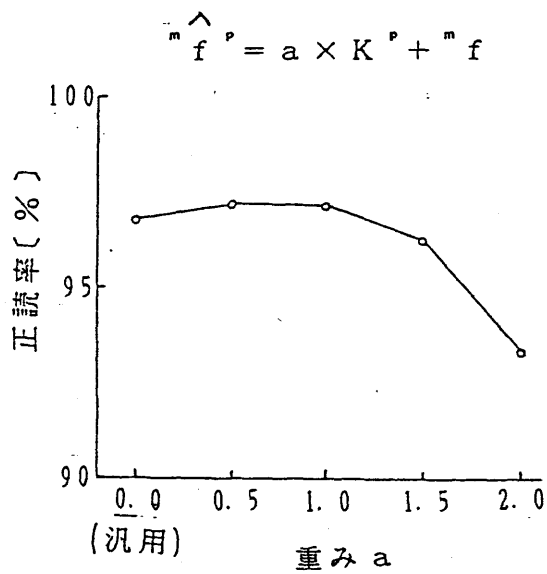


図 6. 10 個性ベクトルの重み

6. 5 むすび

本章では、特定筆記者が事前に筆記していない字種においても、適応型辞書が有効に機能するように、少数の字種の文字より筆記者の字種に依存しない個性ベクトル（書きぐせ）を抽出する方法を提案した。そして、第4章で述べた加重方向指数ヒストグラム法において、ひらがなの全字種（46字種）を使い、基礎的検討を行った。その結果、次のような知見が得られた。

- (1) 全字種の平均個性ベクトルを図表現して、観察した結果、筆記者による差異が明確となり、書きぐせが抽出できることが確認できた。
- (2) 個性ベクトルを用いた推定ベクトルは、多くの字種で汎用平均ベクトルと比べると、個人平均ベクトルに近いことがユークリッド距離を測ることによって明らかとなった。
- (3) 個性ベクトルを考慮して認識実験すると、汎用辞書の正読率（96.8%）より良い正読率（97.1%）が得られることが確かめられた。

なお、今後の課題として、

- (1) 推定ベクトルを用いる効果をより高めるために、字種をクラスター化しておき⁽⁴⁾、クラスター内において個性ベクトルを抽出する方法を確立すること。
- (2) 手書き漢字認識に本手法を適用する具体的な手法を考察すること。

などが残されている。

第7章 あとがき

この章では、前章までの内容を要約すると共に、今後の課題について述べ、本論文をしめくくる。

第1章では、文字認識研究の意義、歴史と現状、認識方法の概要と本研究の目的を述べ、本論文の位置づけを行った。

第2章では、ストローク構造解析法の問題点の一つである線分の縮退・消失のほとんど生じない細線化アルゴリズムを新たに開発し、その有効性を検討するための認識実験を自由手書きカタカナ文字を対象にして行った。

第3章では、細線化、折線近似された文字パターンから、人工的に変形文字を発生し、被験者にそれを見せ、人間の字種識別基準の計測を行い、新たに提案した”心理的距離値”を用いて認識実験を行った。

第4章では、手書き漢字認識に適した特徴抽出法（加重方向指数ヒストグラム法）の開発と新しい識別関数（疑似ベイズ識別関数）の提案を行い、高精度で手書き教育漢字・ひらがなを認識するシステムを構成した。

第5章では、第4章で述べた方法を筆記者に適応する辞書の作成方法について考察し、新しい適応型辞書を見いだし、自由手書きひらがな文字で基礎的な認識実験を行い、その効果を検証した。

第6章では、字種に依存しない筆記者の個性ベクトルを抽出する一方法を提案し、第4章で述べた文字認識システムに組み込み、そ

の有効性を認識実験により明確にした。

本研究では、手書き文字を高精度で認識するアルゴリズムを研究し、手書きカタカナ、ひらがな、漢字を対象とする多くの認識実験により、その有効性を明らかにした。

今後の研究課題として、

- (1) 認識精度をより向上させ、もっと質の悪い文字（走り書き文字など）を読み取る研究
- (2) より制限の弱い場所（例えば、罫線だけの用紙、白紙の紙、3次元物体上）に書かれた文字を切り出す研究
- (3) クラスタリングなどの手法を用いて、認識精度を落とさずに計算量を減らす研究

などが残されている。

謝 辞

最後に、筆者が名古屋大学大学院在学中に手書き文字認識の研究を始めたときから、三重大学在職中の今日まで、筆者の研究の直接の指導者として、懇切なる御指導と生活面での御配慮を頂きました三重大学三宅康二教授に深く感謝いたします。

また、名古屋大学在学中から筆者の研究の進展を常にあたたく見守って頂き、また、本論文の提出に当たり懇切な御指導を賜りました名古屋大学福村晃夫教授、機会あるごとに御激励頂いた愛知工業大学市川真人教授（元名古屋大学教授）、豊橋技術科学大学本多波雄学長（元名古屋大学教授）、名古屋大学鳥脇純一郎教授、研究上極めて有益な御助言や御激励、貴重な資料（特定筆記者の文字データ等）を御提供頂いた聖徳学園女子短期大学吉村ミツ教授（元名古屋大学）に深謝いたします。

筆者が最初に研究を手掛けた細線化法において貴重な御意見をくださった、と同時に三重大学において公私共に直接御指導を頂いた名古屋大学横井茂樹助教授、具体的な文字認識システム（ストローク構造解析法）について御教授頂き、文字認識方法について熱心な御指導をして頂いた三重大学木村文隆助教授にお礼を申し上げます。

また、細線化法について御意見をくださったリコー（株）田村秀行博士（元通産省電子技術総合研究所）にお礼を申し上げます。

第3章のプログラム作成並びに実験に御協力頂いた村瀬晶彦氏（現松下電器産業（株））、皿井修氏（現松下電子（株））、第4章のプログラム作成並びに実験に御協力頂いた栗田昌徳氏（現富士通

(株))、原田智夫氏(現日本電装)、第5章のプログラム作成並びに実験に御協力頂いた森田裕之氏(現日本電装)、筆者の原稿のタイプ入力やコピーを手伝ってくださった三重大学の結城和子事務官、有益な御意見をくださった豊田工業高等専門学校野澤繁之教授、同学竹下鉄夫助教授、新日本製鉄(株)の竹内斎之郎氏並びに種々の御協力や御意見をくださった元名古屋大学の三宅研究室の皆様、元・現三重大学の三宅研究室の皆様にご感謝いたします。

また、本研究に用いた手書き文字データベースを作成・提供された通産省電子総合技術研究所、富士通の関係諸氏、計算機の使用に関して御世話になった名古屋大学情報・自制教室の皆様、名古屋大学大型計算機センターの皆様、三重大学計算センターの皆様にご厚くお礼申し上げます。

なお、本研究の一部は文部省科学研究費(一般研究(B))(00546200)、奨励研究(A)(61780036)の補助によるものであり、御配慮くださった方々にお礼申し上げます。

文 献

第1章 文 献

- (1) D.H.Hubel and T.N.Wiesel: " Receptive fields, binocular interaction and functional architecture in the cat's visual cortex", J. Physiology, 160, p. 106 (1962).
- (2) G. Tauschek: "Lesende Maschine", Österreichisches Patentamt. Patentschrift No. 116799. Klasse 21a.
Ausgegeben am 10. März 1930. (Angemeldet am 30. Mai 1928. -Beginn der Patentdauer: 15. Oktober 1929).
- (3) G. Tauschek: "Reading Machine", U. S. Patent Office 2, 026, 329, Patented Dec. 31 (1935-12)
- (4) 飯島, 森: " 人間の識別能力に迫るOCR' ASPECT / 7 1' ", 日経エレクトロニクス, p 66-80 (1972-5. 22).
- (5) 通商産業省工業技術院編: " 大型プロジェクトパターン情報処理システム 研究開発成果発表会論文集", 日本産業技術振興協会 (1980-10).
- (6) 橋本新一郎編著: " 文字認識概論", 電気通信協会 (1982-03).
- (7) 中田和男編: " パターン認識とその応用", コロナ社 (1978-09).
- (8) J I S ハンドブック情報処理: " 光学式文字認識のための手

書き文字（片仮名，数字，英字，記号）C 6 2 5 4 -1979, C 6 2 5 5 -1979, C 6 2 5 6 -1981, C 6 2 5 7 -1981, 日本規格協会（1979, 1981）.

（9）森，坂井：“2000字種を100字／秒で読む印刷漢字のCRの開発”，日経エレクトロニクス，pp. 102-128（1977-10. 31）

（10）黒沢，麻田：“自由手書き文字認識”，東芝レビュー，41, 12, pp. 1012-1015（1986-12）.

（11）村瀬，木村，吉村，三宅：“パターン類似度法による手書き平仮名文字認識の実験”，信学技報，PRL 79-2（1979-05）.

（12）村瀬，木村，吉村，三宅：“パターン整合法における特性核の改良とその手書き平仮名文字認識への応用”，信学論（D），J 64-D, 3, pp. 276-283（1981-03）.

（13）飯島泰蔵：“混合類似度による識別理論”，信学技報，PRL 74-24, pp. 45-54（1974-09）.

（14）S. Watanabe: "Karhunen-Loeve Expansion and Factor Analysis Theoretical Remarks and Applications", Trans. 4th Prague Conf. on Information Theory pp. 635-660 (1965).

（15）E. Oja: "Subspace Methods of Pattern Recognition", Research Studies Press, Letchworth and J. Wiley, NY (1983).

小川英光，佐藤誠訳：“パターン認識と部分空間法”，産

業図書 (1 9 8 6 - 0 4)

- (1 6) 奥野, 久米, 芳賀, 吉沢: " 多変量解析法", 日科技連,
p. 1 5 9 - 2 5 7 (1 9 7 1 - 1 0).
- (1 7) K.S.Fu: "Syntactic Methods in Pattern Recognition",
Academic Press(1974).
- (1 8) 宮崎, 星野: " 折線近似による手書きカタカナ認識", 信
学技報, P R L 7 4 - 6 2 (1 9 7 5 - 0 2).
- (1 9) 木村, 吉村, 三宅, 市川: " ストローク構造解析法による
自由手書き片仮名文字認識", 信学論 (D), J 6 2 - D,
1, P P. 1 6 - 2 3 (1 9 7 9 - 0 1).
- (2 0) 内藤, 小森, 淀川: " 手書き漢字認識のためのストローク
密度特徴", 信学論 (D), J 6 4 - D, 8, P P. 7 5
7 - 7 6 4 (1 9 8 1 - 0 8).
- (2 1) K.Yamamoto and S.Mori: " Recognition of handprinted
characters by an outermost point method", Pattern
Recognition, vol.12, pp.229-236(1980-12).
- (2 2) 山本和彦: " 弛緩整合法による手書き教育漢字認識", 信
学論 (D), J 6 5 - D, 9, p p. 1 1 6 7 - 1 1 7 4
(1 9 8 2 - 0 9).
- (2 3) 鶴岡, 栗田, 原田, 木村, 三宅: " 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", 信学論 (D),
J 7 0 - D, 7, p p. 1 3 9 0 - 1 3 9 7 (1 9 8 7 -
0 7).
- (2 4) H.A.Glucksman: "Classification of mixed font alphabe-
tics by characteristic loci", Digest of 1st Ann.IEEE

Computer Conf., pp.137-141(1967).

- (2 5) 森, 森, 山本: " 場の効果法による特徴抽出 - 閉じ状態の抽出 - ", 信学論 (D), J 5 7 - D, 5, p p. 3 0 8 - 3 1 5 (1 9 7 4 - 0 5).
- (2 6) 小森, 川谷, 石川, 飯田: " 特徴集積による手書き片仮名文字の認識 ", 信学論 (D), J 6 3 - D, 1 1, p p. 9 6 2 - 9 6 9 (1 9 8 0 - 1 1).
- (2 7) 吉田, 岩田, 山本, 榎井, 燕山: " 手書き文字認識システム ", 情報処理, 1 7, 7 (1 9 7 6 - 0 7).
- (2 8) 岡 隆一: " セル特徴を用いた手書き漢字の認識 ", J 6 6 - D, 1, p p. 1 7 - 2 4 (1 9 8 3 - 0 1).
- (2 9) 裴, 燕山, 山本: " 手書き漢字認識の一手法 - 多元圧縮法と部分パターン法による認識 - ", J 6 8 - D, 4, p p. 7 7 3 - 7 8 0 (1 9 8 5 - 0 4).
- (3 0) 塩野, 馬場口, 真田, 手塚: " 方向性マッチングによる常用手書き文字の認識 ", 信学論 (D), J 6 3 - D, 5, p p. 4 0 2 - 4 0 9 (1 9 8 0 - 0 5).
- (3 1) 馬場口, 北村, 塩野, 真田, 手塚: " 細線化を用いない文字パターンの方向セグメント抽出の一手法 ", J 6 5 - D, 7, p p. 8 7 4 - 8 8 1 (1 9 8 2 - 0 7).
- (3 2) 斉藤, 山田, 山本: " ずらし類似度による手書き漢字の識別実験 ", J 6 9 - D, 8, p p. 1 1 7 2 - 1 1 7 8 (1 9 8 6 - 0 8).
- (3 3) 森, 坂倉: " 画像認識の基礎 [I] - 前処理と形の特徴抽出 - ", オーム社 (1 9 8 6 - 0 1).

- (34) 電子通信学会編: "文字認識小特集", 電子通信学会誌, 61, 2 (1978-02).
- (35) 電子通信学会編: "日本語情報処理とそのシステム小特集", 電子通信学会誌, 63, 7 (1980-07).
- (36) 画像電子学会編: "文字・画像の入力方式", 画像電子学会誌, 11, 3 (1982-08) (特に, 三宅康二: "OCR入力", pp.137-146).
- (37) 鶴岡, 木村, 吉村, 横井, 三宅: "デジタル図形の一細線化法と手書き文字認識システムへの応用", 信学論(D), J66-D, 5, pp. 525-532 (1983-05).
- (38) 鶴岡, 村瀬, 木村, 横井, 三宅: "人間の字種識別基準を用いた自由手書き片仮名文字認識", 信学論(D), J68-D, 4, pp. 781-788 (1985-04).
- (39) 鶴岡, 森田, 木村, 三宅: "筆記者に対して適応機能を持った自由手書き文字認識", 信学論(D), J70-D, 10 (1987-10).
- (40) 日本語情報処理標準化調査委員会, C専門委員会: "手書教育漢字の字形集", 日本電子工業振興協会 (1981-12).

第2章 文 献

- (1) Blum, H.: "A transformation for extracting new descri-

- ptors of shape", in Models for the Perception of Speech and Visual Form, p.362 (Dunn, W., ed) MIT Press, Cambridge, MA(1967).
- (2) Rosenfeld, A.: " Connectivity in digital pictures", J. Assoc. Comput. Mach., 17, 1, pp. 146-160 (1970).
- (3) Steffanelli, R. and Rosenfeld A.: "Some parallel thinning algorithms for digital pictures, J. Assoc. Comput. Mach., 18, 2, p. 255-264 (April 1971).
- (4) Hilditch, C. J.: " Linear skeletons from square cupboards ", (Meltzer B. and Michie D., ed): Machine Intelligence 4, Edinburgh Univ. Press, p. 403-420 (1969).
- (5) 横井, 鳥脇, 福村: " 標本化された2値図形のトポロジカルな性質について", 信学論 (D), J 56-D, 11, pp. 662-669 (1973-11).
- (6) 田村秀行: " 細線化法についての諸考察", 信学技報, P R L 75-66 (1975-12).
- (7) 鹿野, 鳥脇, 福村: " 濃淡図形を線図形に変換する一方法 - Wave Propagation Method について -", 信学論 (D), J 55-D, 10, pp. 668-675 (1972-10).
- (8) 横井, 鳥脇, 福村: " 濃淡図形における特徴点の抽出と細線化の一方法", 信学論 (D), J 58-D, 10, pp. 601-608 (1975-10).
- (9) 田村秀行: " 図形の細線化についての比較研究", 情処学 I P 研資, 1-1 (1975-06).

- (1 0) 成瀬, 鳥脇, 福村: "濃淡画像の細線化手法の比較研究",
信学論 (D), J 6 0 - D, 1 2, p p. 1 0 9 3 - 1 1
0 0 (1 9 7 7 - 1 2).
- (1 1) 鶴岡, 木村, 吉村, 横井, 三宅: "ディジタル図形の一細
線化法と手書き文字認識システムへの応用", 信学論 (D
) , J 6 6 - D, 5 , p p. 5 2 5 - 5 3 2 (1 9 8 3 -
0 5).
- (1 2) 鶴岡, 木村, 吉村, 三宅, 横井: "並列処理によるディジ
タル2値図形の細線化アルゴリズム", 昭55, 信学総全
大, 1 2 4 9 (1 9 7 5 - 0 3).
- (1 3) 木村, 吉村, 三宅, 市川: "ストローク構造解析法による
自由手書き片仮名文字認識", 信学論 (D), J 6 2 - D,
p p. 1 6 - 2 3 (1 9 7 9 - 0 1).
- (1 4) 木村文隆: "構造解析的手法による手書き文字認識の研究
", 名古屋大学博士論文.

第3章 文 献

- (1) 萩田, 内藤, 増田: "大局的・局所的方向寄与度密度特徴に
よる手書き漢字認識方式", 信学論 (D), J 6 6 - D, 6 ,
p p. 7 2 2 - 7 2 9 (1 9 8 3 - 0 6).
- (2) 裴, 森下, 燕山, 伊崎, 山本: "手書き漢字認識におけるテ
ンプレート複数化の検討", 信学技報, P R L 8 1 - 4 2 (1
9 8 1 - 0 9).

- (3) 岡 隆一: “セル特徴を用いた手書き漢字の認識”, 信学論 (D), J 6 6 - D, 1, p p. 1 7 - 2 4 (1 9 8 3 - 0 1).
- (4) 飯島, 森: “人間の識別能力に迫るOCR'ASPECT/71'”, 日経エレクトロニクス (1 9 7 2 - 0 5. 2 2).
- (5) 栗田, 鶴岡, 横井, 三宅: “加重方向指数ヒストグラムと疑似マハラノビス距離を用いた手書き漢字・ひらがな認識”, 信学技報, P R L 8 2 - 7 9 (1 9 8 3 - 0 1).
- (6) 村瀬, 木村, 吉村, 三宅: “パターン整合法における特性核の改良とその手書き平仮名文字認識への応用”, 信学論 (D), J 6 4 - D, 3, p p. 2 7 6 - 2 8 3 (1 9 8 1 - 0 3).
- (7) 堀尾, 吉村, 三宅, 福村, 市川: “文脈情報のない場合の自由手書き片仮名文字の可能性について”, 昭52東海支部連大, 3 0 7 (1 9 7 7 - 0 9).
- (8) 飯田, 小森: “人間の文字認識能力の評価 一手書き片仮名, 英数字に対する認識能力について”, 信学論 (D), J 6 7 - D, 3, p p. 2 5 7 - 2 6 4 (1 9 8 4 - 0 3).
- (9) 鶴岡, 木村, 吉村, 横井, 三宅: “ディジタル図形の一細線化法と手書き文字認識システムへの応用”, 信学論 (D), J 6 6 - D, 5, p p. 5 2 5 - 5 3 2 (1 9 8 3 - 0 5).
- (1 0) 木村, 吉村, 三宅, 市川: “ストローク構造解析法による自由手書き片仮名文字認識”, 信学論 (D), J 6 2 - D, 1, p p. 1 6 - 2 3 (1 9 7 9 - 0 1).
- (1 1) 渡辺, 吉村, 三宅, 市川: “ストローク抽出法による自由

手書き片仮名文字認識(Ⅱ)", 信学技報, PRL78-14(1978-06).

(12) 鶴岡, 木村, 横井, 三宅: "細線化された手書き文字図形の整形", 昭55東海支部連大, 362(1980-10).

第4章 文 献

(1) 橋本新一郎: "文字認識概論", 電気通信協会, P36-39, (1982-03).

(2) 飯島, 森: "人間の識別能力に迫るOCR'ASPECT/71", 日経エレクトロニクス, 5-22, pp66-88(1972-05.22).

(3) 小森, 川谷, 石川, 飯田: "特徴集積による手書き片仮名文字の認識", 信学論(D), J63-D, 11, pp.962-969(1980-11).

(4) 森, 山本: "場の効果法による特徴抽出", 信学論(D), 57-D, 5, pp.308-315(1974-05).

(5) 栗田 昌徳: "加重方向指数ヒストグラムと疑似マハラノビス距離を用いた手書き漢字・ひらがな認識", 三重大学修士論文(1983-03).

(6) 例えば, 竹内, 柳井: "多変量解析の基礎", 東洋経済新聞新報社, pp.283-285(1972-08).

(7) K. Fukunaga: "Introduction to Statistical Pattern Re-

- cognition", Academic Press, New York and London, (1972).
- (8) 栗田, 鶴岡, 横井, 三宅: "加重方向指数ヒストグラムと疑似マハラノビス距離を用いた手書き漢字・ひらがな認識", 信学技報, P R L 8 2 - 7 9 (1 9 8 3 - 0 1).
- (9) 岡 隆一: "セル特徴を用いた手書き漢字の認識", 信学論 (D), J 6 6 - D, 1, p p. 1 7 - 2 4 (1 9 8 3 - 0 1).
- (1 0) 齊藤, 山田, 山本: "手書き漢字の方向パターンマッチング法による解析", 信学論 (D), J 6 5 - D, 5, p p. 5 5 0 - 5 5 7 (1 9 8 2 - 0 5).
- (1 1) 横井, 鳥脇, 福村: "標本化された2値図形のトポロジカルな性質について", 信学論 (D), J 5 6 - D, 1 1, p p. 6 6 2 - 6 6 9 (1 9 7 3 - 1 1).

第5章 文 献

- (1) 木村, 吉村, 三宅, 市川: "ストローク構造解析法による自由手書き片仮名文字認識", 信学論 (D), J 6 2 - D, 1, P P. 1 6 - 2 3 (1 9 7 9 - 0 1).
- (2) 鶴岡, 村瀬, 木村, 横井, 三宅: "人間の字種識別基準を用いた自由手書き片仮名文字認識", 信学論 (D), J 6 8 - D, 4, P P. 7 8 1 - 7 8 8 (1 9 8 5 - 0 4).
- (3) 吉村, 木村: "手書き文字の個性が現れる特性の機械的計測化とその解析", 信学論 (D), J 6 3 - D, 9, P P. 7

95-902 (1980-09).

- (4) 吉村, 木村, 吉村: "手書き文字認識における個人用テンプレートの有効性について", 信学論(D), J66-D, 4, pp. 454-455 (1983-04).
- (5) 内藤, 増田: "個人性に着目した手書き漢字認識", 信学論(D), J67-D, 4, pp. 480-487 (1984-04).
- (6) 栗田, 鶴岡, 横井, 三宅: "加重方向指数ヒストグラムと疑似マハラノビス距離を用いた手書き漢字・ひらがな認識", 信学技報, PRL82-79 (1983-01).
- (7) 鶴岡, 栗田, 原田, 木村, 三宅: "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", 信学論(D), J70-D, 7, pp. 1390-1397 (1987-07).
- (8) 森田, 鶴岡, 木村, 三宅: "筆記者に適応した自由手書き文字認識(I)", 信学技報, PRL84-23 (1984-07).
- (9) R.O.Duda and P.E.Hart: "Pattern classification and scene analysis", pp.82, A WILEY-INTERSCIENCE PUB. (1973).
- (10) 鶴岡, 木村, 三宅: "手書き文字認識における個性ベクトルの抽出法", 昭61東海支部連大, 571 (1986-10).

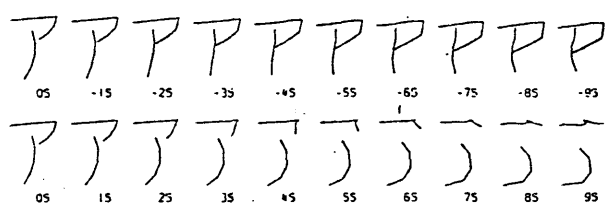
第6章 文献

- (1) 鶴岡, 森田, 木村, 三宅: " 筆記者に対して適応機能を持った自由手書き文字認識", 信学論 (D), J 7 0 - D, 1 0 (1 9 8 7 - 1 0).
- (2) 栗田, 鶴岡, 横井, 三宅: " 加重方向指数ヒストグラムと疑似マハラノビス距離を用いた手書き漢字・ひらがな認識", 信学技報, P R L 8 2 - 7 9 (1 9 8 3 - 0 1).
- (3) 森田, 鶴岡, 木村, 三宅: " 筆記者に適応した自由手書き文字認識 (III) ", 信学技報, P R L 8 5 - 7 8 (1 9 8 6 - 0 3).
- (4) 木村文隆, 西晃, 鶴岡信治, 三宅康二: " クラスタリングに基づく手書き漢字の大分類", 信学論 (D), J 7 0 - D, No. 1, pp. 148-154 (1 9 8 7 - 0 1).

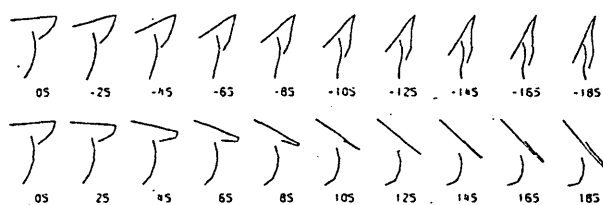
付 録

1. 変形文字パターンの例…………… 193
2. 字種ごとの主成分の図表現の例…………… 199
3. 加重方向指数ヒストグラムの例…………… 204
4. 主なプログラムの説明とソースリスト…………… 209

D=1



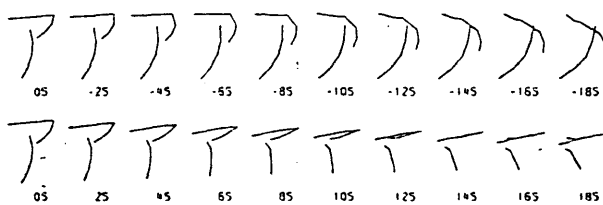
D=7



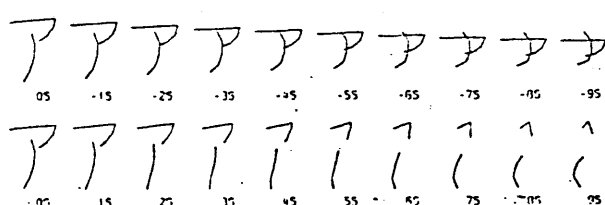
D=2



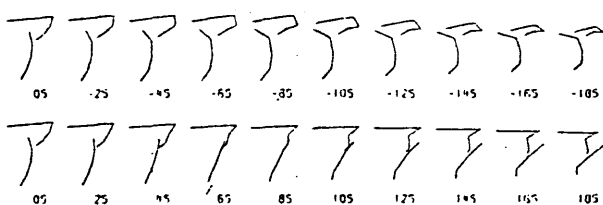
D=8



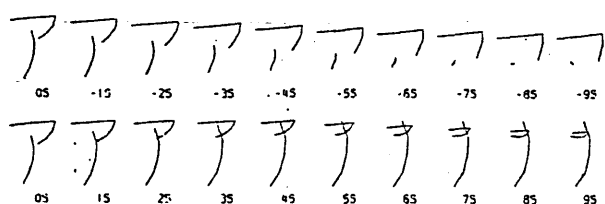
D=3



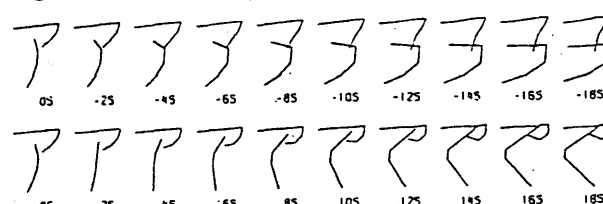
D=9



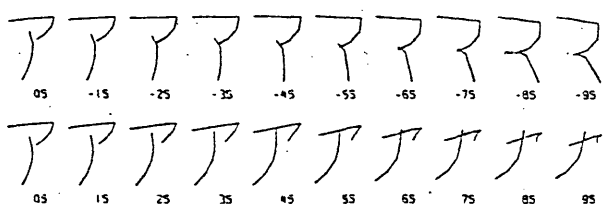
D=4



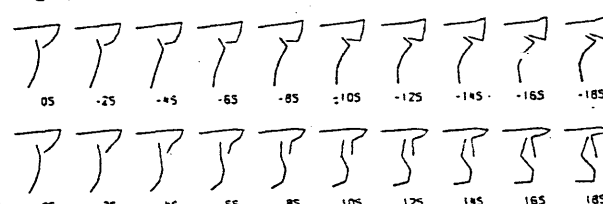
D=10



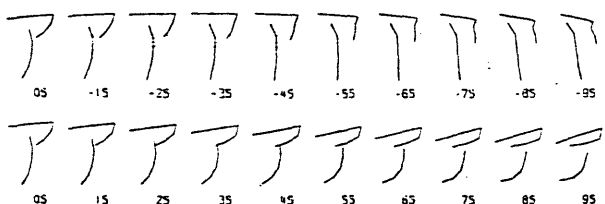
D=5



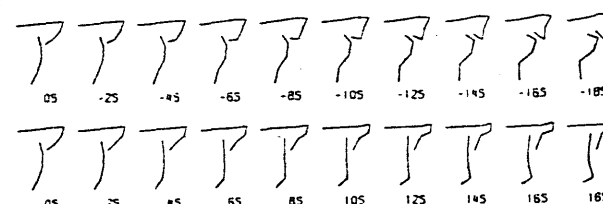
D=11



D=6



D=12

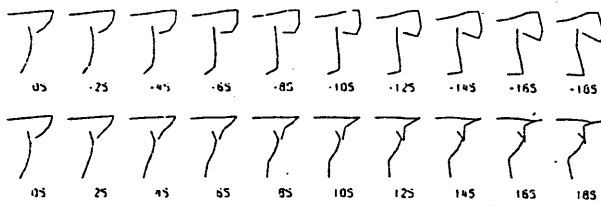


付録1 変形文字パターンの例(1)

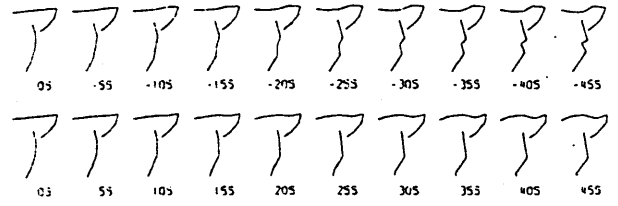
(Dは主成分軸番号, 上段は負方向, 下段は正方向への変形を示す)

(数字は平均から分散の何倍離れているかを表す)

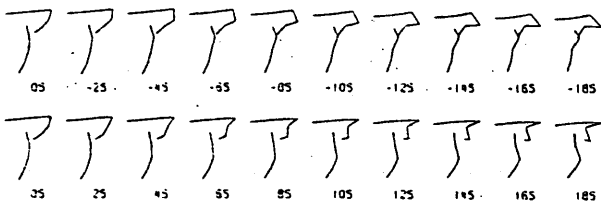
0-13



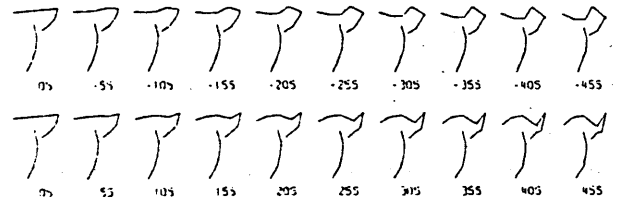
0-19



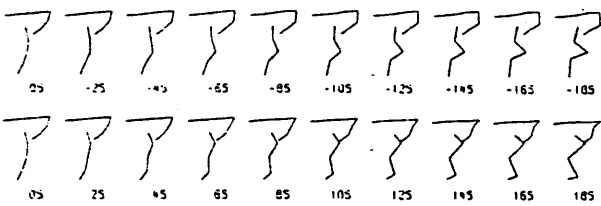
0-14



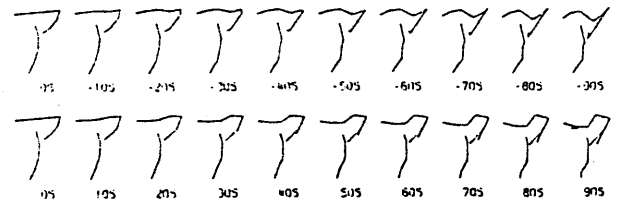
0-20



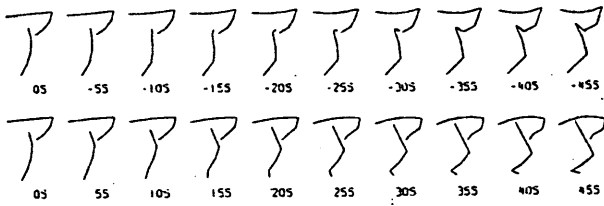
0-15



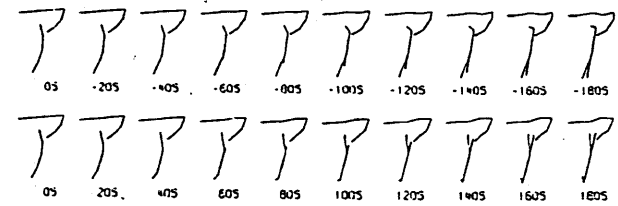
0-21



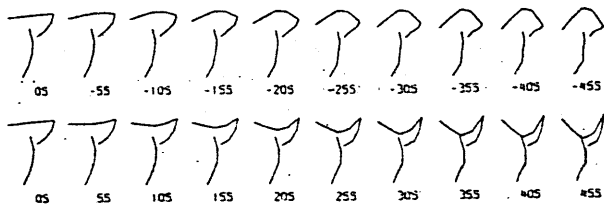
0-16



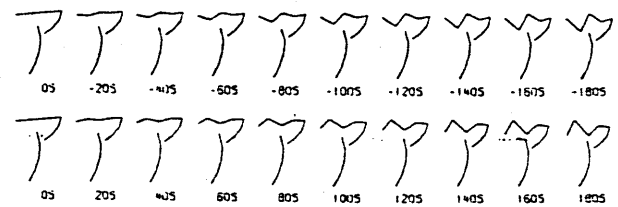
0-22



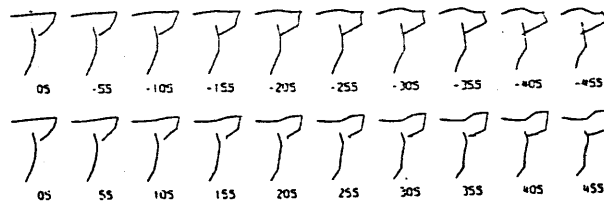
0-17



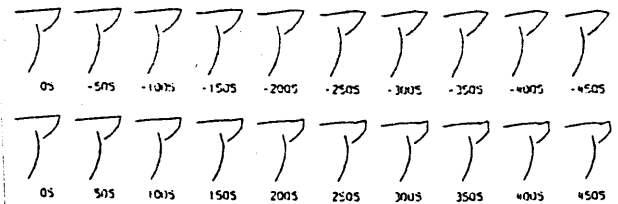
0-23



0-18

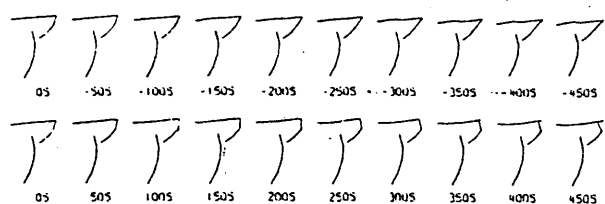


0-24

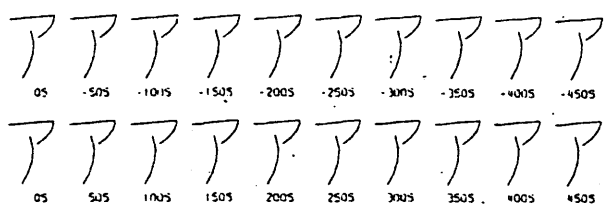


付録1 変形文字パターンの例(2)

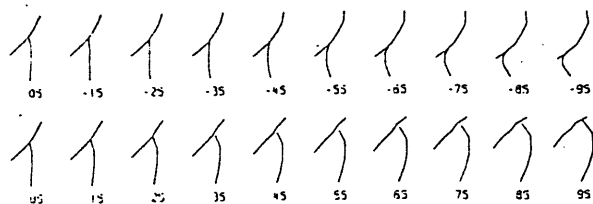
0-25



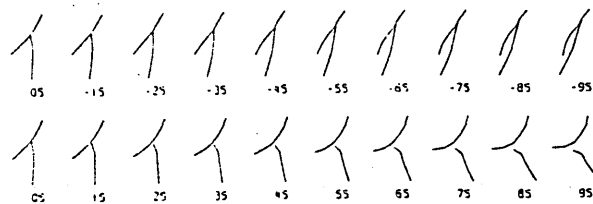
0-26



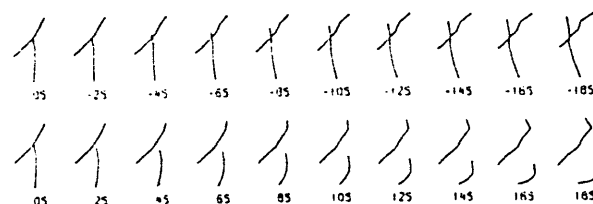
0-4



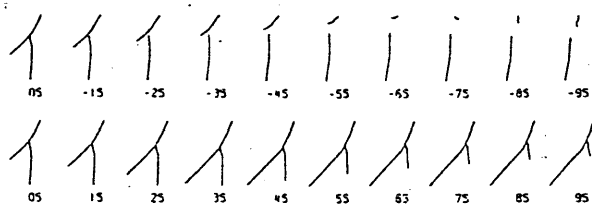
0-5



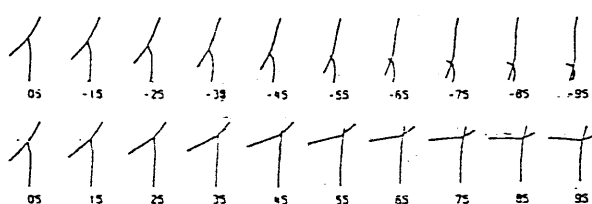
0-6



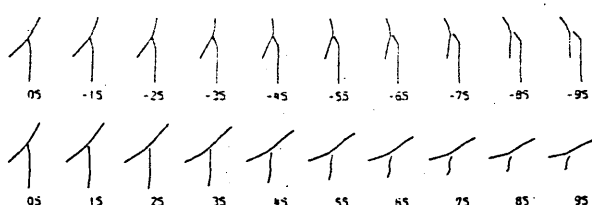
0-1



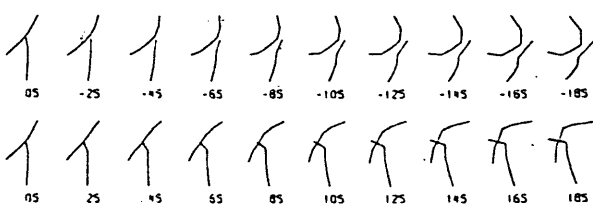
0-2



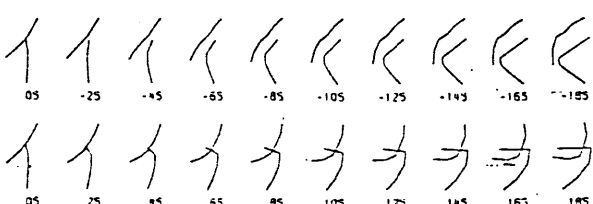
0-3



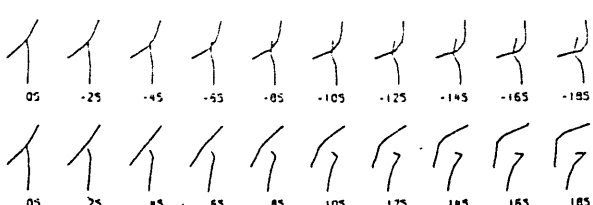
0-7



0-8

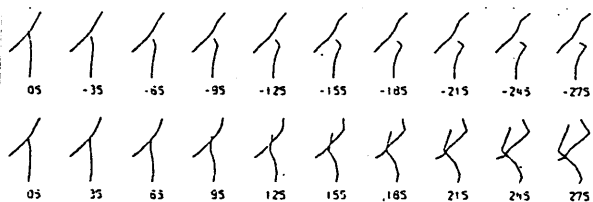


0-9

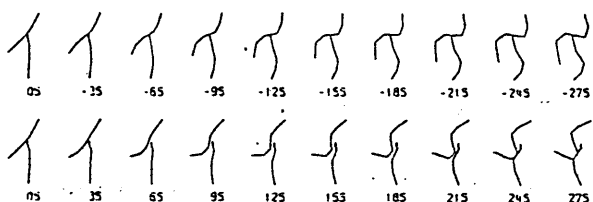


付録1 変形文字パターンの例(3)

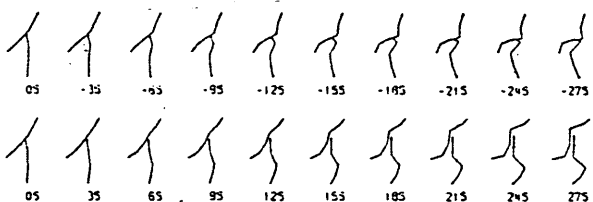
D-10



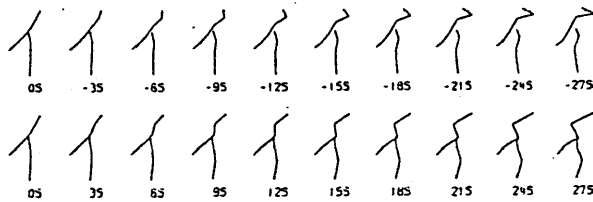
D-11



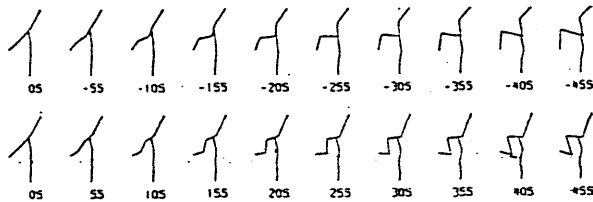
D-12



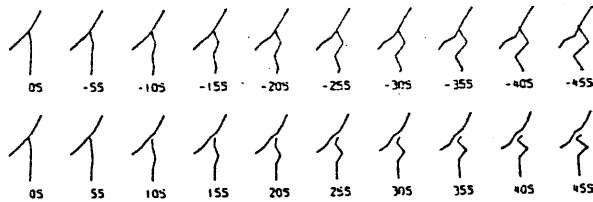
D-13



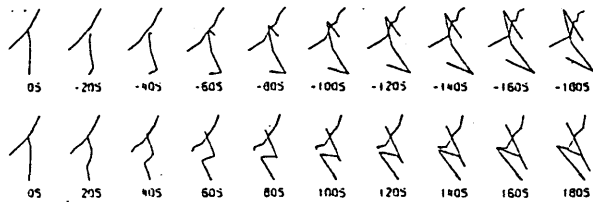
D-14



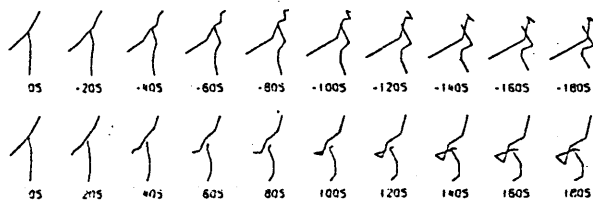
D-15



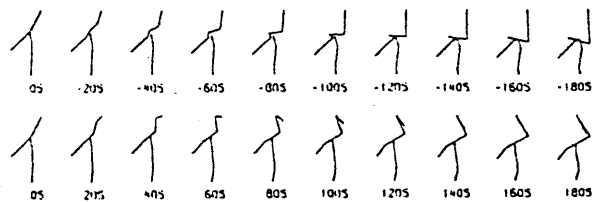
D-16



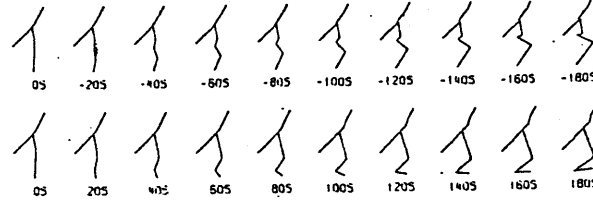
D-17



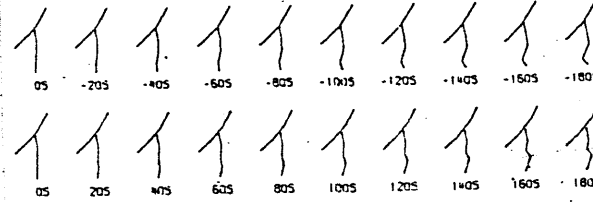
D-18



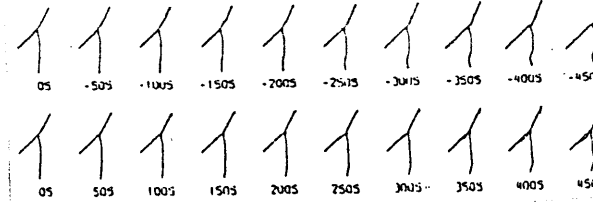
D-19



D-20



D-21



付録1 変形文字パターンの例(4)

0-22

イ イ イ イ イ イ イ イ イ イ
05 505 1005 1505 2005 2505 3005 3505 4005 4505

イ イ イ イ イ イ イ イ イ イ
05 505 1005 1505 2005 2505 3005 3505 4005 4505

0-4

ウ ウ ウ ウ ウ ウ ウ ウ ウ ウ
05 15 25 35 45 55 65 75 85 95

ウ ウ ウ ウ ウ ウ ウ ウ ウ ウ
05 15 25 35 45 55 65 75 85 95

0-5

ウ ウ ウ ウ ウ ウ ウ ウ ウ ウ
05 25 45 65 85 105 125 145 165 185

ウ ウ ウ ウ ウ ウ ウ ウ ウ ウ
05 25 45 65 85 105 125 145 165 185

0-6

ウ ウ ウ ウ ウ ウ ウ ウ ウ ウ
05 15 25 35 45 55 65 75 85 95

ウ ウ ウ ウ ウ ウ ウ ウ ウ ウ
05 15 25 35 45 55 65 75 85 95

0-1

ウ ウ ウ ウ ウ ウ ウ ウ ウ ウ
05 15 25 35 45 55 65 75 85 95

ウ ウ ウ ウ ウ ウ ウ ウ ウ ウ
05 15 25 35 45 55 65 75 85 95

0-2

ウ ウ ウ ウ ウ ウ ウ ウ ウ ウ
05 15 25 35 45 55 65 75 85 95

ウ ウ ウ ウ ウ ウ ウ ウ ウ ウ
05 15 25 35 45 55 65 75 85 95

0-3

ウ ウ ウ ウ ウ ウ ウ ウ ウ ウ
05 15 25 35 45 55 65 75 85 95

ウ ウ ウ ウ ウ ウ ウ ウ ウ ウ
05 15 25 35 45 55 65 75 85 95

0-7

ウ ウ ウ ウ ウ ウ ウ ウ ウ ウ
05 15 25 35 45 55 65 75 85 95

ウ ウ ウ ウ ウ ウ ウ ウ ウ ウ
05 15 25 35 45 55 65 75 85 95

0-8

ウ ウ ウ ウ ウ ウ ウ ウ ウ ウ
05 25 45 65 85 105 125 145 165 185

ウ ウ ウ ウ ウ ウ ウ ウ ウ ウ
05 25 45 65 85 105 125 145 165 185

0-9

ウ ウ ウ ウ ウ ウ ウ ウ ウ ウ
05 25 45 65 85 105 125 145 165 185

ウ ウ ウ ウ ウ ウ ウ ウ ウ ウ
05 25 45 65 85 105 125 145 165 185

付録1 変形文字パターンの例(5)

D-10

ウ	ウ	ウ	ウ	ウ	ウ	ウ	ウ	ウ	ウ
05	-25	-45	-65	-85	-105	-125	-145	-165	-185
ウ	ウ	ウ	ウ	ウ	ウ	ウ	ウ	ウ	ウ
05	25	45	65	85	105	125	145	165	185

D-11

ウ	ウ	ウ	ウ	ウ	ウ	ウ	ウ	ウ	ウ
05	-25	-45	-65	-85	-105	-125	-145	-165	-185
ウ	ウ	ウ	ウ	ウ	ウ	ウ	ウ	ウ	ウ
05	25	45	65	85	105	125	145	165	185

D-12

ウ	ウ	ウ	ウ	ウ	ウ	ウ	ウ	ウ	ウ
05	-25	-45	-65	-85	-105	-125	-145	-165	-185
ウ	ウ	ウ	ウ	ウ	ウ	ウ	ウ	ウ	ウ
05	25	45	65	85	105	125	145	165	185

D-13

ウ	ウ	ウ	ウ	ウ	ウ	ウ	ウ	ウ	ウ
05	-25	-45	-65	-85	-105	-125	-145	-165	-185
ウ	ウ	ウ	ウ	ウ	ウ	ウ	ウ	ウ	ウ
05	25	45	65	85	105	125	145	165	185

D-14

ウ	ウ	ウ	ウ	ウ	ウ	ウ	ウ	ウ	ウ
05	-25	-45	-65	-85	-105	-125	-145	-165	-185
ウ	ウ	ウ	ウ	ウ	ウ	ウ	ウ	ウ	ウ
05	25	45	65	85	105	125	145	165	185

D-15

ウ	ウ	ウ	ウ	ウ	ウ	ウ	ウ	ウ	ウ
05	-35	-65	-95	-125	-155	-185	-215	-245	-275
ウ	ウ	ウ	ウ	ウ	ウ	ウ	ウ	ウ	ウ
05	35	65	95	125	155	185	215	245	275

D-16

ウ	ウ	ウ	ウ	ウ	ウ	ウ	ウ	ウ	ウ
05	-35	-65	-95	-125	-155	-185	-215	-245	-275
ウ	ウ	ウ	ウ	ウ	ウ	ウ	ウ	ウ	ウ
05	35	65	95	125	155	185	215	245	275

D-17

ウ	ウ	ウ	ウ	ウ	ウ	ウ	ウ	ウ	ウ
05	-35	-65	-95	-125	-155	-185	-215	-245	-275
ウ	ウ	ウ	ウ	ウ	ウ	ウ	ウ	ウ	ウ
05	35	65	95	125	155	185	215	245	275

D-18

ウ	ウ	ウ	ウ	ウ	ウ	ウ	ウ	ウ	ウ
05	-35	-65	-95	-125	-155	-185	-215	-245	-275
ウ	ウ	ウ	ウ	ウ	ウ	ウ	ウ	ウ	ウ
05	35	65	95	125	155	185	215	245	275

D-19

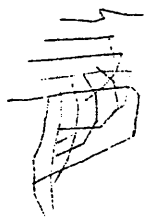
ウ	ウ	ウ	ウ	ウ	ウ	ウ	ウ	ウ	ウ
05	-35	-65	-95	-125	-155	-185	-215	-245	-275
ウ	ウ	ウ	ウ	ウ	ウ	ウ	ウ	ウ	ウ
05	35	65	95	125	155	185	215	245	275

D-20

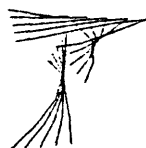
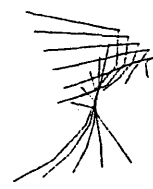
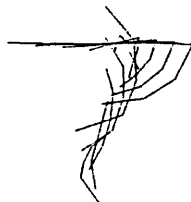
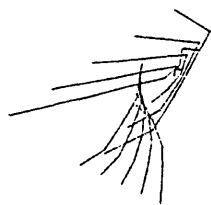
ウ	ウ	ウ	ウ	ウ	ウ	ウ	ウ	ウ	ウ
05	-55	-105	-155	-205	-255	-305	-355	-405	-455
ウ	ウ	ウ	ウ	ウ	ウ	ウ	ウ	ウ	ウ
05	55	105	155	205	255	305	355	405	455

D-21

ウ	ウ	ウ	ウ	ウ	ウ	ウ	ウ	ウ	ウ
05	-55	-105	-155	-205	-255	-305	-355	-405	-455
ウ	ウ	ウ	ウ	ウ	ウ	ウ	ウ	ウ	ウ
05	55	105	155	205	255	305	355	405	455



$S = 0.11E+00$ $S = 0.10E+00$ $S = 0.90E-01$ $S = 0.78E-01$ $S = 0.68E-01$
 $D = 1$ $D = 2$ $D = 3$ $D = 4$ $D = 5$



$S = 0.48E-01$ $S = 0.44E-01$ $S = 0.43E-01$ $S = 0.33E-01$ $S = 0.31E-01$
 $D = 6$ $D = 7$ $D = 8$ $D = 9$ $D = 10$



$S = 0.17E-01$ $S = 0.15E-01$ $S = 0.14E-01$ $S = 0.11E-01$ $S = 0.95E-02$
 $D = 11$ $D = 12$ $D = 13$ $D = 14$ $D = 15$

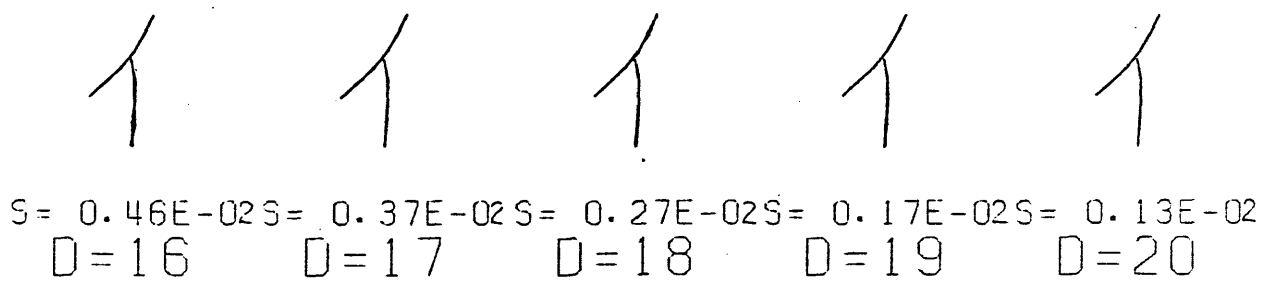
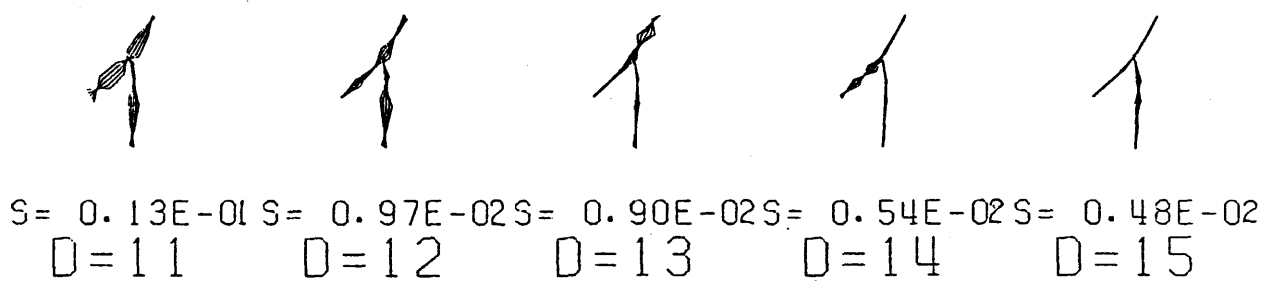
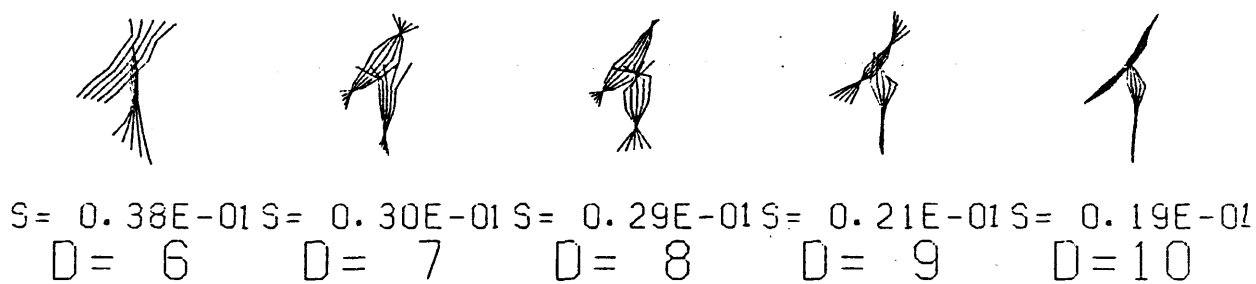
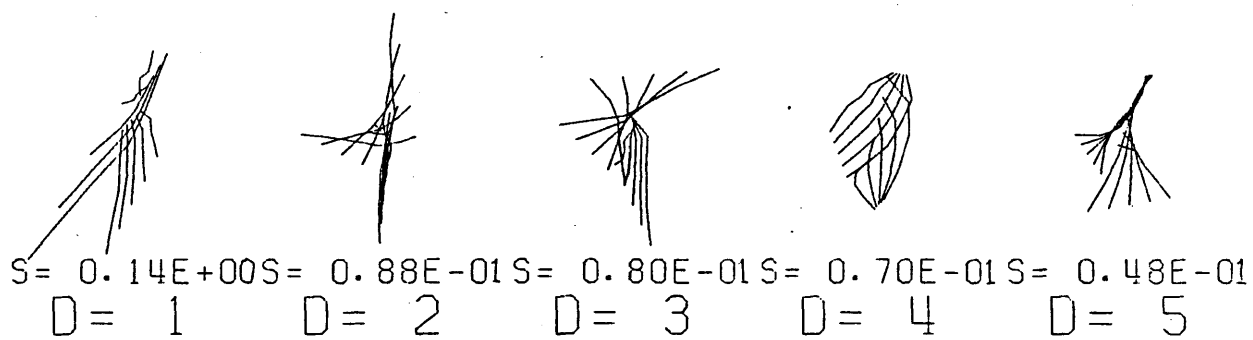


$S = 0.66E-02$ $S = 0.58E-02$ $S = 0.50E-02$ $S = 0.38E-02$ $S = 0.37E-02$
 $D = 16$ $D = 17$ $D = 18$ $D = 19$ $D = 20$

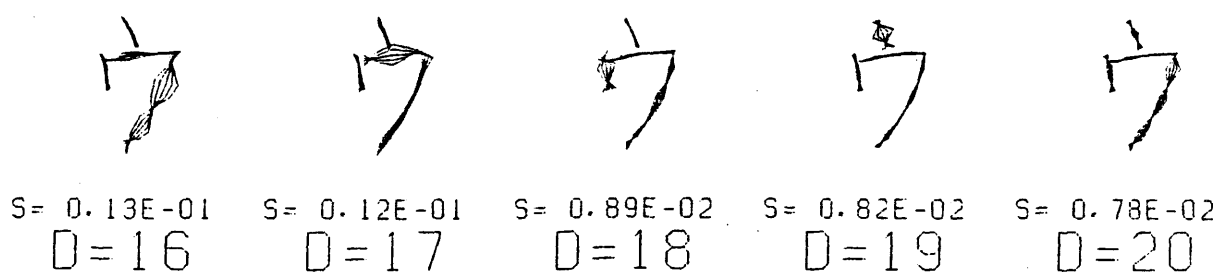
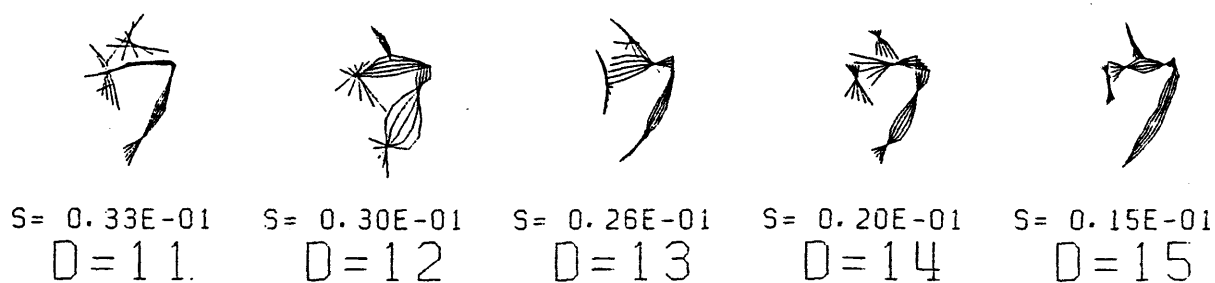
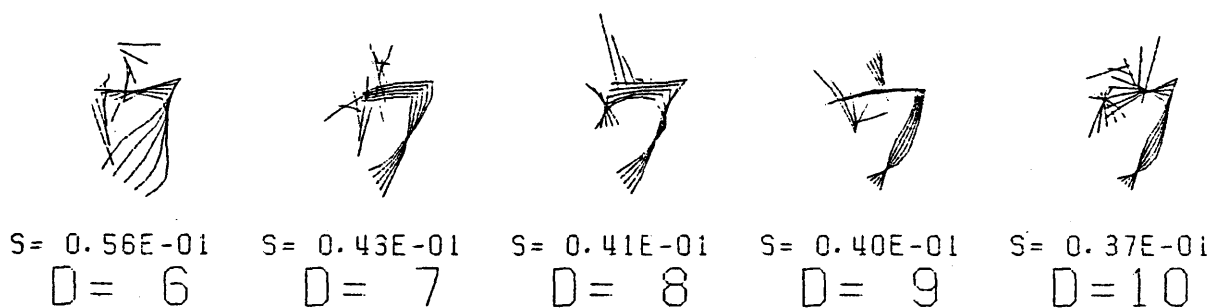
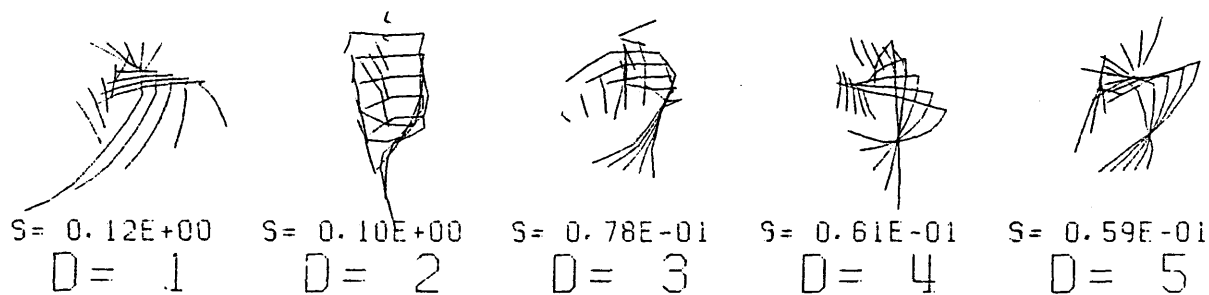
付録2 字種ごとの主成分の図表現の例(1)

(Dは主成分軸番号, Sは分散の大きさを示す.

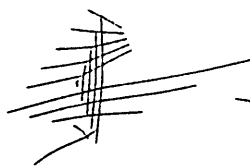
変形文字パターンは分散の5倍ごとに表示.)



付録2 字種ごとの主成分の図表現の例(2)



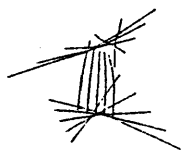
付録2 字種ごとの主成分の図表現の例(3)



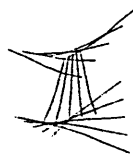
S= 0.12E+00
D= 1



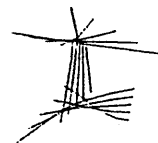
S= 0.82E-01
D= 2



S= 0.70E-01
D= 3



S= 0.63E-01
D= 4



S= 0.56E-01
D= 5



S= 0.44E-01
D= 6



S= 0.34E-01
D= 7



S= 0.31E-01
D= 8



S= 0.24E-01
D= 9



S= 0.22E-01
D=10



S= 0.21E-01
D=11



S= 0.14E-01
D=12



S= 0.12E-01
D=13



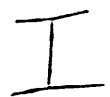
S= 0.10E-01
D=14



S= 0.92E-02
D=15



S= 0.49E-02
D=16



S= 0.32E-02
D=17



S= 0.30E-02
D=18

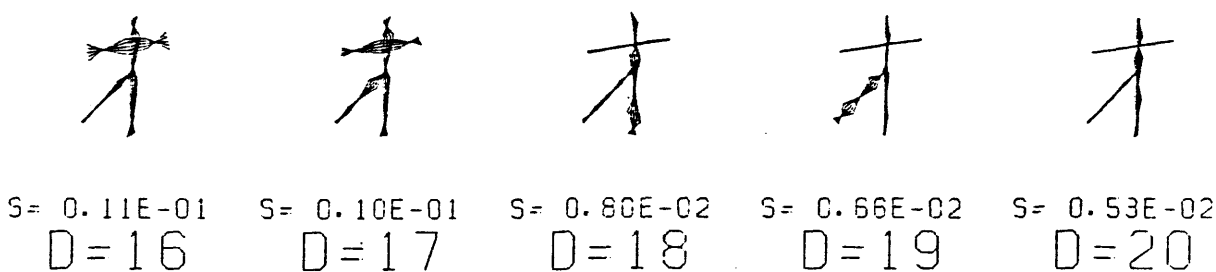
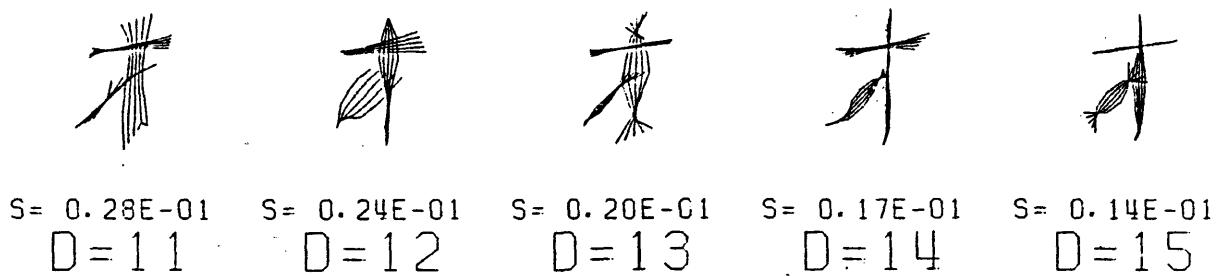
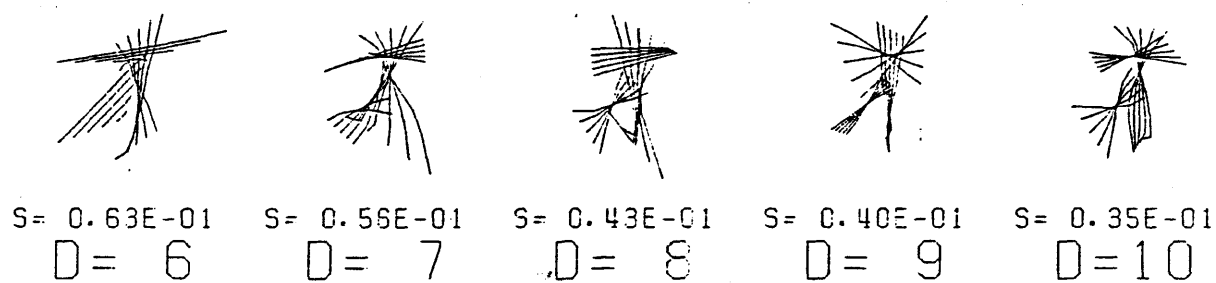
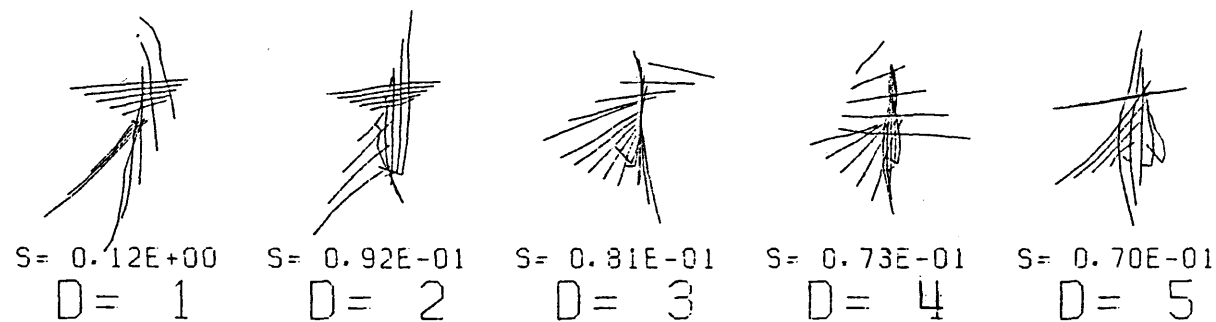


S= 0.28E-02
D=19

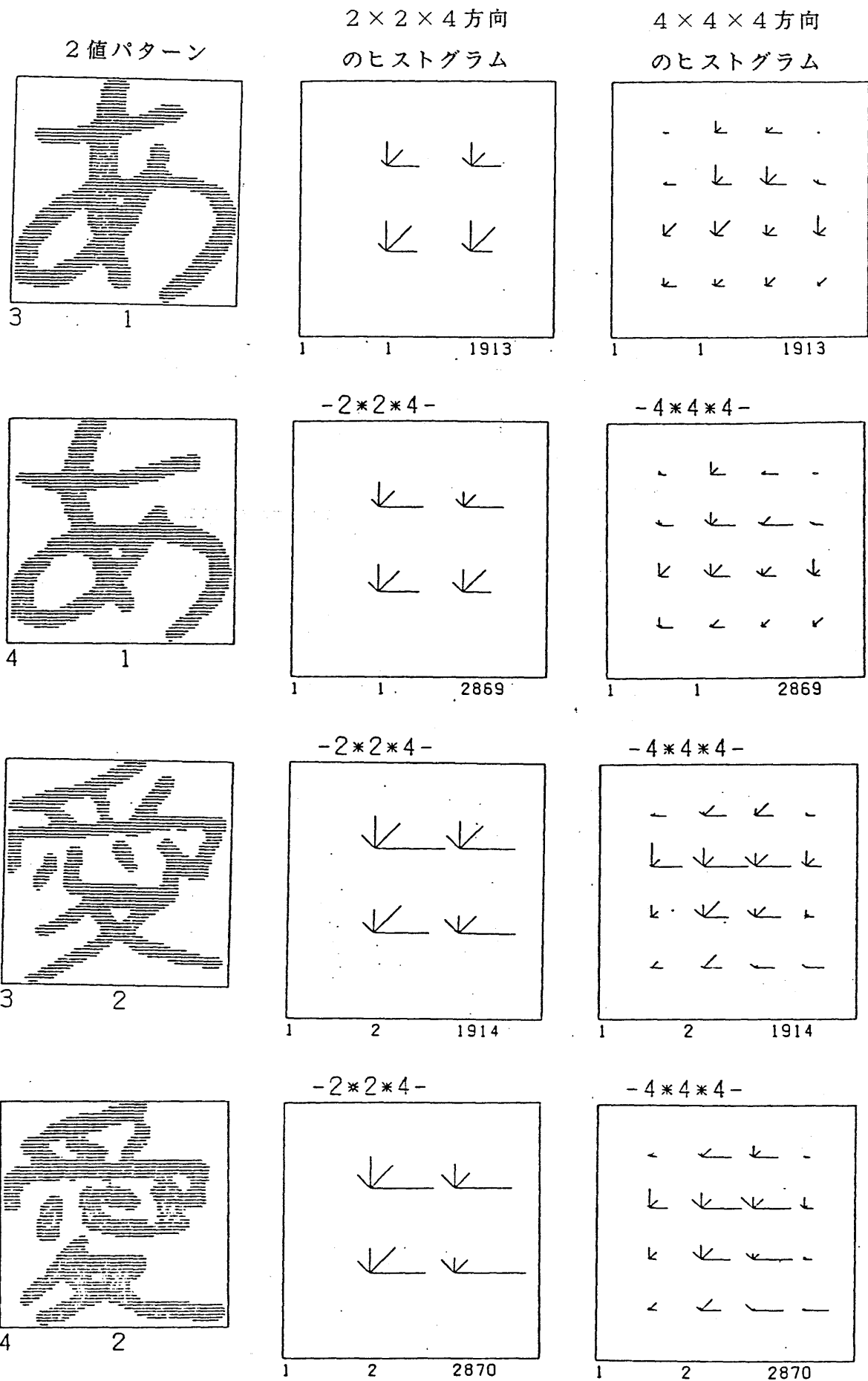


S= 0.25E-02
D=20

付録2 字種ごとの主成分の図表現の例(4)

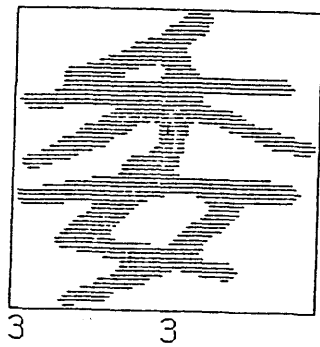


付録2 字種ごとの主成分の図表現の例(5)



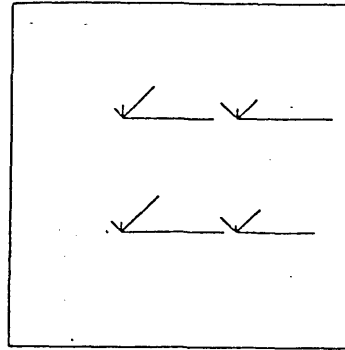
付録3 加重方向指数ヒストグラムの例(1)

(1つの小領域に4方向の方向指数のヒストグラムを表示)



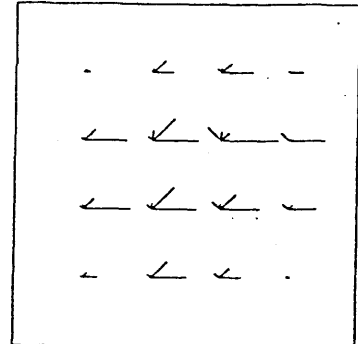
3 3

-2*2*4-

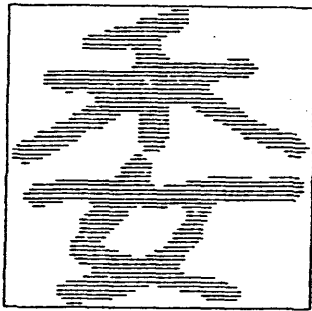


1 3 1915

-4*4*4-

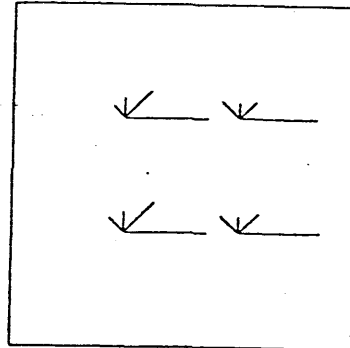


1 3 1915



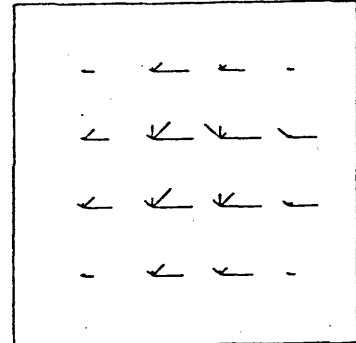
4 3

-2*2*4-

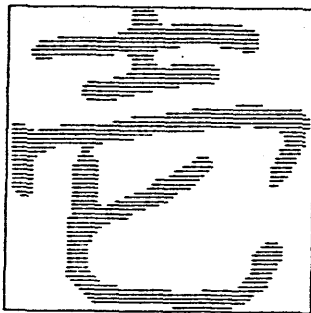


1 3 2871

-4*4*4-

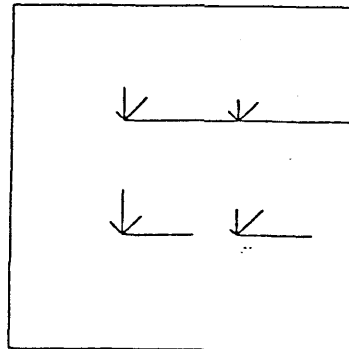


1 3 2871



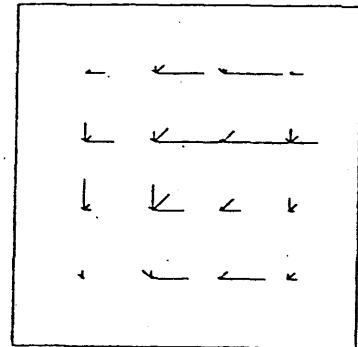
3 4

-2*2*4-

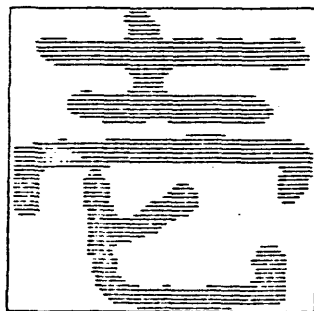


1 4 1916

-4*4*4-

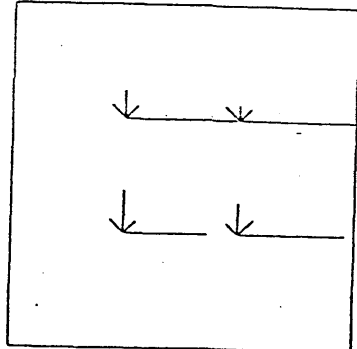


1 4 1916



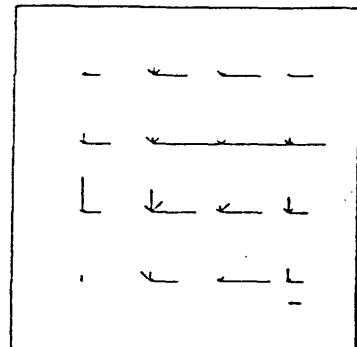
4 4

-2*2*4-



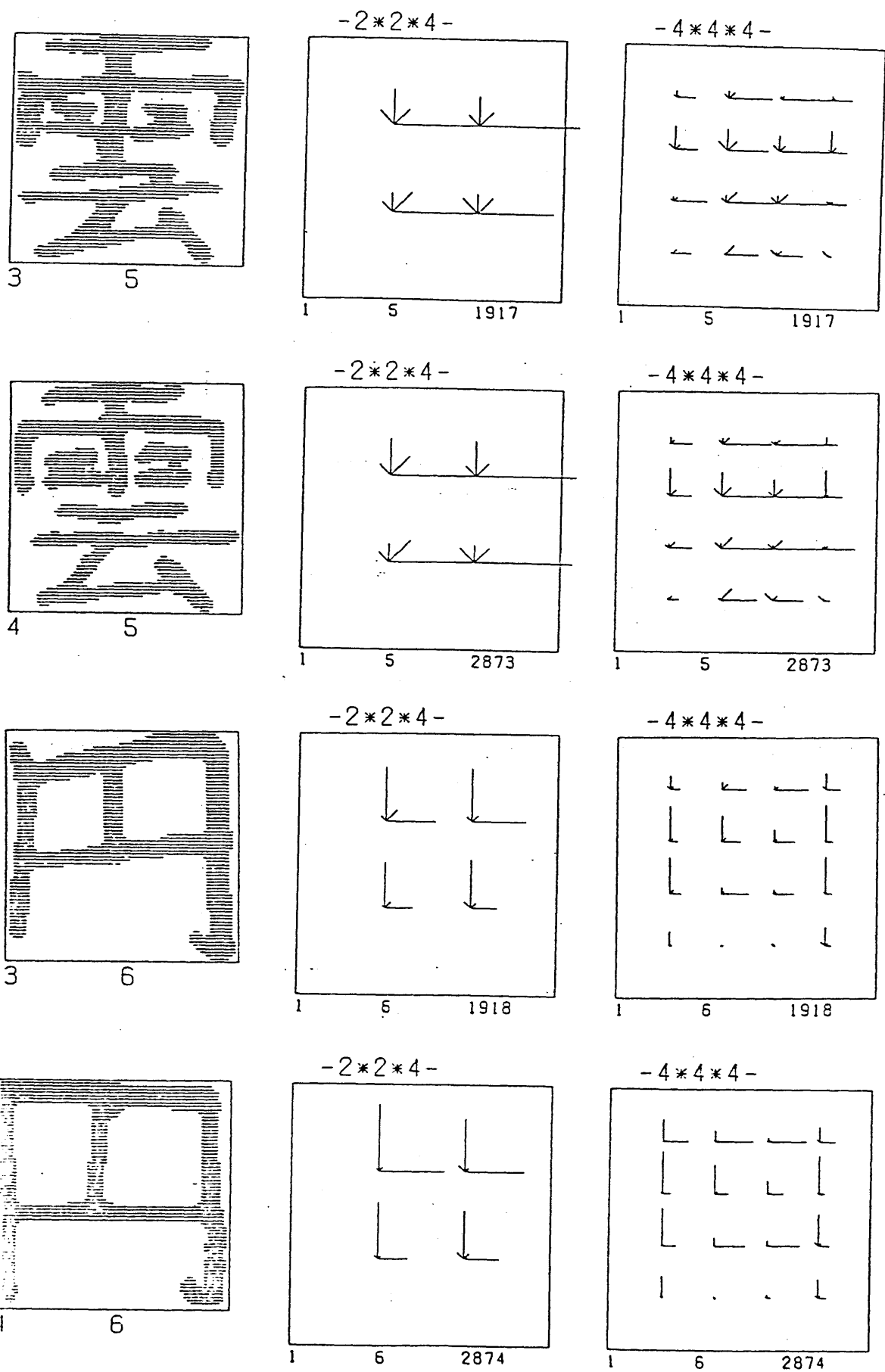
1 4 2872

-4*4*4-

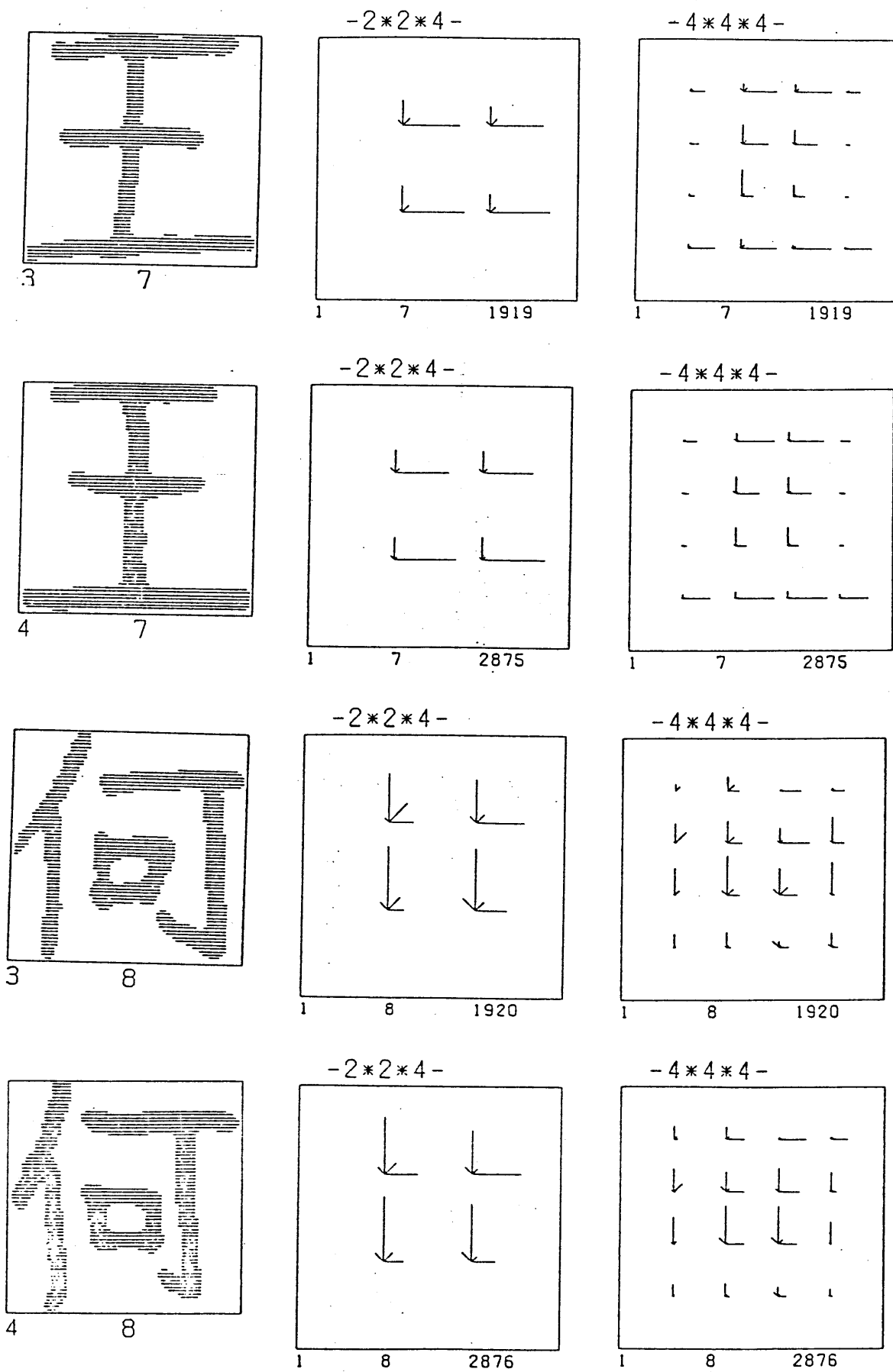


1 4 2872

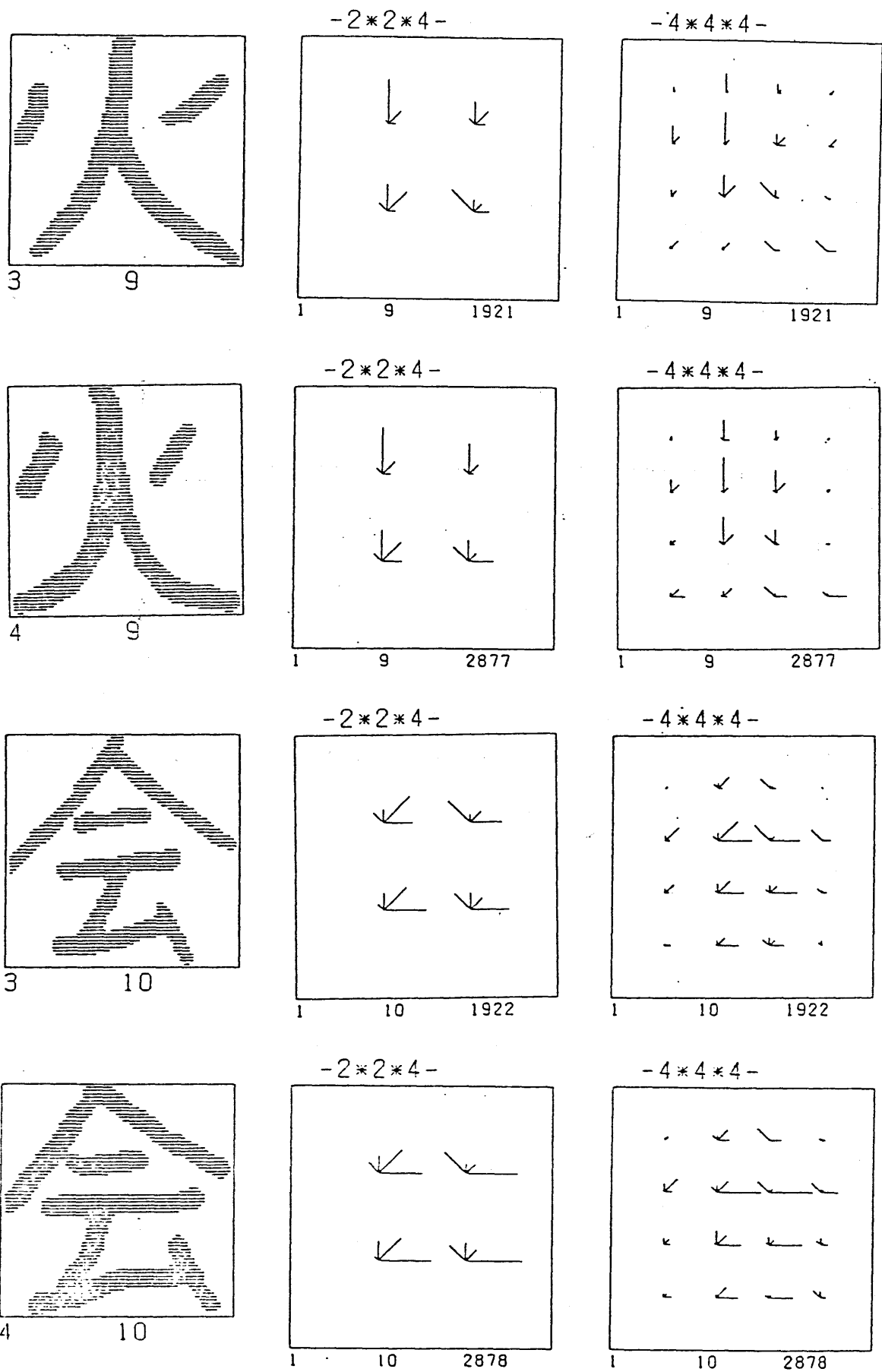
付録3 加重方向指数ヒストグラムの例(2)



付録3 加重方向指数ヒストグラムの例(3)



付録3 加重方向指数ヒストグラムの例(4)



付録3 加重方向指数ヒストグラムの例(5)

SMOOTH Smoothing

非線形平滑化

作成 鶴岡信治	1985年 7月
形式 サブルーチン	言語: FORTRAN サイズ: 35行

(1) 概要

2値図形、濃淡画像の非線形平滑化を行う。4近傍に非0要素が3個以上ある0要素は、1要素の平均濃度値をその点の濃度値とする。したがって2値図形において2ビットまでの穴は除去することができ、1ビットの輪郭部のへこみを取り除ける。8近傍に0要素が8つある1要素は、0要素にする。すなわち、孤立点を除去する。

(2) 使用法

CALL SMOOTH(IA, I1, J1, IS, IE, JS, JE, IB, IT)

引数	型と種類	属性	内容
IA	整数型 2次元配列	入力	処理したい画像
I1	整数型	入力	IAの整合寸法 (第1添字の値)
J1	整数型	入力	" (第2添字の値)
IS	整数型	入力	平滑化する範囲 (x方向の下限)

IE	整数型	入力	" (x方向の上限)
JS	整数型	入力	" (y方向の下限)
JE	整数型	入力	" (y方向の上限)
IB	整数型 2次元配列	出力	処理結果の画像
IT	整数型	入力	IAを2値化するしきい値

(3) 計算法

概要で述べたことを並列演算で行う。IA=IBとすると、逐次演算で行うことになり、多少並列演算と結果が異なるが、画像配列の節約になる。

(1) 近傍 (IA→f(x), IB→g(x)とする)

4近傍: $N^4(x) = \{X_1, X_3, X_5, X_7\}$
 $= \{X_k \mid k \in S_1\}$

8近傍: $N^8(x) = \{X_1, X_2, \dots, X_8\}$
 $= \{X_k \mid k \in S\}$

但し, S_1, S はそれぞれ添字集合
 $\{1, 3, 5, 7\}, \{1, 2, \dots, 8\}$

X_4	X_3	X_2
X_5	X	X_1
X_6	X_7	X_8

図1 要素Xとその近傍 X_k

(2) 2値画像 (濃淡画像についても以下のように考える)

$$b(x) = \begin{cases} 1 & \dots f(x) \geq IT \\ 0 & \dots f(x) < IT \end{cases}$$

点集合: $B = \{X \mid b(x)=1\}, V = \{X \mid b(x)=0\}$

(3)穴の除去

$\sum b(x_k) \geq 3$ となる画素については

$$g(x) = \frac{1}{N} \sum_{k \in S1 \cap B} f(x_k)$$

但し、NはS1∩Bを満たす要素の数

(4)孤立点の除去

$\sum_{k \in S} b(x_k) = 0$ となる画素については、 $g(x) = 0$ とする。

(5)その他の画素 $g(x) = f(x)$

(4)子ルーチンなし

NORMM3

Normarize Multi-level Image No.3

多値画像の外接重心合せによる位置・大きさの正規化

作成 鶴岡信治	1985年 7月
形式 サブルーチン	言語: FORTRAN サイズ: 75行

(1)概要

文字の外接長方形と文字の重心を求め、外接長方形の中心に文字の重心が位置するように位置・大きさの正規化をする。

(2)使用法

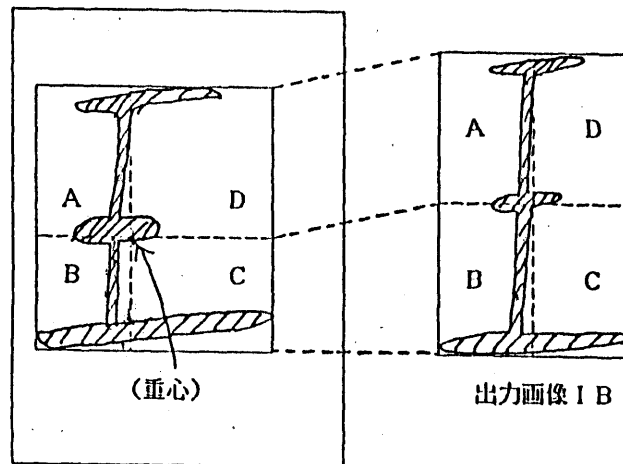
CALL NORMM3(IA,I1,J1,IB,I2,J2,IT,ILL)

引数	型と種類	属性	内容
IA	整数型 2次元配列	入力	正規化前の多値画像
I1	整数型	入力	IAの整合寸法(第1添字の値)
J1	整数型	入力	" (第2添字の値)
IB	整数型	出力	位置・大きさの正規化済みの画像
I2	整数型	入力	IBの整合寸法(第1添字の値)

J2	整数型	入力	" (第2添字の値)
IT	整数型	入力	IAを2値化するしきい値
ILL	整数型	出力	0:正常終了 0以外:図の大きさが求まらない

(3) 計算法

入力画像IAにおいて文字部分の外接長方形、重心を求め、出力画像の中心に文字の重心がくるように、4つの小領域で別々に縦、横の伸縮率を計算し、位置・大きさの正規化をする。



入力画像IA

図1 位置・大きさの正規化

(4) 子ルーチン

NSIZEM (外接長方形を求める)

GCENTM (重心座標を求める)

(5) 備考

従来から外接枠合わせ、重心合わせの正規化法はよく用いられているが、本方法は三重大学で開発された方法であり、両者の中間の性質を示すと考えられる。特に文字線分の飛び出し、つき出し等について有効と考えられる。

PSIZEB

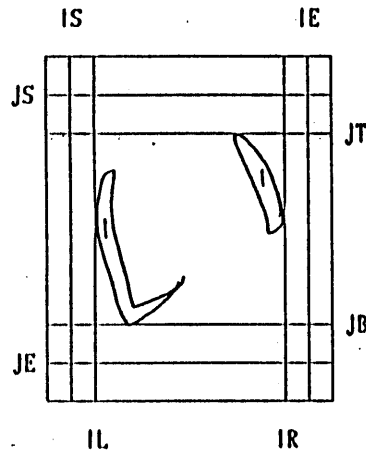
Object Size for binary picture

2値図形について図の大きさ（右端，左端，上端，下端）を求める。

作成 鶴岡信治	1985年 7月
形式 サブルーチン	言語：FORTRAN サイズ：46行

(1) 概要

IA (I1, J1) で与えられた画像の
特定の範囲 (IS, JS), (IE, JE)
内で1である場所の右端，左端，
上端，下端を求める。



(2) 使用法

CALL PSIZEB(IA, I1, J1, IS, IE, JS, JE, IL, IR, JT, JB, ILL)

引数	型と種類	属性	内容
IA	整数型 2次元配列	入力	処理したい画像

I1	整数型	入力	IAの整合寸法（第1添字の値）
J1	整数型	入力	" （第2添字の値）
IS	整数型	入力	IA中の検索する範囲（左端）
IE	整数型	入力	" （右端）
JS	整数型	入力	" （上端）
JE	整数型	入力	" （下端）
IL	整数型	出力	IA中の1要素の左端の位置
IR	整数型	出力	IA中の1要素の右端の位置
JT	整数型	出力	IA中の1要素の上端の位置
JB	整数型	出力	IA中の1要素の下端の位置
ILL	整数型	出力	0: 正常終了 1000: 全部0であり，上端が 求まらない。 2000: 下端が求まらない 3000: 左端が求まらない 4000: 右端が求まらない

(3) 計算法

画像配列IAを左上から走査して、はじめて1となる行を上端とする。
他の端についても同様であり、走査の方向だけが異なる。

(4) 子ルーチン なし

PSIZEM

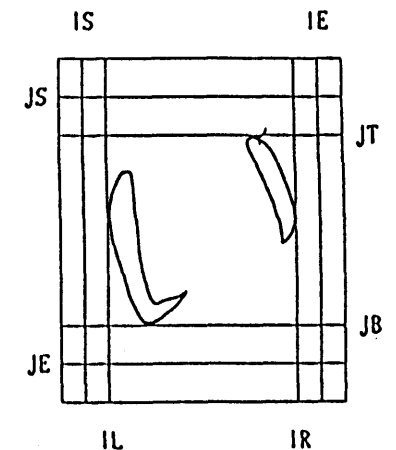
Object Size for multilevel picture

濃淡画像について図の大きさ(右端, 左端, 上端, 下端)を求める。

作成 鶴岡信治	1985年 7月
形式 サブルーチン	言語: FORTRAN サイズ: 46行

(1) 概要

IA (IL, JL) で与えられた画像
の特定の範囲 (IS, JS), (IE, JE)
内でしきい値IT以上である場所の
左端, 右端, 上端, 下端を求める。
PSIZEBの濃淡図形版。



(2) 使用法

CALL PSIZEM(IA, IL, JL, IS, IE, JS, JE, IL, IR, JT, JB, IT, ILL)

引数	型と種類	属性	内容
IA	整数型 2次元配列	入力	処理したい画像

II	整数型	入力	IAの整合寸法（第1添字の値）
JI	整数型	入力	〃 （第2添字の値）
IS	整数型	入力	IA中の検索する範囲（左端）
IE	整数型	入力	〃 （右端）
JS	整数型	入力	〃 （上端）
JE	整数型	入力	〃 （下端）
IL	整数型	出力	IA中の1要素の左端の位置
IR	整数型	出力	IA中の1要素右端の位置
JT	整数型	出力	IA中の1要素上端の位置
JB	整数型	出力	IA中の1要素下端の位置
IT	整数型	入力	IAを2値化するしきい値
ILL	整数型	出力	0:正常終了 1000:全部0であり,上端が 求まらない。 2000:下端が求まらない 3000:左端が求まらない 4000:右端が求まらない

(3) 計算法

画像配列IAを左上から走査して,はじめてしきい値IT以上となる行を上端とする。

他の端についても同様であり,走査の方向だけが異なる。

(4) 子ルーチン なし

NORMB3

Normarize Binary Image No.3

2値画像の外接枠・重心合わせによる位置・大きさの正規化

作成	鶴岡信治	1985年 7月
形式	サブルーチン	言語: FORTRAN サイズ: 75行

(1) 概要

文字の外接長方形と文字の重心を求め、外接長方形の中心に文字の重心が位置するように位置・大きさの正規化をする。

(2) 使用法

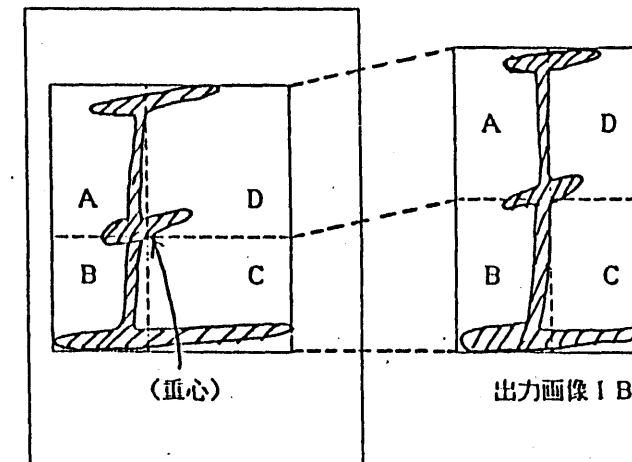
CALL NORMB3(IA,II,J1,IB,I2,J2,ILL)

引数	型と種類	属性	内容
IA	整数型 2次元配列	入力	正規化前の2値画像
II	整数型	入力	IAの整合寸法 (第1添字の値)
J1	整数型	入力	" (第2添字の値)
IB	整数型 2次元配列	出力	位置・大きさの正規化済の画像
I2	整数型	入力	IBの整合寸法 (第1添字の値)

J2	整数型	入力	" (第2添字の値)
ILL	整数型	出力	0: 正常終了 0以外: 図の大きさが求まらない

(3) 計算法

入力画像IAにおいて文字部分の外接長方形、重心を求め、出力画像の中心に文字の重心がくるように、4つの小領域で別々に縦、横の伸縮率を計算し、位置・大きさの正規化をする。



入力画像 I A

図1 位置・大きさの正規化

(4) 子ルーチン

NSIZEB (外接長方形を求める)

GCENTB (重心座標を求める)

(5) 備考

従来から外接枠合わせ, 重心合わせの正規化法はよく用いられているが, 本方法は三重大学で開発された方法であり, 両者の中間の性質を示すと考えられる. 特に文字線分の飛び出し, つき出し等について有効と考えられる.

HISTSS

Histogram Sakusei

小領域内の4方向のヒストグラムを作成する.

作成 鶴岡信治	1985年 7月
形式 サブルーチン	言語: FORTRAN サイズ: 36行

(1) 概要

輪郭線の点列(x,y座標値)が与えられたとき, 4方向(双方向, 図1参照)の方向指数を求め, 指定された分割数により作られる小領域内での方向指数のヒストグラムを作成する.

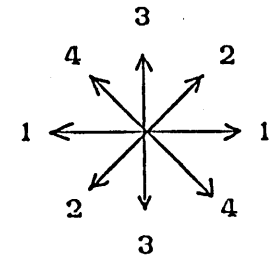


図1 方向指数

(2) 使用法

CALL HISTSS(IX,IY,IN,IS,ISI,IIIST,III,JII,II,JII)

引数	型と種類	属性	内容
IX	整数型 1次元配列	入力	輪郭点列のX座標 (0が輪郭の区切り)

IV	整数型 1次元配列	入力	輪郭点列のY座標 (0が輪郭の区切り)
IN	整数型 1次元配列	出力	各輪郭点の方向指数
IS	整数型	入力	IX, IV, INの整合寸法
ISI	整数型	入力	IX, IV, INの有効範囲
IIIST	整数型 3次元配列	出力	各小領域内の4方向の 方向指数ヒストグラム (III, JII, 4)
III	整数型	入力	X方向の小領域の数
JII	整数型	入力	Y方向の小領域の数
II	整数型	入力	画像のX方向の大きさ
J1	整数型	入力	画像のY方向の大きさ

(3) 計算法

まず輪郭点列 (IX, IV) の各点について4方向の方向指数を1次元配列INに求め、小領域の数 (III, JII) と画像の大きさ (II, J1) により、小領域の大きさを計算し ($II/III, J1/JII$)、各小領域内における方向指数のヒストグラムを作成する。IX, IVについてはSPIDERの輪郭追跡プログラムBDFL2の結果を用いるとよい。

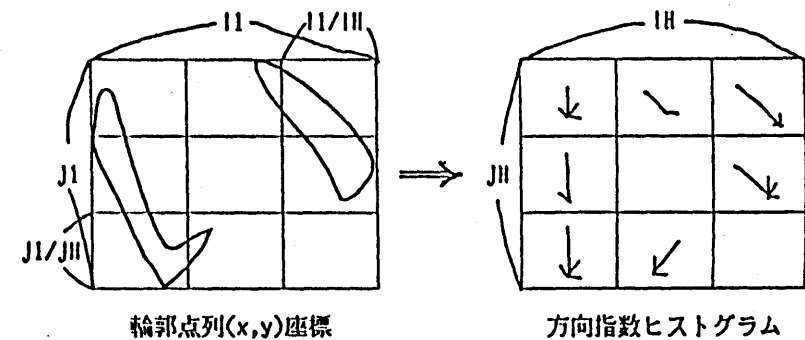


図2 方向指数ヒストグラムの作成

(4) プログラム例

```

PARAMETER(II=66, J1=65, JBL=1000, III=7, JII=7)
INTEGER IC(II, J1), JBLX(JBL), JBLY(JBL), INDEX4(JBL)
INTEGER IIIST(III, JII, 4)
:
:
CALL BDFL2(IC, IC, II, J1, JBLX, JBLY, JBL, JBL1, NBD, 8, KERR)
IF(KERR.EQ.-1)WRITE(6,*)'BDFL2 ERR(JBL OVER)'
CALL IIISTSS(JBLX, JBLY, INDEX4, JBL, JBL1, IIIST, III, JII, II, J1)
WRITE(6,6001)((IIIST(I, J, K); I=1, III), K=1, 4), J=1, JII)
6001 FORMAT(1H ,5X,7I4,2X,7I4,2X,7I4)
:
:

```


FEATUREH

Feature Extraction for 16-level Image 多値文字画像の特徴抽出

作成	鶴岡信治 1985年 10月
形式	コンプリート・プログラム 言語;FORTRAN サイズ;150行

(1) 概要

多値文字画像(ETL-8H)から特徴ベクトル(加重方向指数ヒストグラム)を得るまでの各処理毎の途中結果をラインプリンタに出力する。

(2) 使用法

ジョブ制御文のデータセットが用意されているのでSUBMITすればよい。
SUBMIT SHART(FEATUREH)

(3) 使用しているサブルーチン名

POUT16,FILTM8,BINARY,SMOOTH,NORMH3,MSWAKU,BDFL2(SPIDER37),HISTSS,WGT11
4,WEIGTI

参考文献

1)原田, 鶴岡, 木村, 三宅: "加重方向指数ヒストグラムと疑似ベイズ識別法を用いた手書き漢字・ひらがな認識", 信学技報, PRL83-68(1984).

FEATUREB

Feature Extraction for Binary Image 2値文字画像の特徴抽出

作成	鶴岡信治 1985年 10月
形式	コンプリート・プログラム 言語;FORTRAN サイズ;140行

(1) 概要

2値文字画像(ETL-8B)から特徴ベクトル(加重方向指数ヒストグラム)を得るまでの各処理毎の途中結果をラインプリンタに出力する。

(2) 使用法

ジョブ制御文のデータセットが用意されているのでSUBMITすればよい。
SUBMIT SHART(FEATUREB)
EXAMPLE

(3) 使用しているサブルーチン名

POUT2B,NORMB3,SMOOTH,MSWAKU,BDFL2(SPIDER37),HISTSS,WGTIN4,WEIGTI

参考文献

1)原田, 鶴岡, 木村, 三宅: "加重方向指数ヒストグラムと疑似ベイズ識別法を用いた手書き漢字・ひらがな認識", 信学技報, PRL83-68(1984).

CREFILEB

Feature Vector File Creation for Binary Image

2値文字画像の特徴ベクトルファイルの作成

作成	鶴岡信治 1985年 10月
形式	コンプリート・プログラム 言語;FORTRAN サイズ;160行

(1) 概要

2値文字画像(ETL-8B)から特徴ベクトル(加重方向指数ヒストグラム)を抽出し、ファイルをディスクに作成する。

(2) 使用法

ジョブ制御文のデータセットが用意されているのでSUBMITすればよい。

SUBMIT SMART(CREFILEB)

(3) 使用しているサブルーチン名

FILTM8,NORMB3,SMOOTH,MSWAKU,BDFL2(SPIDER37),HISTSS,WGTIN4,WEIGTI

参考文献

1)原田, 鶴岡, 木村, 三宅:" 加重方向指数ヒストグラムと疑似ベイズ識別法を用いた手書き漢字・ひらがな認識", 信学技報, PRL83-68(1984).

CREFILEH

Feature Vector File Creation for 16-level Image

多値文字画像の特徴ベクトルファイルの作成

作成	鶴岡信治 1985年 10月
形式	コンプリート・プログラム 言語;FORTRAN サイズ;170行

(1) 概要

多値文字画像(ETL-8H)から特徴ベクトル(加重方向指数ヒストグラム)を抽出し、ファイルをディスクに作成する。

(2) 使用法

ジョブ制御文のデータセットが用意されているのでSUBMITすればよい。

SUBMIT SMART(CREFILEH)

(3) 使用しているサブルーチン名

FILTM8,BINARY,SMOOTH,NORMM3,MSWAKU,BDFL2(SPIDER37),HISTSS,WGTIN4,WEIGTI

参考文献

1)原田, 鶴岡, 木村, 三宅:" 加重方向指数ヒストグラムと疑似ベイズ識別法を用いた手書き漢字・ひらがな認識", 信学技報, PRL83-68(1984).

PLTBIA

LINE DRAWING FOR BINARY·MULTI-LEVEL IMAGE USING XY-PLOTTER
XY-PLOTTERによる2値・多値画像の線画表現

作成	鶴岡信治	1986年	7月
形式	サブルーチン	言語; FORTRAN	サイズ; 69行

(1) 概要

文字画像を2値化して、文字領域をXY-PLOTTERを用いて、複数の横線（一行に1本）で描く。入力画像は2値画像でも多値画像でも良い。

(2) 使用法

CALL PLTBIA(IA,I1,J1,X0,Y0,DX,DY,IT,MNO,MCA)

引数	型と種類	属性	内容
IA	整数型 2次元配列	入力	表示したい画像
I1	整数型	入力	IPの整合寸法（第1添字の値）
J1	整数型	入力	IPの整合寸法（第2添字の値）
X0	実数型	入力	紙上の原点のx座標
Y0	実数型	入力	紙上の原点のy座標
DX	実数型	入力	紙上の文字枠のx方向の大きさ

DY	実数型	入力	紙上の文字枠のy方向の大きさ
IT	整数型	入力	2値化のしきい値
MNO	整数型	入力	文字の番号（6桁以内の数字）
MCA	整数型	入力	字種の番号（6桁以内の数字）

(3) 計算法

文字を左上から水平に走査していき、しきい値IT以上の点を見つけると、線を引き始め（ペンダウンし）、しきい値以下である点を見つけるとペンアップする。

高速に線画を書くために、奇数行目では、左から右へ線を引き、偶数行目では、右から左へ線を引く、工夫がしてある。

文字枠の左下に数字MNOをその右に数字MCAをかく。A4サイズ(21.0mmX25.0mm)では、X0を2.0、Y0を21.0とし、DXを4.0、DYを4.0とすると、3 X 4個の画像の表示ができる。

(4) 子ルーチン

FRAME,NUMBER,PLOT

(5) プログラム例

```

INTEGER IA(70,70)
C*** PLOTTER NO SYOKICHI NO SETTEI ***
X0=2.0
Y0=21.0
CALL XINT
CALL XVIEWP(0.0,0.0,21.0,25.0)
DX=4.0
DY=4.0
ITHRSII=1
C*** MOJI NITUI TE NO LOOP ***

```

```

MOJICA=2
DO 1000 MOJINN=1,20
  READ(8) IA
  CALL PLTBIA(IA,70,70,X0,Y0,DX,DY,ITHRSH,MOJINN,MOJICA)

```

```

C*** MOJI WAKU NO HIDARI SITA NO SEIJO ***

```

```

X0=X0+6.0

```

```

IF(X0.GT.16.0) THEN

```

```

  X0=2.0

```

```

  Y0=Y0-6.0

```

```

  IF(Y0.LT.0.0) THEN

```

```

C*** NEW PAGE ***

```

```

  CALL XVIEWP(0.0,0.0,21.0,25.0)

```

```

  X0=2.0

```

```

  Y0=21.0

```

```

  END IF

```

```

END IF

```

```

1000 CONTINUE

```

```

CALL XEND

```

```

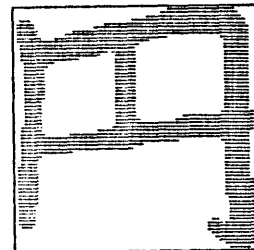
STOP

```

```

END

```



出力例

PLTH4A

LINE DRAWING OF 4-DIRECTION INDEX HISTOGRAM USING XY-PLOTTER

XY-PLOTTERによる4方向指数ヒストグラムの線画表現

作成	鶴岡信治	1986年	7月
形式	サブルーチン	言語; FORTRAN	サイズ; 90行

(1) 概要

4方向指数ヒストグラムを、XY-PLOTTERで各小領域の中心に放射状に描く。頻度はその方向の長さで表現されている。1画面に4文字分のヒストグラムをかく。

(2) 使用法

CALL PLTH4A(HIST4,IHH,JHH,MESG,MOJINN,MOJICA,MOJINN,MP)

引数	型と種類	属性	内容
HIST4	実数型 3次元配列	入力	表示したい4方向のヒストグラム (IHH,JHH,4)
IHH	整数型	入力	HIST4の整合寸法 (第1添字の値)
JHH	整数型	入力	HIST4の整合寸法 (第2添字の値)
MESG	文字型	入力	表示したい文字列 (8文字以内)
MOJINO	整数型	入力	文字の通し番号 (7桁以内)

MOJICA	整数型	入力	字種の番号(7桁以内)
MOJINN	整数型	入力	字種内の番号(7桁以内)
MP	整数型	入力	画面内の位置の番号(1-4)

(3) 計算法

ヒストグラムの長さをSIN,COSで極座標に変換し、各小領域の中心位置に放射状に線をひく。長さの正規化はしていない。

1画面に4つのヒストグラムをかき、ヒストグラムの位置はMPで制御できる。すなわち、MPが1で紙面の左上、2で右上、3で左下、4で右下に表示する。またこのサブルーチンを呼ぶ度にMPは1ずつ増加する。すなわち、最初に表示したい場所を指定して置けば、次からは自動的に決定される。

文字枠の上にMSGを、文字枠の下に左から順にMOJINO,MOJICA,MOJINNをかく。

(4) サブルーチン

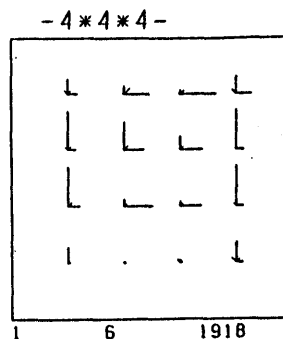
PLTH4B(PLOT),FRAME,SYMBOL,NUMBER,XVIEWP

(5) プログラム例

```

REAL HIST4(4,4,4)
CHARACTER MSG*8
DATA MSG/'-4*4*4- '/
CALL XINT
MP=1
C*** MOJI NITUI TE NO LOOP
DO 1000 MOJINO=1,10
  READ(8) NVOL,MOJICA,HIST4
  CALL PLT4A(NVOL,MOJICA,MOJINO,HIST4,4,4,MSG,MP)
1000 CONTINUE
  CALL XEND
STOP
END

```



PLTH4B

LINE DRAWING OF 4-DIRECTION INDEX HISTOGRAM USING XY-PLOTTER
XY-PLOTTERによる4方向指数ヒストグラムの線画表現の子ルーチン

作成	鶴岡信治	1986年	7月
形式	サブルーチン	言語; FORTRAN	サイズ; 25行

(1) 概要

小領域の4方向指数ヒストグラムを、XY-PLOTTERで描く。頻度はその方向の長さで表現されている。

(2) 使用法

CALL PLTH4B(HIST4,IHH,JHH,I,J,R,CX,CY)

引数	型と種類	属性	内容
HIST4	実数型 3次元配列	入力	表示したい4方向のヒストグラム (IHH,JHH,4)
IHH	整数型	入力	HIST4の整合寸法(第1添字の値)
JHH	整数型	入力	HIST4の整合寸法(第2添字の値)
I	整数型	入力	小領域の位置(1-IHH)
J	整数型	入力	小領域の位置(1-JHH)

R	実数型	入力	ヒストグラムの単位長さ当りの半径
CX	実数型	入力	ヒストグラムの中心のX座標
CY	実数型	入力	ヒストグラムの中心のY座標

(3) 計算法

ヒストグラムの長さをSIN,COSで極座標に変換し、各小領域の中心位置に放射状に4本の線をひく。長さの正規化はしていないので、親ルーチンでRを調整すること。

(4) 子ルーチン

PLOT

THIN4 I (IA, IB, IE, JE, ITYPE)

- (I) 機能：2 値（0，1）画像の細線化を行う。線の縮退，消失が少なくなる ように配慮。
- (II) 使用方法：
- IA（IE，JE）：入力画像，大きさ（IE，JE）の二次元配列
- IB（IE，JE）：出力画像， " "
- IE，JE ：上記整合配列の大きさが入る単純変数
- ITYPE ：連結性の種類を指定する単純変数（ ITYPE = 4 → 4 連結
 " 8 → 8 連結）
- 入出力関係 ：入力= IA，IE，JE，ITYPE，出力= IB
- (III) アルゴリズム：
- 入力画像の各点における連結数を求め，連結数が1 である点を順に削除していく。一方向走査逐次型。縮退を防ぐため特別の条件を導入。詳細は下記文献参照。
- (IV) 注意：
- (1) 入力画像（2 値）の値0 の部分は“地”の部分とみなされ，値1 の部分が細線化される。
本ルーチンの出力 IB では，細められた結果の図形上の点は1，その他の点は0 となっている。
- (2) 入力画像 IA と出力画像 IB は主プログラムでは同一配列でもよい。但し，このときは入力画像は保存されない。
- (3) 入力画像の縁の1，2，IE 行目，1，2，JE 列目は細線化の対象とはならず，出力には常に0 が入る。
- (4) 線分の縮退が少なくなるように，アルゴリズム設計上で若干の工夫がなされている。
- (5) 出力結果は下記論文で定義した完全な線幅1 の図形（線図形）となる。
- (6) キーワード：thinning, binary, sequential

文献

- (1) 鶴岡, 木村, 吉村, 三宅: "デジタル2値図形の一細線化法", 電子通信学会パターン認識と学習研究会資料 PRL78-47 (1978. 10)

付録 4 主なプログラムの説明とソースリスト

(名古屋大学大型計算機センター S L I P に収録)

THNG2

1. 目的・機能

鶴岡の逐次型アルゴリズムによる2値図形の細線化。

2. Calling Sequence

CALL THNG2 (IP, JP, ISX, ISY, NC)

3. 引数の説明

IP (ISX, ISY) : 入力2値図形。 (I)
 JP (ISX, ISY) : 出力2値図形。 (O)
 NC : 得られる心線の連結性 (4または8)。 (I)

4. 引数についての条件

- (1) IP, JPについての条件はTHNG1の4.(1)と同じ。
- (2) NCは出力として得られる心線の連結性を決定するパラメータで、NC=4のとき4-連結心線、NC=8のとき完全8-連結心線が得られる。

5. 使用上の注意

- (1)~(4)はTHNG1のこの項と同じ。
- (5) このアルゴリズムは、文献〔2〕に述べられているHilditchの逐次型アルゴリズム (THNG1) を変形して4-連結心線を得る課題を達成するよう工夫されたものである。8-連結心線は、一たん4-連結心線を得た後で、8-消去可能点を除去することによって得られる。従って、THNG1による8-連結心線とは異なったものが得られる。
- (6) この方法の欠点として、入力図形中に8-連結の意味でのみ連結している部分があると、一たん4-連結の基準で周辺点を除去するためこの部分が切断されてしまう。この方法は本来4-連結用のもので、8-連結の目的にはTHNG1を用いた方が安全である。ただし、上のような心配がない場合は、一たん4-連結心線を経る方が、得られる心線の形としては好ましいことが多い。

6. アルゴリズム

文献〔1〕の細線化アルゴリズムのSTEP2 (4-連結細線化) 及びSTEP3 (8-連結細線化のための後処理) をインプリメントをしたものである。雑音除去用の前処理であるSTEP1は、このプログラムには含まれていない。

参考文献

- 〔1〕 鶴岡他, "デジタル2値図形の細線化法," 信学技報, PRL78-47, Oct. 1978.
- 〔2〕 田村秀行, "細線化法についての諸考察," 同上, PRL75-66, Dec. 1975.

7. エラー処理

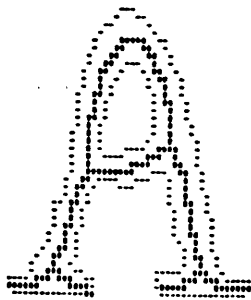
なし。

8. 特記事項

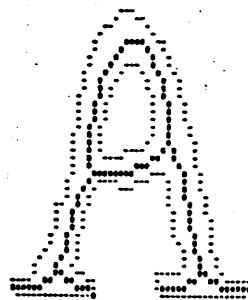
8-連結細線化の場合、4-連結の処理が後からという変則な手続きのため、途中のステップでリターンするようにはプログラムされていない。他のプログラムと同様KSTEPという引数を導入するのは容易である。

9. 実験例

(入力図形と表示方法はTHNG1と同じ)



(a) 4-連結細線化



(b) 8-連結細線化

DATA SET NAME : W49215A.SMART.FORT(SMOOTH)

```

10002 SUBROUTINE SMOOTH(KP,KPX,KPY,I1,I2,J1,J2,KPP,IT)
20002 CS CALL SMOOTH(KP,KPX,KPY,I1,I2,J1,J2,KPP,IT)
30001 CP*** 4-KINBOU DE 1-PIXEL GA 3 IJYO NARA HEIJIN NOODOCHI *****
40001 CP*** 9-KINBOU SUBETE GA 0 HARA 0 NI SURU *****
50001 CA I KP(KPX,KPY):HYURYOKU GAZOU I
60001 CA I KP(KPX,KPY):YOKO HOUKOU NO OOKISA I
70001 CA I KP(KPY):TATE HOUKOU NO OOKISA I
80001 CA I I1:SYORI HANI (SAGEN) I
90001 CA I I2:SYORI HANI (UGEN) I
100001 CA I J1:SYORI HANI (JYUGEN) I
110001 CA I J2:SYORI HANI (KAGEN) I
111001 CA I IT:SIKIICHI(=1:BINARY) I
120001 CA O KPP(KPX,KPY):SITSURYOKU GAZOU I
130001 CD 1905.06.22 CODED BY S.TSURUOKA
140001 CK SMOOTH,FILTER,KINBOU
150001 C*****
160001 INTEGER KP(KPX,KPY),KPP(KPX,KPY)
170001 C*****
180001 DO 50 J=J1,J2
190001 DO 50 I=I1,I2
200000 KPP(I,J)=KP(I,J)
210001 IF(KP(I,J).GE.IT) GOTO 1000
220001 C**** 4-RENKETSU NO ANA WO UMERU ****
230001 L=0
240001 HC=0
250001 IF(KP(I+1,J).LT.IT) GOTO 30,
260001 HC=HC+1
270001 L=L+KP(I+1,J)
280001 30 IF(KP(I,J-1).LT.IT) GOTO 31
290001 HC=HC+1
300000 L=L+KP(I,J-1)
310001 31 IF(KP(I-1,J).LT.IT) GOTO 32,
320001 HC=HC+1
330001 L=L+KP(I-1,J)
340001 32 IF(KP(I,J+1).LT.IT) GOTO 33
350001 HC=HC+1
360001 L=L+KP(I,J+1)
370001 33 CONTINUE
380001 IF(HC.LT.3) GOTO 50
390001 KPP(I,J)=L/HC
400000 GOTO 50
410001 1000 CONTINUE
420001 C*** 8-RENKETSU KORITSUTEN JYOKYO ***
430001 IF(KP(I+1,J).GE.IT) GOTO 50
440001 IF(KP(I+1,J-1).GE.IT) GOTO 50
450001 IF(KP(I,J-1).GE.IT) GOTO 50
460001 IF(KP(I-1,J-1).GE.IT) GOTO 50
470001 IF(KP(I-1,J).GE.IT) GOTO 50
480001 IF(KP(I-1,J+1).GE.IT) GOTO 50
490001 IF(KP(I,J+1).GE.IT) GOTO 50
500000 IF(KP(I+1,J+1).GE.IT) GOTO 50
510001 KPP(I,J)=0
520001 50 CONTINUE
530001 RETURN
540001 END

```

DATA SET NAME : W49215A.SMART.FORT(WGTIN4)

```

00010 SUBROUTINE WGTIN4(FILT,K1,K2,ILL)
00020 CS CALL WGTIN4(FILT,K1,K2,ILL)
00030 CP*** DAINI HYOUHONKA NO OMOMI KANSUU ***
00040 CA FILT(5,5):OMOMI KANSUU
00042 CA K1:OMOMI KANSUU NO OOKISA(=5)
00043 CA K2:OMOMI KANSUU NO OOKISA(=5)
00044 CA ILL:0=NORMAL END,1000=K1,K2 ERROR
00050 CK FILTER
00060 CD CODED BY S.TSURUOKA 85.06.24
00070 REAL FILT(5,5),D(5,5)
00080 C
00090 DATA D/.0,.0092,.0169,.0092,.0,.0092,.0573,.1054,.0573,.0092,
00100 0.0169,.1054,.1939,.1054,.0169,.0092,.0573,.1054,.0573,.0092,
00110 0.0,.0092,.0169,.0092,.0/
00120 C-----
00121 ILL=0
00122 IF(K1.NE.5)GOTO 10
00123 IF(K2.NE.5)GOTO 10
00130 DO 100 I=1,5
00140 DO 200 J=1,5
00150 FILT(I,J)=D(I,J)
00160 200 CONTINUE
00170 100 CONTINUE
00180 RETURN
00181 10 CONTINUE
00182 ILL=1000
00183 RETURN
00190 END

```

付録4 主なプログラムの説明とソースリスト

(名古屋大学大型計算機センターSMARTに収録)

DATA SET NAME : W49215A.SMART.FORT(HORMM3)

```

00010 SUBROUTINE NORMM3(IA,I1,J1,IB,I2,J2,IT,ILL)
00020 CS CALL NORMM3(IA,I1,J1,IB,I2,J2,IT,ILL)
00030 C
00040 CP** GYUSIN 4-BUNKATSU SEIKIKA (GYUSIN WO FRAME NO TYUSIN TO SURU) ***
00050 CP** 4-RYOUIKI DE SHISYUKU-HI WO KAETE ICHI OOKISA NO SEIKIKA WO SURU
00060 C
00070 CA I IA(I1,J1): NYUURYOKU TACHI GAZOU I
00080 CA I I1:IA NO DAIICHI SEIGOU SUNPOU I
00090 CA I J1:IA NO DAIINI SEIGOU SUNPOU I
00100 CA 0 IB(I2,J2): SYUTSURYOKU TACHI GAZOU I
00110 CA I I2:IB NO DAIICHI SEIGOU SUNPOU I
00120 CA I J2:IB NO DAIINI SEIGOU SUNPOU I
00130 CA I IT:SEIKIICHI I
00140 CA I ILL:ERROR CODE(0:NORMAL) I
00140 CD CODED BY S.TSURUOKA 1985.06.24
00150 CK ICHI-OOKISA NO SEIKIKA
00160 C *****
00170 C
00180 INTEGER IA(I1,J1),IB(I2,J2)
00190 C** JYUNBI **
00200 C*** (IXC,IYC):WAKU NO TYUSIN ***
00210 IXC=(I2+1)/2
00220 IYC=(J2+1)/2
00230 C*** MOJI NO GAISETSU WAKU WO MOTOMERU ***
00240 CALL HSIZEH(IA,I1,J1,IL,IR,JT,JB,IT,ILL)
00250 IF(ILL.NE.0) RETURN
00260 CT WRITE(6,*) 'IL,IR,JT,JB : ',IL,IR,JT,JB
00270 C*** MOJI NO GYUUSIN WO MOTOMERU ***
00280 CALL GCENTH(IA,I1,J1,IGX,IGY,SVX,SVY,IT,ILL)
00290 IF(ILL.NE.0) RETURN
00300 CT WRITE(6,*) 'IGX,IGY,SVX,SVY : ',IGX,IGY,SVX,SVY
00310 C***
00320 HI2=FLOAT(I2)/2.
00330 I22=HI2
00340 HJ2=FLOAT(J2)/2.
00350 J22=HJ2
00360 C
00370 C*** 1ST SUBIMAGE (RIGHT-UPPER) ***
00380 C
00390 A11=FLOAT(IR-IGX+1)/HI2
00400 A21=FLOAT(IGY-JT+1)/HJ2
00410 A12=IGX-A11*FLOAT(IXC)
00420 A22=IGY-A21*FLOAT(IYC)
00430 CT WRITE(6,*) '1ST SUBIMAGE ',A11,A21,A12,A22
00440 C*** IMAGE REPRODUCTION **
00450 DO 100 I=I22,I2
00460 ID=A11*FLOAT(I)+A12+0.5
00470 IF(ID.LT.1) ID=1
00480 IF(ID.GT.I1) ID=I1
00490 DO 100 J=1,J22
00500 JD=A21*FLOAT(J)+A22+0.5
00510 IF(JD.LT.1) JD=1
00520 IF(JD.GT.J1) JD=J1
00530 IB(I,J)=IA(ID,JD)
00540 100 CONTINUE
00550 C

```

```

00560 C*** 2ND SUBIMAGE (LEFT-UPPER) ***
00570 C
00580 A11=FLOAT(IGX-IL+1)/HI2
00590 A21=FLOAT(IGY-JT+1)/HJ2
00600 A12=IGX-A11*FLOAT(IXC)
00610 A22=IGY-A21*FLOAT(IYC)
00620 CT WRITE(6,*) '2ND SUBIMAGE ',A11,A21,A12,A22
00630 DO 200 I=1,I22
00640 ID=A11*FLOAT(I)+A12+0.5
00650 IF(ID.LT.1) ID=1
00660 IF(ID.GT.I1) ID=I1
00670 DO 200 J=1,J22
00680 JD=A21*FLOAT(J)+A22+0.5
00690 IF(JD.LT.1) JD=1
00700 IF(JD.GT.J1) JD=J1
00710 IB(I,J)=IA(ID,JD)
00720 200 CONTINUE
00730 C
00740 C*** 3RD SUBIMAGE (LEFT-LOWER) ***
00750 C
00760 A11=FLOAT(IGX-IL+1)/HI2
00770 CT WRITE(6,*) 'JB,IGY,HJ2',JB,IGY,HJ2
00780 A21=FLOAT(JB-IGY+1)/HJ2
00790 A12=IGX-A11*FLOAT(IXC)
00800 A22=IGY-A21*FLOAT(IYC)
00810 CT WRITE(6,*) '3RD SUBIMAGE ',A11,A21,A12,A22
00820 DO 300 I=1,I22
00830 ID=A11*FLOAT(I)+A12+0.5
00840 IF(ID.LT.1) ID=1
00850 IF(ID.GT.I1) ID=I1
00860 DO 300 J=J22,J2
00870 JD=A21*FLOAT(J)+A22+0.5
00880 IF(JD.LT.1) JD=1
00890 IF(JD.GT.J1) JD=J1
00900 IB(I,J)=IA(ID,JD)
00910 300 CONTINUE
00920 C
00930 C*** 4TH SUBIMAGE (RIGHT-LOWER) ***
00940 C
00950 A11=FLOAT(IR-IGX+1)/HI2
00960 A21=FLOAT(JB-IGY+1)/HJ2
00970 A12=IGX-A11*FLOAT(IXC)
00980 A22=IGY-A21*FLOAT(IYC)
00990 CT WRITE(6,*) '4TH SUBIMAGE ',A11,A21,A12,A22
01000 DO 400 I=I22,I2
01010 ID=A11*FLOAT(I)+A12+0.5
01020 IF(ID.LT.1) ID=1
01030 IF(ID.GT.I1) ID=I1
01040 DO 400 J=J22,J2
01050 JD=A21*FLOAT(J)+A22+0.5
01060 IF(JD.LT.1) JD=1
01070 IF(JD.GT.J1) JD=J1
01080 IB(I,J)=IA(ID,JD)
01090 400 CONTINUE
01100 C*****
01110 RETURN
END

```

DATA SET NAME : W49215A.SMART.FORT(NORMB3)

```

10000 SUBROUTINE NORMB3(IA,I1,J1,IB,I2,J2,ILL)
20000 CS CALL NORMB3(IA,I1,J1,IB,I2,J2,ILL)
30000 C
40000 CP** GYUSIN 4-BUNKATSU SEIKIKA (GYUSIN NO FRAME NO TYUSIN TO SURU) ***
50000 CP** 4-RYOUIKI DE SINSYUKU-HI WO KAETE ICHI OOKISA NO SEIKIKA WO SURU
60000 C
70000 CA I IA(I1,J1): INPUT FIGURE I
80000 CA I I1:IA NO DAIICHI SEIGOU SUNPOU I
90000 CA I J1:IA NO DAINI SEIGOU SUNPOU I
100000 CA O IB(I2,J2): OUTPUT FIGURE I
110000 CA I I2:IB NO DAIICHI SEIGOU SUNPOU I
120000 CA I J2:IB NO DAINI SEIGOU SUNPOU I
130000 CA O ILL:ERROR CODE (0:NORMAL END) I
140000 CD CODED BY S.TSURUOKA 1985.06.24
150000 CK ICHI-OKISA NO SEIKIKA
160000 C *****
170000 C
180000 INTEGER IA(I1,J1),IB(I2,J2)
190000 C** JYUUBI **
200000 C*** (IXC,IYC):WAKU NO TYUSIN ***
210000 IXC=(I2+1)/2
220000 IYC=(J2+1)/2
230000 C*** MOJI NO GAISETSU WAKU WO MOTOMERU ***
240000 CALL NSIZEB(IA,I1,J1,IL,IR,JT,JB,ILL)
250000 IF(ILL.NE.0) RETURN
260000 CT WRITE(6,*) ' IL,IR,JT,JB : ',IL,IR,JT,JB
270000 C*** MOJI NO GYUUSIN WO MOTOMERU ***
280000 CALL GCENTB(IA,I1,J1,IGX,IGY,SVX,SVY,ILL)
290000 IF(ILL.NE.0) RETURN
300000 CT WRITE(6,*) ' IGX,IGY,SVX,SVY : ',IGX,IGY,SVX,SVY
310000 C***
320000 HI2=FLOAT(I2)/2.
330000 I22=HI2
340000 HJ2=FLOAT(J2)/2.
350000 J22=HJ2
360000 C
370000 C*** 1ST SUBIMAGE (RIGHT-UPPER) ***
380000 C
390000 A11=FLOAT(IR-IGX+1)/HI2
400000 A21=FLOAT(IGY-JT+1)/HJ2
410000 A12=IGX-A11*FLOAT(IXC)
420000 A22=IGY-A21*FLOAT(IYC)
430000 CT WRITE(6,*) '1ST SUBIMAGE ',A11,A21,A12,A22
440000 C*** IMAGE REPRODUCTION **
450000 DO 100 I=I22,I2
460001 ID=A11*FLOAT(I)+A12+0.5
470000 IF(ID.LT.1) ID=1
480000 IF(ID.GT.I1) ID=I1
490000 DO 100 J=J22,J2
500001 JD=A21*FLOAT(J)+A22+0.5
510000 IF(JD.LT.1) JD=1
520000 IF(JD.GT.J1) JD=J1
530000 IB(I,J)=IA(ID,JD)
540000 100 CONTINUE
550000 C
560000 C*** 2ND SUBIMAGE (LEFT-UPPER) ***

```

DATA SET NAME : W49215A.SMART.FORT(NORMB3)

```

570000 C
580000 A11=FLOAT(IGX-IL+1)/HI2
590000 A21=FLOAT(IGY-JT+1)/HJ2
600000 A12=IGX-A11*FLOAT(IXC)
610000 A22=IGY-A21*FLOAT(IYC)
620000 CT WRITE(6,*) '2ND SUBIMAGE ',A11,A21,A12,A22
630000 DO 200 I=1,I22
640001 ID=A11*FLOAT(I)+A12+0.5
650000 IF(ID.LT.1) ID=1
660000 IF(ID.GT.I1) ID=I1
670000 DO 200 J=1,J22
680001 JD=A21*FLOAT(J)+A22+0.5
690000 IF(JD.LT.1) JD=1
700000 IF(JD.GT.J1) JD=J1
710000 IB(I,J)=IA(ID,JD)
720000 200 CONTINUE
730000 C
740000 C*** 3RD SUBIMAGE (LEFT-LOWER) ***
750000 C
760000 A11=FLOAT(IGX-IL+1)/HI2
770000 CT WRITE(6,*) 'JB,IGY,HJ2',JB,IGY,HJ2
780000 A21=FLOAT(JB-IGY+1)/HJ2
790000 A12=IGX-A11*FLOAT(IXC)
800000 A22=IGY-A21*FLOAT(IYC)
810000 CT WRITE(6,*) '3RD SUBIMAGE ',A11,A21,A12,A22
820000 DO 300 I=1,I22
830001 ID=A11*FLOAT(I)+A12+0.5
840000 IF(ID.LT.1) ID=1
850000 IF(ID.GT.I1) ID=I1
860000 DO 300 J=J22,J2
870001 JD=A21*FLOAT(J)+A22+0.5
880000 IF(JD.LT.1) JD=1
890000 IF(JD.GT.J1) JD=J1
900000 IB(I,J)=IA(ID,JD)
910000 300 CONTINUE
920000 C
930000 C*** 4TH SUBIMAGE (RIGHT-LOWER) ***
940000 C
950000 A11=FLOAT(IR-IGX+1)/HI2
960000 A21=FLOAT(JB-IGY+1)/HJ2
970000 A12=IGX-A11*FLOAT(IXC)
980000 A22=IGY-A21*FLOAT(IYC)
990000 CT WRITE(6,*) '4TH SUBIMAGE ',A11,A21,A12,A22
1000000 DO 400 I=I22,I2
1010001 ID=A11*FLOAT(I)+A12+0.5
1020000 IF(ID.LT.1) ID=1
1030000 IF(ID.GT.I1) ID=I1
1040000 DO 400 J=J22,J2
1050001 JD=A21*FLOAT(J)+A22+0.5
1060000 IF(JD.LT.1) JD=1
1070000 IF(JD.GT.J1) JD=J1
1080000 IB(I,J)=IA(ID,JD)
1090000 400 CONTINUE
1100000 C*****
1110000 RETURN
1120000 END

```

DATA SET NAME : W49215A.SMART.FORT(PSIZEB)

```

00010 SUBROUTINE PSIZEB(IA,I1,J1,IS,IE,JS,JE,IL,IR,JT,JB,KERR)
00020 CS CALL PSIZEB(IA,I1,J1,IS,IE,JS,JE,IL,IR,JT,JB,KERR)
00030 CP*** IA(I1,J1) YORI TOKUTEI HANI(IS-IE,JS-JE)NAI DE SATAN(IL),
00040 CP*** UTAN(IR),JYOTAN(JT),KATAN(JB) WO MOTOMERU
00050 C
00060 CA I IA(I1,J1):INPUT IMAGE I
00070 CA I I1:IP NO DAIICHI SEIGOU SUNPOU I
00080 CA I J1:IP NO DAINI SEIGOU SUNPOU I
00090 CA I IS:SYORI HANI NO SATAN I
00100 CA I IE:SYORI HANI NO UTAN I
00110 CA I JS:SYORI HANI NO JYOTAN I
00120 CA I JE:SYORI HANI NO KATAN I
00130 CA 0 IL:MOJI RYOUIKI NO SATAN I
00140 CA 0 IR:MOJI RYOUIKI NO UTAN I
00150 CA 0 JT:MOJI RYOUIKI NO TOP I
00160 CA 0 JB:MOJI RYOUIKI NO BOTTOM I
00161 CA 0 KERR:CONDITION CODE I
00162 CE KERR=0:NORMAL END
00163 CE KERR=10000:ALL-0
00164 CE KERR=20000:KATAN GA MOTOMARANAI
00165 CE KERR=40000:SATAN GA MOTOMARANAI
00166 CE KERR=80000:UTAN GA MOTOMARANAI
00167 C
00170 CD*** CODED BY S.TSURUOKA 1985.06.24
00180 CK*** OOKISA NO SOKUTEI
00190 C*****
00200 C
00210 INTEGER IA(I1,J1)
00220 KERR=0
00230 DO 10 J=JS,JE
00240 DO 10 I=IS,IE
00250 IF (IA(I,J).GT.0) GOTO 1000
00260 10 CONTINUE
00270 C*****
00280 KERR=100
00290 RETURN
00291 C*****
00300 1000 CONTINUE
00310 JT=J
00320 JS=J1-JE+1
00330 JE=J1-JT+1
00340 DO 20 JJ=JS,JE
00350 J=J1-JJ+1
00360 DO 20 I=IS,IE
00370 IF (IA(I,J).GT.0) GOTO 1010
00380 20 CONTINUE
00390 KERR=200
00400 RETURN
00401 C*****
00410 1010 CONTINUE
00420 JB=J
00430 DO 30 I=IS,IE
00440 DO 30 J=JT,JB
00450 IF (IA(I,J).GT.0) GOTO 1020
00460 30 CONTINUE
00470 KERR=400

```

DATA SET NAME : W49215A.SMART.FORT(PSIZEB)

```

00480 RETURN
00481 C*****
00490 1020 CONTINUE
00500 IL=I
00510 IS=I1-IE+1
00520 IC=I1-IL+1
00530 DO 40 II=IS,IE
00540 I=I1-II+1
00550 DO 40 J=JT,JB
00560 IF (IA(I,J).GT.0) GOTO 1030
00570 40 CONTINUE
00580 KERR=800
00590 RETURN
00591 C*****
00600 1030 CONTINUE
00610 IR=I
00620 990 CONTINUE
00621 C*****
00630 RETURN
00640 END

```

DATA SET NAME : W49215A.SMART.FORT(P SizEM)

```

00010 SUBROUTINE PSIZEM(IA,I1,J1,IS,IE,JS,JE,IL,IR,JT,JB,IT,KERR)
00020 CS CALL PSIZEM(IA,I1,J1,IS,IE,JS,JE,IL,IR,JT,JB,IT,KERR)
00030 CP*** IA(I1,J1) YORI TOKUTEI HANI(IS-IE,JS-JE)HAI DE SATAN(IL),
00040 CP*** UTAN(IR),JYOTAN(JT),KATAN(JB) WO MOTOMERU
00050 C
00060 CA I IA(I1,J1):INPUT IMAGE I
00070 CA I I1:IP NO DAIICHI SEIGOU SUNPOU I
00080 CA I J1:IP NO DAINI SEIGOU SUNPOU I
00090 CA I IS:SYORI HANI NO SATAN I
00100 CA I IE:SYORI HANI NO UTAN I
00110 CA I JS:SYORI HANI NO JYOTAN I
00120 CA I JE:SYORI HANI NO KATAN I
00130 CA O IL:HOJI RYOUIKI NO SATAN I
00140 CA O IR:HOJI RYOUIKI NO UTAN I
00150 CA O JT:HOJI RYOUIKI NO TOP I
00160 CA O JB:HOJI RYOUIKI NO BOTTOM I
00161 CA I IT:SIXIICHI I
00170 CA O KERR:CONDITION CODE I
00180 CE KERR=0:NORMAL END
00190 CE KERR=10000:ALL-0
00200 CE KERR=20000:KATAN GA MOTOMARANAI
00210 CE KERR=40000:SATAN GA MOTOMARANAI
00220 CE KERR=30000:UTAN GA MOTOMARANAI
00230 C
00240 CD*** CODED BY S.TSURUOKA 1985.06.24
00250 CK*** OOKISA NO SOKUTEI
00260 C*****
00270 C
00280 INTEGER IA(I1,J1)
00290 KERR=0
00300 DO 10 J=JS,JE
00310 DO 10 I=IS,IE
00320 IF (IA(I,J).GE.IT) GOTO 1000
00330 10 CONTINUE
00340 C*****
00350 KERR=100
00360 RETURN
00370 C*****
00380 1000 CONTINUE
00390 JT=J
00400 JS=J1-JE+1
00410 JE=J1-JT+1
00420 DO 20 JJ=JS,JE
00430 J=J1-JJ+1
00440 DO 20 I=IS,IE
00450 IF (IA(I,J).GE.IT) GOTO 1010
00460 20 CONTINUE
00470 KERR=200
00480 RETURN
00490 C*****
00500 1010 CONTINUE
00510 JB=J
00520 DO 30 I=IS,IE
00530 DO 30 J=JT,JR
00540 IF (IA(I,J).GE.IT) GOTO 1020
00550 30 CONTINUE

```

DATA SET NAME : W49215A.SMART.FORT(P SizEM)

```

00560 KERR=400
00570 RETURN
00580 C*****
00590 1020 CONTINUE
00600 IL=I
00610 IS=I1-IE+1
00620 IE=I1-IL+1
00630 DO 40 II=IS,IE
00640 I=I1-II+1
00650 DO 40 J=JT,JB
00660 IF (IA(I,J).GE.IT) GOTO 1030
00670 40 CONTINUE
00680 KERR=800
00690 RETURN
00700 C*****
00710 1030 CONTINUE
00720 IR=I
00730 990 CONTINUE
00740 C*****
00750 RETURN
00760 END

```

DATA SET NAME : W49215A.SMART.FORT(HISTSS)

```

10000  SUBROUTINE HISTSS(JBLX,JBLY,INDEX4,JBL,JBL1,JHIST,IH,JH,I1,J1)
20000  CS  CALL HISTSS(JBLX,JBLY,INDEX4,JBL,JBL1,JHIST,IH,JH,I1,J1)
30000  CP*** XY-RINKAKUTEN RETSU YORI SUBREGION NI OKERU 4-HOUKOU SISUU
40000  CP*** HISTGRAM O MOTOMERU          HOUKOU SISUU
50000  CD*** CODED BY S.TSURUOKA          -----> J
60000  CN*** CALL BDFL2 NO KEKKA(RINKAKU      I 4 3 2
70000  CN*** TENRETSU) O INPUT(JBLX,JBLY)      J 1 1 X 1
80000  C***          I 2 3 4
90000  CA I  JBLX(JBL):RINKAKUTENRETSU NO X-ZAHYO
100000 CA I  JBLY(JBL):RINKAKUTENRETSU NO Y-ZAHYO
110000 CA 0  INDEX4(JBL):RINKAKUTENRETSU NO 4-HOUKOUSISUU
120000 CA I  JBL:HAIRETSU JBLX,JBLY,INDEX4 NO SEIGOUSUNPOU
130000 CA I  JBL1:JBLX,JBLY,INDEX4 NO YUUKOU HANI
140000 CA 0  JHIST(IH,JH,4):BUNKATSU RYOUIKI DENO 4-HOUKOUSISUU HISTGRAM
150000 CA I  IH:YOKO HOUKOU NO RYOUIKI SUU
160000 CA I  JH:TATE HOUKOU NO RYOUIKI SUU
170000 CA I  (I1,J1):JBLX,JBLY NO MOTOMETA TOKI NO GAZOO NO OOKISA
180000 CK  HOUKOUSISUU,HISTGRAM,RINKAKUTENRETSU,BUNKATSU-RYOUIKI
190000 C****
200000      INTEGER JBLX(JBL),JBLY(JBL),INDEX4(JBL)
210000      INTEGER JHIST(IH,JH,4)
220000 C***
230000      INDEX4(I)=0
240000      LS=2
250000 *** XY-RINKAKUTEN RETSU YORI 4-HOUKOU SISUU WO INDEX4 NI MOTOMERU ***
260000 10 CONTINUE
270000      DO 100 I=LS,JBL1
280000      IF(JBLX(I).EQ.0) GOTO 101
290000      KX=JBLX(I)-JBLX(I-1)
300000      KY=JBLY(I)-JBLY(I-1)
310000      IF(KX.EQ.1.AND.KY.EQ.0) THEN
320000          INDEX4(I)=1
330000          GOTO 100
340000      END IF
350000      IF(KX.EQ.1.AND.KY.EQ.-1) THEN
360000          INDEX4(I)=2
370000          GOTO 100
380000      END IF
390000      IF(KX.EQ.0.AND.KY.EQ.-1) THEN
400000          INDEX4(I)=3
410000          GOTO 100
420000      END IF
430000      IF(KX.EQ.-1.AND.KY.EQ.-1) THEN
440000          INDEX4(I)=4
450000          GOTO 100
460000      END IF
470000      IF(KX.EQ.-1.AND.KY.EQ.0) THEN
480000          INDEX4(I)=1
490000          GOTO 100
500000      END IF
510000      IF(KX.EQ.-1.AND.KY.EQ.1) THEN
520000          INDEX4(I)=2
530000          GOTO 100
540000      END IF
550000      IF(KX.EQ.0.AND.KY.EQ.1) THEN
560000          INDEX4(I)=3

```

DATA SET NAME : W49215A.SMART.FORT(HISTSS)

```

570000      GOTO 100
580000      END IF
590000      IF(KX.EQ.1.AND.KY.EQ.1) THEN
600000          INDEX4(I)=4
610000          GOTO 100
620000      END IF
630000      INDEX4(I)=0
640000 100 CONTINUE
650000      GOTO 200
660000 ***** RINKAKU TSUISEKI KAISITEN NO SYORI *****
670000 101 LS=I+2
680000      INDEX4(I)=0
690000      INDEX4(I+1)=0
700000      GOTO 10
710000 ***** 4-HOUKOU NO HISTGRAM O MOTOMERU *****
720000 200 CONTINUE
730000 ***** SIZE OF SUBREGION *****
740000      ISR=I1/IH
750000      JSR=J1/JH
760000 ***** JHIST NO SYOKIKA *****
770000      DO 300 KH=1,4
780000      DO 300 JJ=1,JH
790000      DO 300 II=1,IH
800000      JHIST(II,JJ,KH)=0
810000 300 CONTINUE
820000 ***** BUNKATSU RYOUIKI DENO HISTGRAM NO SAKUSEISURU *****
830000      DO 400 I=1,JBL1
840000      KH=INDEX4(I)
850000      IF(KH.EQ.0) GOTO 400
860000      IX=JBLX(I)
870000      IY=JBLY(I)
880000      NSRX=(IX-1)/ISR+1
890000      NSRY=(IY-1)/JSR+1
891001      IF(NSRX.GT.IH) NSRX=IH
892001      IF(NSRY.GT.JH) NSRY=JH
900000      JHIST(NSRX,NSRY,KH)=JHIST(NSRX,NSRY,KH)+1
910000 400 CONTINUE
920000      RETURN
930000      END

```

DATA SET NAME : W49215A.EXAMPLE.FORT(FEATUREH)

DATA SET NAME : W49215A.EXAMPLE.FORT(FEATUREH)

```

00010 *****00570 C*****
00020 *****00580 C***
00030 ***** MAIN ROUTINE (FEATURE EXTRACTION TEST)00590 CALL FILTMB(IA,I1,J1,IB)
00040 ***** KAJYU HOUKOU SISU HISTGRAM HOU (ETL-8)00600 WRITE(6,*) ' *** FILTMB OUTPUT ***'
00050 ***** CODED BY S.TSURUOKA (ENG.ELECTRONICS, MIE UNIV.)00610 CALL POUT16(IB,I1,J1,ILL)
00060 ***** 1985.6.1700620 C*** 2-CHIKA ***
00070 *****00630 JTND=1
00080 *****00640 CALL BINARY(IB,I1,J1,IA,JTHD)
00090 ***** PARAMETER(I1=128,J1=128,I2=70,J2=70)00650 WRITE(6,*) ' *** BINARY OUTPUT ***'
00100 ***** PARAMETER(JBL=2000,IH=7,JH=7)00660 CALL POUT2B(IA,I1,J1,ILL)
00110 ***** PARAMETER(INH=4,JHH=4,K1=5,K2=5,IP=1,JP=1,IS=2,JS=2)00670 C*** SMOOTHING ***
00120 *****00680 I11=I1-1
00130 C MCAMAX : MOJI CATAGORY SUU (1-956)00690 J11=J1-1
00140 C MNOMAX : MOJI SUU (1 CATEGORY NAI : 1-160 )00700 CALL SMOOTH(IA,I1,J1,2,I11,2,J11,IB,1)
00150 *****00710 WRITE(6,*) ' *** SMOOTH OUTPUT ***'
00160 ***** PARAMETER(MCAMAX=2,MNOMAX=1)00720 CALL POUT2B(IB,I1,J1,ILL)
00170 *****00730 C*** ICHI-OKKISA NO SEIKIKA ***
00180 ***** INTEGER IA(I1,J1),IB(I1,J1)00740 CALL NORMM3(IB,I1,J1,IAA,I2,J2,1,ILL)
00190 ***** INTEGER IAA(I2,J2),IBB(I2,J2),IC(I2+1,J2+2)00750 WRITE(6,*) ' *** NORMM3 OUTPUT ***'
00200 ***** INTEGER JBLX(JBL),JBLY(JBL),INDEX4(JBL)00760 CALL POUT2B(IAA,I2,J2,ILL)
00210 ***** INTEGER JHIST(IH,JH,4),IDI(13),JHISTK(IH,JH)00770 C***
00220 ***** REAL HIST(INH,JHH,4),HW(5,5),HISTK(INH,JHH)00780 C*****
00230 ***** CHARACTER IDC(1)*800790 C*** HEKATSUKA ***
00240 *****00800 CALL FILTMB(IAA,I2,J2,IBB)
00250 C***00810 WRITE(6,*) ' *** FILTMB OUTPUT ***'
00260 ***** DO 1000 MOJICA=1,MCAMAX00820 CALL POUT2B(IBB,I2,J2,ILL)
00270 ***** WRITE(6,*) 'MOJI CATEGORY =' ,MOJICA00830 C*** IA NI 1 BIT NO WAKU O TSUKERU ***
00280 ***** DO 2000 MOJINN=1,MNOMAX00840 CALL MSHAKU(IBB,I2,J2,IC,1)
00290 C***00850 WRITE(6,*) ' *** MSHAKU OUTPUT ***'
00300 *****00860 I3=I2+2
00310 C*** NVOL:VOLUME NO.(1-32) , MOJINP:VOL. NAI NO NUMBER(1-5)00870 J3=J2+2
00320 *****00880 CALL POUT16(IC,I3,J3,ILL)
00330 C***00890 C***
00340 ***** NVOL=(MOJINN-1)/5+100900 *****
00350 ***** MOJINP=MOJINN-(NVOL-1)*500910 *** BORDER FOLLOWING ( 8-RENKETSU ) USING SPIDER(BDFL2) **
00360 ***** MOJINO=MOJICA+(MOJINP-1)*95600920 *** INPUT(1,3,4,10), OUTPUT(2,4,5,6,7,8,11) **
00370 *****00930 *** 1ST:INPUT 2D BINARY IMAGE(I,J)(0,1)(FRAME 1BIT 0) **
00380 ***00940 *** 2ND:OUTPUT 2D IMAGE (0,1,2(TSUISEKI TEN)) **
00390 *** READ MOJI DATA IA(I1,J1) FROM DISK (ETL-8H)00950 *** 3RD:IMAGE SIZE(X) ,4TH:IMAGE SIZE(Y) **
00400 *** INPUT(1,2) ,OUTPUT(3,4,5,6)00960 *** 5TH:BORDER LIST(X)(RINKAKU,NO SAKAIME NI 0),6TH:BORDER LIST(Y) **
00410 *** NVOL:VOLUME NUMBER(1-32) ,MOJINO:MOJI NUMBER(1-7777)00970 *** 7TH:ARRAY SIZE OF 5,6 ,8TH:EFFISIENT SIZE OF 5,6 **
00420 *** IDI:JIS CODE WADO NO ZOKUSEI ,IDC:KANJI NO YOMI00980 *** 9TH:NUMBER OF CONECTED COMPONENT **
00430 *** IA:MOJI PATTERN(128,127) ,ILL:ERROR CODE00990 *** 10TH:CONECTED NUMBER(4 OR 8) ,11TH:ERROR CODE **
00440 ***01000 ***
00450 *****01010 *****
00460 C***01020 C
00470 ***** CALL RETL8H(NVOL,MOJINO,IDI,IDC,IA,ILL)01030 CALL BDFL2(IC,IC,I3,J3,JBLX,JBLY,JBL,JBL1,NBD,8,KERR)
00480 ***** WRITE(6,6009) NVOL,MOJINO,IDI(2),IDC01040 IF(KERR.EQ.-1) THEN
00490 6009 FORMAT(1H ,5X,'NVOL=',I4,5X,'MOJINO=',I4,5X,'JIS CODE=',Z4,5X,01050 WRITE(6,*) 'BDFL2 ERROR(JBL OVER) MOJINO=',MOJINO
00500 * 'YOMI:',Z8)01060 GOTO 1000
00510 ***** IF(ILL.NE.0) THEN01070 ELSE
00520 ***** WRITE(6,6900)01080 WRITE(6,6011) 'JBLX:',(JBLX(I),I=1,JBL1)
00530 6900 FORMAT(1H ,*** READ ROUTINE(RET8H) ERROR ***)01090 6011 FORMAT(1H ,A,40I3)
00540 ***** STOP01100 WRITE(6,6011) 'JBLY:',(JBLY(I),I=1,JBL1)
00550 ***** END IF01110 END IF
00560 ***** CALL POUT16(IA,I1,J1,ILL)01120 C***

```


DATA SET NAME I W49215A.EXAMPLE.FORT(FEATUREH)

DATA SET NAME I W49215A.EXAMPLE.FORT(FEATUREB)

```

01130 *****
01140 *** 4-HOUKOU SISU HISTGRAM O MOTOMERU
01150 *** INPUT(1,2,4,5,7,8,9,10) , OUTPUT(3,6)
01160 *** 1ST:BORDER LIST(X) , 2ND:BORDER LIST(Y)
01170 *** 3RD:4-DIRECTION INDEX, 4TH:ARRAY SIZE OF 1,2,3
01180 *** 5TH:EFFICIENT SIZE OF 1,2,3 , 6TH:HISTGRAMS OF SUB REGION
01190 *** 7TH:SIZE(X) OF 6 , 8TH:SIZE(Y) OF 6
01200 *** 9TH:SIZE(X) OF IMAGE , 10TH:SIZE(Y) OF IMAGE
01210 *****
01220 C
01230 CALL HISTSS(JBLX,JBLY,INDEX4,JBL,JBL1,JHIST,IH,JH,I3,J3)
01240 WRITE(6,6000) MOJINO
01250 6000 FORMAT(1H ,5X,'KAGYU HOUKOU HISTGRAM MOJINO=',I4)
01260 WRITE(6,6001) (((JHIST(I,J,K),I=1,IH),K=1,4),J=1,JH)
01270 6001 FORMAT(1H ,5X,7I4,2X,7I4,2X,7I4,2X,7I4)
01280 C*** SECOND SAMPLING (7,7)--->(4,4) ***
01290 CALL WGTIN4(WW,S,5,ILL)
01300 DO 200 KH=1,4
01310 DO 210 I=1,IH
01320 DO 210 J=1,JH
01330 JHISTK(I,J)=JHIST(I,J,KH)
01340 210 CONTINUE
01350 CALL WEIGT1(JHISTK,IH,JH,HISTK,IHH,JHH,WW,S,5,
01360 * IP,JP,IS,JS,ILL)
01370 DO 220 I=1,IHH
01380 DO 220 J=1,JHH
01390 HIST(I,J,KH)=HISTK(I,J)
01400 220 CONTINUE
01410 200 CONTINUE
01420 WRITE(6,6002) (((HIST(I,J,K),I=1,IHH),K=1,4),J=1,JHH)
01430 6002 FORMAT(1H ,10X,4F5.1,5X,4F5.1,5X,4F5.1,5X,4F5.1)
01440 *****
01450 ***** LOOP END ***
01460 2000 CONTINUE
01470 1000 CONTINUE
01480 STOP
01490 END

10001 *****
20001 *****
30001 ***** MAIN ROUTINE (FEATURE EXTRACTION TEST)
40001 ***** KAGYU HOUKOU SISU HISTGRAM HOU (ETL-8B)
50001 ***** CODED BY S.TSURUOKA (ENG.ELECTRONICS, MIE UNIV.)
60001 ***** 1985.6.17
70001 *****
80001 *****
90001 PARAMETER(I1=64,J1=63,JBL=1000,IH=7,JH=7,NEND=2)
100001 PARAMETER(IHH=4,JHH=4,K1=5,K2=5,IP=1,JP=1,IS=2,JS=2)
110001 C
120000 INTEGER IA(I1,J1),IB(I1,J1),IC(I1+2,J1+2)
130001 INTEGER JBLX(JBL),JBLY(JBL),INDEX4(JBL)
140001 INTEGER JHIST(IH,JH,4),IDI(13),JHISTK(IH,JH)
150001 REAL HIST(IHH,JHH,4),WW(S,5),HISTK(IHH,JHH)
160001 CHARACTER IDC*4
170001 C
180001 *****
190001 ***
200000 *** NEND...MOJI NO.,NVOL=VOLUME NO.
210001 ***
220001 NVOL=1
230001 DO 100 N=1,NEND
240001 C
250001 *****
260001 ***
270001 *** READ MOJI DATA IA(I1,J1) FROM DISK (ETL-8B)
280001 *** INPUT(1,2) ,OUTPUT(3,4,5,6)
290001 *** NVOL:VOLUME NUMBER(1-32) ,MOJINO:MOJI NUMBER(1-????)
300000 *** IDI:JIS CODE NADO NO ZOKUSEI ,IDC:KANJI NO YOMI
310001 *** IA:MOJI PATTERN(64,63) ,ILL:ERROR CODE
320001 ***
330001 *****
340001 MOJINO=N
350001 CALL RETL8B(NVOL,MOJINO,IDI,IDC,IA,ILL)
360001 WRITE(6,6009) NVOL,MOJINO,IDI(2)
370001 6009 FORMAT(1H ,5X,'NVOL=',I4,5X,'MOJINO=',I4,5X,'JIS CODE=',Z4,/)
380001 IF(ILL.NE.0) THEN
390001 WRITE(6,6900)
400000 6900 FORMAT(1H ,*** READ ROUTINE(RET8B) ERROR ***)
410001 STOP
420001 END IF
430001 *** KANJI PATTERN NO PRINT ***
440001 CALL POUT2B(IA,I1,J1,ILL)
450002 ***** ICHI-OKKISA NO SEIKIKA *****
451004 CALL FILTH8(IA,I1,J1,IB)
460004 CALL NORMB3(IB,I1,J1,IA,I1,J1,ILL)
460104 CALL FILTH8(IA,I1,J1,IB)
461003 WRITE(6,*)'*** NORMB3 ***'
470003 CALL POUT2B(IB,I1,J1,ILL)
480001 *****
490001 *** SHOOTING (HISENKEI FILTERING)
500001 *** INPUT(1,2,3,4,5,6,7),OUTPUT(8)
510000 *** 1ST:INPUT 2D-IMAGE(IA(I1,J1)) ,2ND:1ST-SIZE OF IA
520001 *** 3RD:2ND-SIZE OF IA ,4TH:LOWER BOUND OF HORIZONTAL
530001 *** 5TH:UPPER BOUND OF HORIZONTAL ,6TH:LOWER BOUND OF VERTICAL

```

DATA SET NAME : W49215A.EXAMPLE.FORT(CREFILEH)

DATA SET NAME : W49215A.EXAMPLE.FORT(CREFILEH)

```

00010 *****
00020 *****
00030 ***** MAIN ROUTINE ( TOKUCHOU CHUUSITSU FILE NO SAKUSEI )
00040 ***** KAJYU HOUKOU SISU HISTGRAM HOU (ETL-8:16-CHI)
00050 ***** CODED BY S.TSURUOKA (ENG.ELECTRONICS, MIE UNIV.)
00060 ***** 1985.6.17
00070 *****
00080 *****
00090 ***** PARAMETER(I1=128,J1=127,I2=70,J2=70)
00100 ***** PARAMETER(JBL=2000,IH=7,JH=7)
00110 ***** PARAMETER(IHH=4,JHH=4,K1=5,K2=5,IP=1,JP=1,IS=2,JS=2)
00120 *****
00130 C ***** MCAMAX=MOJI CATEGORY SUU ( 1-956 )
00140 C ***** MNOMAX=MOJI SUU ( 1 CATEGORY NAI: 1-160 )
00150 *****
00160 ***** PARAMETER(MCAMAX=2,MNOMAX=2)
00170 *****
00180 ***** INTEGER IA(I1,J1),IB(I1,J1)
00190 ***** INTEGER IAA(I2,J2),IBB(I2,J2),IC(I2+2,J2+2)
00200 ***** INTEGER JBLX(JBL),JBLY(JBL),INDEX4(JBL)
00210 ***** INTEGER JHIST(IH,JH,4),IDI(13),JHISTK(IH,JH)
00220 ***** REAL HIST(IHH,JHH,4),NH(5,5),HISTK(IHH,JHH)
00230 ***** CHARACTER IDC(1)*8
00240 C *****
00250 *****
00260 C ***** OPEN(11,FILE='HW0885.EBH1WORK',STATUS='OLD',ACCESS='SEQUENTIAL',
00270 C ***** 1 FORM='UNFORMATTED')
00280 *****
00290 ***** DO 1000 MOJICA=1,MCAMAX
00300 C ***** WRITE(6,*) 'MOJI CATEGORY =',MOJICA
00310 ***** DO 2000 MOJINN=1,MNOMAX
00320 *****
00330 ***** NVOL:VOLUME NO.(1-32), MOJINP:VOL. NAI NO NUMBER(1-5)
00340 *****
00350 ***** NVOL=(MOJINN-1)/5+1
00360 ***** MOJINP=MOJINN-(NVOL-1)*5
00370 ***** MOJINO=MOJICA+(MOJINP-1)*956
00380 C *****
00390 *****
00400 *****
00410 ***** READ MOJI DATA IA(I1,J1) FROM DISK (ETL-8B)
00420 ***** INPUT(1,2), OUTPUT(3,4,5,6)
00430 ***** NVOL:VOLUME NUMBER(1-32), MOJINO:MOJI NUMBER(1-????)
00440 ***** IDI:JIS CODE HADO NO ZOKUSEI, IDC:KANJI NO YOMI
00450 ***** IA:MOJI PATTERN(64,63), ILL:ERROR CODE
00460 *****
00470 *****
00480 ***** CALL RETLBH(NVOL,MOJINO,IDI,IDC,IA,ILL)
00490 ***** WRITE(6,6009) NVOL,MOJINO,IDI(2),IDC
00500 ***** 6009 FORMAT(1H,5X,'NVOL=',I4,5X,'MOJINO=',I4,5X,'JIS CODE=',24,5X,
00510 ***** * 'YOMI:',Z8)
00520 ***** IF(ILL.NE.0) THEN
00530 ***** WRITE(6,6900)
00540 ***** 6900 FORMAT(1H,5X,'*** READ ROUTINE(RETLBH) ERROR ***')
00550 ***** STOP
00560 ***** END IF

```

```

00570 *** KANJI PATTERN NO PRINT ***
00580 C CALL POUT16(IA,I1,J1,ILL)
00590 **
00600 *****
00610 *** 8-KINBOU HEIKINCHI FILTER ***
00620 CALL FILTMB(IA,I1,J1,IB)
00630 C*** 2-CHIKA ***
00640 JTHD=1
00650 CALL BINARY(IB,I1,J1,IA,JTHD)
00660 *****
00670 *** SMOOTHING ( HISENKEI FILTERING )
00680 *** INPUT(1,2,3,4,5,6,7),OUTPUT(8)
00690 *** 1ST:INPUT 2D-IMAGE(IA(I1,J1)) ,2ND:1ST-SIZE OF IA
00700 *** 3RD:2ND-SIZE OF IA ,4TH:LOWER BOUND OF HORIZONTAL
00710 *** 5TH:UPPER BOUND OF HORIZONTAL ,6TH:LOWER BOUND OF VERTICAL
00720 *** 7TH:LOWER BOUND OF VERTICAL ,8TH:OUTPUT 2D-IMAGE(IB(I1,J1))
00730 *****
00740 ***
00750 I1=I1-1
00760 J1=J1-1
00770 CALL SMOOTH(IA,I1,J1,2,I11,2,J11,IB,1)
00780 C WRITE(6,*) ' *** SMOOTH OUTPUT ***'
00790 C CALL POUT2B(IB,I1,J1,ILL)
00800 C*** ICHI-OKISA NO SEIKIKA ***
00810 CALL NORMM3(IB,I1,J1,IAA,I2,J2,ILL)
00820 IF(ILL.NE.0)WRITE(6,*)'NORMM3 ERROR ILL=',ILL,'MOJINO=',MOJINO
00830 CALL FILTMB(IAA,I2,J2,IBB)
00840 C
00850 *****
00860 *** IA NI 1 BIT NO WAKU O TSUKERU ***
00870 *** INPUT(1,2,3,5), OUTPUT(4)
00880 *** 1ST:INPUT IMAGE(IB(I1,J1)) ,2ND:1ST-SIZE OF IB
00890 *** 3RD:2ST-SIZE OF IB ,4TH:OUTPUT IMAGE(IC(I1+1,J1+1))
00900 *** 5TH:0 WO OKU HABA
00910 *****
00920 CALL MSWAKU(IBB,I2,J2,IC,1)
00930 C WRITE(6,*) ' *** MSWAKU OUTPUT ***'
00940 I3=I2+2
00950 J3=J2+2
00960 C CALL POUT2B(IC,I3,J3,ILL)
00970 C
00980 *****
00990 *** BORDER FOLLOWING ( 8-RENKETSU ) USEING SPIDER(BDFL2)
01000 *** INPUT(1,3,4,10), OUTPUT(2,4,5,6,7,8,11)
01010 *** 1ST:INPUT 2D BINARY IMAGE(I,J)(0,1)(FRAME 1BIT 0)
01020 *** 2ND:OUTPUT 2D IMAGE (0,1,2(TSUISEKI TEN))
01030 *** 3RD:IMAGE SIZE(X) ,4TH:IMAGE SIZE(Y)
01040 *** 5TH:BORDER LIST(X)(RINKAKU NO SAKAIME NI 0),6TH:BORDER LIST(Y)
01050 *** 7TH:ARRAY SIZE OF 5,6 ,8TH:EFFISIENT SIZE OF 5,6
01060 *** 9TH:NUMBER OF CONECTED COMPONENT
01070 *** 10TH:CONECTED NUMBER(4 OR 8) ,11TH:ERROR CODE
01080 ***
01090 *****
01100 C
01110 CALL BDFL2(IC,IC,I3,J3,JBLX,JBLY,JBL,JBL1,NBD,8,KERR)
01120 IF(KERR.EQ.-1) THEN

```

DATA SET NAME : W49215A.EXAMPLE.FORT(FEATUREB)

DATA SET NAME : W49215A.EXAMPLE.FORT(FEATUREB)

```

540001 *** 7TH:LOWER BOUND OF VERTICAL ,8TH:OUTPUT 2D-IMAGE(IB(I1,J1)) 1100001
550001 ***** 1110001 6000 FORMAT(1H ,5X,'***** HOUKOU HISTGRAM MOJINO=',I4,' *****')
560001 *** 1120001 *** HISTGRAMS PRINT ***
570001 I11=I1-1 1130001 WRITE(6,6001) (((JHIST(I,J,K),I=1,IH),K=1,4),J=1,JH)
580001 J11=J1-1 1140001 6001 FORMAT(1H ,5X,7I4,2X,7I4,2X,7I4,2X,7I4)
590001 CALL SMOOTH(IB,I1,J1,2,I11,2,J11,IA,1) 1150001 ***
600001 WRITE(6,*) ' *** SMOOTH OUTPUT ***' 1160001 *** SECOND SAMPLING (7,7)-->(4,4) ***
610001 CALL POUT2B(IA,I1,J1,ILL) 1170001 *** READ WEIGHT FUNCTION(WW) ***
620001 ***** 1180001 ***
630001 *** IA NI 1 BIT HO WAKU O TSUKERU *** 1190001 CALL WGTIN4(WW,5,5,ILL)
640001 *** INPUT(1,2,3,5) ,OUTPUT(4) 1200001 DO 200 KH=1,4
650001 *** 1ST:INPUT IMAGE(IA(I1,J1)) ,2ND:1ST-SIZE OF IA 1210001 DO 210 I=1,IH
660001 *** 3RD:2ST-SIZE OF IA ,4TH:OUTPUT IMAGE(IC(I1+1,J1+1)) 1220001 DO 210 J=1,JH
670001 *** 5TH:0 WO OKU HABA 1230001 JHISTK(I,J)=JHIST(I,J,KH)
680001 ***** 1240001 210 CONTINUE
690001 CALL MSHAKU(IA,I1,J1,IC,1) 1250001 CALL WEIGT1(JHISTK,IH,JH,HISTK,IHH,JHH,WW,5,5,
700000 WRITE(6,*) ' *** MSHAKU OUTPUT ***' 1260001 * IP,JP,IS,JS,ILL)
710001 I2=I1+2 1270001 IF(ILL.NE.0) WRITE(6,*) 'WEIGT1 ILL=',ILL
720001 J2=J1+2 1280001 DO 220 I=1,IHH
730001 CALL POUT2B(IC,I2,J2,ILL) 1290000 DO 220 J=1,JHH
740001 C 1300001 HIST(I,J,KH)=HISTK(I,J)
750001 ***** 1310001 220 CONTINUE
760001 *** BORDER FOLLOWING ( B-RENKETSU ) USEING SPIDER(BDFL2) 1320001 200 CONTINUE
770001 *** INPUT(1,3,4,10), OUTPUT(2,4,5,6,7,8,11) 1330001 ***
780001 *** 1ST:INPUT 2D BINARY IMAGE(I,J)(0,1)(FRAME 1BIT 0) 1340001 *** FEATURE VECTOR OUTPUT ***
790000 *** 2ND:OUTPUT 2D IMAGE (0,1,2(TSUISEKI TEN)) 1350001 ***
800001 *** 3RD:IMAGE SIZE(X) ,4TH:IMAGE SIZE(Y) 1360001 WRITE(6,*) ' KAGJU HOUKOU SISUU HISTGRAM(4-HOUKOU)'
810001 *** 5TH:BORDER LIST(X)(RINKAKU NO SAKAIME NI 0),6TH:BORDER LIST(Y) 1370001 WRITE(6,6002) (((HIST(I,J,K),I=1,IHH),K=1,4),J=1,JHH)
820001 *** 7TH:ARRAY SIZE OF 5,6 ,8TH:EFFICIENT SIZE OF 5,6 1380001 6002 FORMAT(1H ,10X,4F5.1,5X,4F5.1,5X,4F5.1,5X,4F5.1)
830001 *** 9TH:NUMBER OF CONECTED COMPONENT 1390001 *****
840001 *** 10TH:CONECTED NUMBER(4 OR 8) ,11TH:ERROR CODE 1400001 *** LOOP END ***
850001 *** 1410001 100 CONTINUE
860001 ***** 1420001 STOP
870001 C 1430001 END
880001 CALL BDFL2(IC,IC,I2,J2,JBLX,JBLY,JBL,JBL1,NBD,8,KERR)
890000 IF(KERR,EQ,-1) THEN
900001 WRITE(6,*) 'BDFL2 ERROR(JBL OVER) NOJINO=',MOJINO
910001 GOTO 100
920001 ELSE
930001 *** BORDER LIST(X,Y) PRINT ***
940001 WRITE(6,6011) 'JBLX:',(JBLX(I),I=1,JBL1)
950001 6011 FORMAT(1H ,A,40I3)
960001 WRITE(6,6011) 'JBLY:',(JBLY(I),I=1,JBL1)
970001 END IF
980001 C
990000 *****
1000001 *** 4-HOUKOU SISU HISTGRAM O MOTOMERU **
1010001 *** INPUT(1,2,4,5,7,8,9,10) ,OUTPUT(3,6) **
1020001 *** 1ST:BORDER LIST(X) ,2ND:BORDER LIST(Y) **
1030001 *** 3RD:4-DIRECTION INDEX ,4TH:ARRAY SIZE OF 1,2,3 **
1040001 *** 5TH:EFFICIENT RANGE OF 1,2,3 ,6TH:HISTGRAMS OF SUB REGION **
1050001 *** 7TH:SIZE(X) OF 6 ,8TH:SIZE(Y) OF 6 **
1060001 *** 9TH:SIZE(X) OF IMAGE ,10TH:SIZE(Y) OF IMAGE **
1070001 *****
1080001 C
1090000 CALL HISTSS(JBLX,JBLY,INDEX4,JBL,JBL1,JHIST,IH,JH,I1,J1)

```

DATA SET NAME : W49215A.EXAMPLE.FORT(CREFILEB)

DATA SET NAME : W49215A.EXAMPLE.FORT(CREFILEB)

```

10001 *****
20001 *****
30001 ***** MAIN ROUTINE ( TOKUCHOU CHUUSITSU FILE NO SAKUSEI )
40001 ***** KAJYU HOUKOU SISU HISTGRAM HOU (ETL-8B)
50001 ***** CODED BY S.TSURUOKA (ENG.ELECTRONICS, MIE UNIV.)
60001 ***** 1985.6.17
70001 *****
80001 *****
90001 ***** PARAMETER(I1=64,J1=63,JBL=1000,IH=7,JH=7,NEND=2)
100001 ***** PARAMETER(IHH=4,JHH=4,K1=5,K2=5,IP=1,JP=1,IS=2,JS=2)
110000 *****
120001 C ***** MCAMAX=MOJI CATEGORY SUU
130001 C ***** MHOMAX=MOJI SUU ( 1 CATEGORY NAT)
140001 *****
150001 ***** PARAMETER(MCAMAX=10,MHOMAX=50)
160001 ***** INTEGER IA(I1,J1),IB(I1,J1),IC(I1+2,J1+2)
170001 ***** INTEGER JBLX(JBL),JBLJ(JBL),INDEX4(JDL)
180001 ***** INTEGER JHIST(IH,JH,4),IDI(13),JHISTK(IH,JH)
190001 ***** REAL HIST(IHH,JHH,4),WH(5,5),HISTK(IHH,JHH)
200001 ***** CHARACTER IDC=4
210000 C *****
220001 *****
230005 C ***** OPEN(11,FILE='HW08B5.EBBWORK1',STATUS='NEW',ACCESS='SEQUENTIAL',
240001 C ***** 1 FORM='UNFORMATTED')
250001 *****
260001 ***** NEND...MOJI NO.,NVOL=VOLUME NO.
270001 *****
280001 ***** NVOL=1
290001 ***** DO 1000 MOJICA=1,MCAMAX
300001 ***** WRITE(6,*) 'MOJI CATEGORY =' ,MOJICA
310000 ***** DO 2000 MOJINN=1,MHOMAX
320001 ***** MOJINO=MOJINN+(MOJICA-1)*160
330001 C *****
340001 *****
350001 *****
360001 ***** READ MOJI DATA IA(I1,J1) FROM DISK (ETL-8B)
370001 ***** INPUT(1,2) ,OUTPUT(3,4,5,6)
380001 ***** NVOL:VOLUME NUMBER(1-32) ,MOJINO:MOJI NUMBER(1-7777)
390001 ***** IDI:JIS CODE HADO NO ZOKUSEI ,IDC:KANJI NO YOMI
400001 ***** IA:MOJI PATTERN(64,63) ,ILL:ERROR CODE
410000 *****
420001 *****
430001 ***** CALL RETL8B(NVOL,MOJINO,IDI,IDC,IA,ILL)
440001 ***** WRITE(6,6009) NVOL,MOJINO,IDI(2)
450006 6009 FORMAT(1H ,5X,'NVOL=',I4,5X,'MOJINO=',I4,5X,'JIS CODE=',Z4)
460001 ***** IF(ILL.NE.0) THEN
470001 ***** WRITE(6,6900)
480001 6900 ***** FORMAT(1H ,5X,'*** READ ROUTINE(RET8B) ERROR ***')
490001 ***** STOP
500001 ***** END IF
510000 ***** KANJI PATTERN NO PRINT ***
520001 C ***** CALL POUT2B(IA,I1,J1,ILL)
530001 *****
540001 ***** 8-KINBOU HEIKINCHI FILTER ***
550001 ***** CALL FILTMB(IA,I1,J1,IB)
560001 *****
570001 ***** SMOOTHING ( HISENKEI FILTERING )
580001 ***** INPUT(1,2,3,4,5,6,7),OUTPUT(8)
590001 ***** 1ST:INPUT 2D-IMAGE(IA(I1,J1)) ,2ND:1ST-SIZE OF IA
600001 ***** 3RD:2ND-SIZE OF IA ,4TH:LOWER BOUND OF HORIZONTAL
610001 ***** 5TH:UPPER BOUND OF HORIZONTAL ,6TH:LOWER BOUND OF VERTICAL
620000 ***** 7TH:LOWER BOUND OF VERTICAL ,8TH:OUTPUT 2D-IMAGE(IB(I1,J1))
630001 *****
640001 *****
650001 ***** I11=I1-1
660001 ***** J11=J1-1
670001 ***** CALL SMOOTH(IB,I1,J1,2,I11,2,J11,IA,1)
680001 C ***** WRITE(6,*) ' *** SMOOTH OUTPUT ***'
690001 C ***** CALL POUT2B(IA,I1,J1,ILL)
700001 C *****
710000 *****
720001 ***** IA NI 1 BIT NO WAKU O TSUKERU ***
730001 ***** INPUT(1,2,3,5) ,OUTPUT(4)
740001 ***** 1ST:INPUT IMAGE(IA(I1,J1)) ,2ND:1ST-SIZE OF IA
750001 ***** 3RD:2ST-SIZE OF IA ,4TH:OUTPUT IMAGE(IC(I1+1,J1+1))
760001 ***** 5TH:0 WO OKU HABA
770001 *****
780001 ***** CALL MSWAKU(IA,I1,J1,IC,1)
790001 C ***** WRITE(6,*) ' *** MSWAKU OUTPUT ***'
800001 ***** I2=I1+2
810001 ***** J2=J1+2
820001 C ***** CALL POUT2B(IC,I2,J2,ILL)
830001 C *****
840001 *****
850001 ***** BORDER FOLLOWING ( 8-RENKETSU ) USEING SPIDER(BDFL2)
860001 ***** INPUT(1,3,4,10) , OUTPUT(2,4,5,6,7,8,11)
870001 ***** 1ST:INPUT 2D BINARY IMAGE(I,J)(0,1)(FRAME 1BIT 0)
880001 ***** 2ND:OUTPUT 2D IMAGE (0,1,2(TSUISEKI TEN))
890001 ***** 3RD:IMAGE SIZE(X) ,4TH:IMAGE SIZE(Y)
900000 ***** 5TH:BORDER LIST(X)(RINKAKU NO SAKAIME NI 0),6TH:BORDER LIST(Y)
910001 ***** 7TH:ARRAY SIZE OF 5,6 ,8TH:EFFISIENT SIZE OF 5,6
920001 ***** 9TH:NUMBER OF CONECTED COMPONENT
930001 ***** 10TH:CONECTED NUMBER(4 OR 8) ,11TH:ERROR CODE
940001 *****
950001 *****
960001 C *****
970001 ***** CALL BDFL2(IC,IC,I2,J2,JBLX,JBLJ,JBL,JBL1,NBD,8,KERR)
980001 ***** IF(KERR.EQ.-1) THEN
990001 ***** WRITE(6,*) 'BDFL2 ERROR(JBL OVER) NOJINO=',MOJINO
1000000 ***** GOTO 1000
1010001 ***** ELSE
1020001 ***** BORDER LIST(X,Y) PRINT ***
1030001 C ***** WRITE(6,6011) 'JBLX:',(JBLX(I),I=1,JBL1)
1040001 C6011 ***** FORMAT(1H ,A,40I3)
1050001 C ***** WRITE(6,6011) 'JBLJ:',(JBLJ(I),I=1,JBL1)
1060001 ***** END IF
1070001 C *****
1080001 *****
1090001 ***** 4-HOUKOU SISU HISTGRAM O MOTOMERU
1100000 ***** INPUT(1,2,4,5,7,8,9,10) ,OUTPUT(3,6)
1110001 ***** 1ST:BORDER LIST(X) ,2ND:BORDER LIST(Y)
1120001 ***** 3RD:4-DIRECTION INDEX ,4TH:ARRAY SIZE OF 1,2,3

```

DATA SET NAME : W49215A.EXAMPLE.FORT(CREFILEH)

```

01130      WRITE(6,*) 'BDFL2 ERROR(JBL OVER) NOJINO=',NOJINO
01140      GOTO 1000
01150      ELSE
01160 ***    BORDER LIST(X,Y) PRINT      ***
01170 C      WRITE(6,6011) 'JBLX:',(JBLX(I),I=1,JBL1)
01180 C6011    FORMAT(1H ,A,40I3)
01190 C      WRITE(6,6011) 'JBLY:',(JBLY(I),I=1,JBL1)
01200      END IF
01210 C
01220 *****
01230 ***    4-HOUKOU SISU HISTGRAM O MOTOMERU      **
01240 ***    INPUT(1,2,4,5,7,8,9,10)      ,OUTPUT(3,6)      **
01250 ***    1ST:BORDER LIST(X)      ,2ND:BORDER LIST(Y)      **
01260 ***    3RD:4-DIRECTION INDEX      ,4TH:ARRAY SIZE OF 1,2,3      **
01270 ***    5TH:EFFICIENT RANGE OF 1,2,3      ,6TH:HISTGRAMS OF SUB REGION      **
01280 ***    7TH:SIZE(X) OF 6      ,8TH:SIZE(Y) OF 6      **
01290 ***    9TH:SIZE(X) OF IMAGE      ,10TH:SIZE(Y) OF IMAGE      **
01300 *****
01310 C
01320      CALL HISTSS(JBLX,JBLY,INDEX4,JBL,JBL1,JHIST,IH,JH,I3,J3)
01330 C      WRITE(6,6000) NOJINO
01340 C6000    FORMAT(1H ,5X,'***    HOUKOU HISTGRAM NOJINO=',I4,'    ***')
01350 ***    HISTGRAMS PRINT      ***
01360 C      WRITE(6,6001) (((JHIST(I,J,K),I=1,IH),K=1,4),J=1,JH)
01370 C6001    FORMAT(1H ,5X,7I4,2X,7I4,2X,7I4,2X,7I4)
01380 ***
01390 ***    SECOND SAMPLING (7,7)--->(4,4)      ***
01400 ***    READ WEIGHT FUNCTION(WH)      ***
01410 ***
01420      CALL WGTIN4(WH,5,5,ILL)
01430      DO 200 KH=1,4
01440      DO 210 I=1,IH
01450      DO 210 J=1,JH
01460      JHISTK(I,J)=JHIST(I,J,KH)
01470      210 CONTINUE
01480      CALL WEIGT1(JHISTK,IH,JH,HISTK,INH,JHH,WH,5,5,
01490      * IP,JP,IS,JS,ILL)
01500      IF(ILL.NE.0) WRITE(6,*) 'WEIGT1 ILL=',ILL
01510      DO 220 I=1,INH
01520      DO 220 J=1,JHH
01530      HIST(I,J,KH)=HISTK(I,J)
01540      220 CONTINUE
01550      200 CONTINUE
01560 ***
01570 ***    FEATURE VECTOR OUTPUT      ***
01580 ***
01590      WRITE(6,*) 'KAGYU HOUKOU HISTGRAM (4-HOUKOU)'
01600      WRITE(6,6002) (((HIST(I,J,K),I=1,INH),K=1,4),J=1,JHH)
01610      6002    FORMAT(1H ,10X,4F5.1,5X,4F5.1,5X,4F5.1,5X,4F5.1)
01620 C
01630 ***    FILE OUTPUT      ***
01640      WRITE(11) MOJICA,MOJINH,HIST
01650 *****
01660 ***    LOOP END      ***
01670      2000 CONTINUE
01680      1000 CONTINUE

```

DATA SET NAME : W49215A.EXAMPLE.FORT(CREFILEH)

```

01690      CLOSE(11,STATUS='KEEP')
01700      STOP
01710      END

```

DATA SET NAME : W49215A.EXAMPLE.FORT(CREFILEB)

```

1130001 *** 5TH:EFFICIENT RANGE OF 1,2,3 ,6TH:HISTGRAMS OF SUB REGION **
1140001 *** 7TH:SIZE(X) OF 6 ,8TH:SIZE(Y) OF 6 **
1150001 *** 9TH:SIZE(X) OF IMAGE ,10TH:SIZE(Y) OF IMAGE **
1160001 *****
1170001 C
1180001 CALL HISTSS(JBLX,JBL Y,INDEX4,JBL,JBL1,JHIST,IH,JH,I1,J1)
1190001 C WRITE(6,6000) MOJINO
1200000 C6000 FORMAT(1H ,5X,'*** HOUKOU HISTGRAM MOJINO=',I4,' ***')
1210001 *** HISTGRAMS PRINT ***
1220001 C WRITE(6,6001) (((JHIST(I,J,K),I=1,IH),K=1,4),J=1,JH)
1230001 C6001 FORMAT(1H ,5X,7I4,2X,7I4,2X,7I4,2X,7I4)
1240001 ***
1250001 *** SECOND SAMPLING (7,7)-->(4,4) ***
1260001 *** READ WEIGHT FUNCTION(WW) ***
1270001 ***
1280003 CALL WGTIN4(WW,5,5,ILL)
1290001 DO 200 KH=1,4
1300000 DO 210 I=1,IH
1310001 DO 210 J=1,JH
1320001 JHISTK(I,J)=JHIST(I,J,KH)
1330001 210 CONTINUE
1340002 C CALL WEIGT4(JHISTK,HISTK,WW)
1350002 CALL WEIGT1(JHISTK,IH,JH,HISTK,IHH,JHH,WW,K1,K2,
1360002 * IP,JP,IS,JS,ILL)
1370001 IF(ILL.NE.0) WRITE(6,*) 'WEIGT1 ILL=',ILL
1380001 DO 220 I=1,IHH
1390001 DO 220 J=1,JHH
1400001 HIST(I,J,KH)=HISTK(I,J)
1410000 220 CONTINUE
1420001 200 CONTINUE
1430001 ***
1440001 *** FEATURE VECTOR OUTPUT ***
1450001 ***
1460001 C WRITE(6,*) 'KAGYU HOUKOU HISTGRAM (4-HOUKOU)'
1470001 C WRITE(6,6002) (((HIST(I,J,K),I=1,IHH),K=1,4),J=1,JHH)
1480001 C6002 FORMAT(1H ,10X,4F5.1,5X,4F5.1,5X,4F5.1,5X,4F5.1)
1490001 C
1500001 *** FILE OUTPUT ***
1510003 WRITE(11) MOJICA,MOJINN,HIST
1520000 *****
1530001 *** LOOP END ***
1540001 2000 CONTINUE
1550001 1000 CONTINUE
1560001 CLOSE(11,STATUS='KEEP')
1570001 STOP
1580001 END

```