T2R2 東京科学大学 リサーチリポジトリ
Science Tokyo Research Repository

論文 / 著書情報
Article / Book Information

題目(和文)	高精度文字認識のための特徴抽出フィルタマスク学習の研究
Title(English)	
著者(和文)	黒澤由明
Author(English)	Yoshiaki Kurosawa
出典(和文)	学位:博士(工学), 学位授与機関:東京工業大学, 報告番号:甲第7435号, 授与年月日:2008年9月25日, 学位の種別:課程博士, 審査員:佐藤 誠
Citation(English)	Degree:Doctor (Engineering), Conferring organization: Tokyo Institute of Technology, Report number:甲第7435号, Conferred date:2008/9/25, Degree Type:Course doctor, Examiner:
学位種別(和文)	博士論文
Type(English)	Doctoral Thesis

Powered by T2R2 (Science Tokyo Research Repository)

高精度文字認識のための特徴抽出フィルタマスク学習の研究

黒沢由明

目 次

第 1 章　序論（高精度文字認識をどう実現するか） **1**
1.1　研究意義 2
1.1.1　文字認識技術の現状 2
1.1.2　高精度文字認識の課題と解決策 2
1.1.3　類似文字識別の重要性 4
1.1.4　競合学習の導入による文字認識高精度化への期待 4
1.2　本研究の要点 5
1.2.1　本研究におけるフィルタマスク学習開発のねらい 5
1.2.2　本研究の意義 6
1.3　研究対象となる文字認識システムの構成 6
1.3.1　研究分野 6
1.3.2　文字認識システムの全体構成 6
1.3.3　類似文字識別部の構成 7
1.3.4　学習システムの構成 8
1.4　FML とは何か 8
1.4.1　フィルタリングによる特徴抽出 8
1.4.2　フィルタマスクを学習によって生成する 9
1.4.3　非一様マスクの導入 10
1.4.4　FML 開発のアプローチ 11
1.4.5　FML 開発における新提案 11
1.4.6　研究の位置付けと目的 12
1.5　論文の構成 14

第 2 章　準備（文字認識高精度化と FML） **16**
2.1　文字認識をどう高精度化するか 16
2.1.1　個別文字認識の高精度化 16
2.1.2　構造解析的パターン認識と統計的パターン認識の相補性 16
2.2　輪郭構造マッチング法 17
2.2.1　輪郭構造マッチング法における特徴抽出とパターン照合 17
2.2.2　輪郭構造マッチング法における類似文字識別法 19
2.3　パターンの球面分布に基づく部分空間法の高精度化 20
2.3.1　超球面上のパターン分布 20
2.3.2　部分空間法と疑似ベイズとの比較 22
2.4　類似文字識別を高精度化する FML 23
2.4.1　FML の導入 23
2.4.2　FML における競合学習導入の背景 23

第 3 章　フィルタマスクの基礎検討 **24**
3.1　特徴抽出のためのフィルタリング 24
3.1.1　線形フィルタ 24

3.2 疑似濃淡表現におけるフィルタリング 26
3.2.1 疑似濃淡表現とは何か 26
3.2.2 疑似濃淡表現の従来方式 26
3.2.3 多段分割量子化法 28
3.2.4 平均濃度近似法 29
3.2.5 各方式による２値表現結果の比較 29
3.2.6 FML 開発に向けての開発指針 31

第 4 章 フィルタマスク学習の提案 32
4.1 特徴抽出学習と特徴抽出フィルタ学習 32
4.1.1 特徴抽出学習の考え方 32
4.1.2 特徴抽出フィルタ学習とその種類 33
4.2 フィルタマスク学習（ＦＭＬ）の基本構造 34
4.2.1 ＦＭＬの枠組 34
4.2.2 認識精度評価式と降下法 35
4.2.3 基礎的なＦＭＬの更新式 36
4.3 確率的降下法およびニューラルネットとの関係 38
4.3.1 確率的降下法 38
4.3.2 ニューラルネットとの関係 39
4.4 内積型類似度に基づくＦＭＬ 41
4.4.1 内積演算を用いた正規化無視型ＦＭＬ 41
4.4.2 内積演算を用いた正規化考慮型ＦＭＬ 41
4.4.3 正準化 42
4.4.4 内積演算を用いた正準化考慮型ＦＭＬ 43

第 5 章 フィルタマスク学習の実現 44
5.1 文字認識への応用 44
5.1.1 全体構成 44
5.1.2 大きさ正規化 45
5.1.3 フィルタリング 46
5.1.4 認識部 47
5.1.5 学習部 47
5.2 FML の高度化 48
5.2.1 多面化による FML の高度化 48
5.2.2 多面化 FML における学習の対称性の問題 50
5.2.3 多面化 FML における直交化の手法 50
5.2.4 多重化による FML の高度化 51
5.3 FML の実装 53
5.3.1 特徴抽出部の実装 53
5.3.2 認識部の実装 54
5.3.3 学習部の実装 54

第 6 章 文字認識における学習実験 55
6.1 実験環境 55
6.1.1 実験システム 55
6.1.2 実験で用いた文字パターン・データ 56
6.2 ユークリッド距離に基づく FML の学習実験 56

6.2.1 実験で用いた初期フィルタマスク 57
6.2.2 類似文字ペア「ボポ」を対象とする FML の学習実験 57
6.2.3 類似文字ペア「間問」を対象とする FML の学習実験 60
6.2.4 類似文字グループ「門構え」を対象とする FML 実験 61
6.2.5 認識精度の評価と結論 61
6.3 内積型フィルタマスク学習 63
6.3.1 内積型とユークリッド距離型 FML の比較 63
6.3.2 内積型フィルタマスク学習での認識精度 64
6.4 多面化 FML の実験 65
6.4.1 ガウス関数型初期マスクによる多面化 FML の実験 65
6.4.2 ディレイド・スタートによる多面化 FML での認識精度 67
6.4.3 方向性を持った初期マスクによる多面化 FML の実験 68
6.4.4 方向性を持った初期マスクによる多面化 FML での認識精度 69
6.5 多重化 FML の実験 71
6.5.1 多重化 FML で生成されるフィルタマスク 71
6.5.2 多重化 FML で達成される認識精度 73
6.6 FML で生成したマスクを LVQ 以外の認識方式に適用する実験 74
6.6.1 「ボポ」「間問」の文字セットで１面 FML のケース 74
6.6.2 「門構え」の文字セットで多面化 FML のケース 75
6.6.3 「門構え」の文字セットで多重化 FML のケース 76
6.6.4 文字認識実験結果に対する考察 77
6.7 学習プロセスにおける課題と解決策 77
6.7.1 パラメータ設定の問題 77
6.7.2 学習の崩壊とその対策 79

第 7 章 結果と結論 **83**

付 録 A 単純類似度型 FML における正規化と正準化 **85**
A.1 正規化考慮の場合 85
A.2 正準化考慮の場合 86

付 録 B LVQ の発散問題 **89**
B.1 LVQ の発散問題の概要 89
B.2 窓関数が step 関数の場合 89
B.3 窓関数が sigmoid 関数の場合 91
B.4 窓関数が sato 窓関数の場合 93

付 録 C 学習崩壊の観察 **96**
C.1 降下法型学習プロセスの可視化 96
C.2 LVQ 可視化による学習崩壊の観察 96
C.3 評価関数の断面 98

謝辞 **102**

参考文献 **103**

第１章　序論（高精度文字認識をどう実現するか）

長い地道なパターン認識研究の努力と執念にもかかわらず，その成果を人の認識能力と比較する時，そこにはまだ多くの解決すべき課題が残されている．

手書き文字認識の例で言えば漢字の類似パターン識別がその一つである．

カタカナや漢字には極めて類似した類似文字ペアが幾つも存在している．極端な例としてカタカナの「エカタトニハヘメリロ」については他の字種の中にまったく同形のものがある．このように文字形状に類似文字や同形文字が多いのは人が文脈知識や単語知識を用いて文字を読んでいるからである．人は文字の１つ１つの認識結果の曖昧さを文や文脈の知識で補いつつ文字読み取りをしている．したがって人間にとって類似文字はもちろんのこと同型文字さえ問題にならない．

これは文字認識設計者にとっては困った問題である．皇帝では無いから「文字の形を変えろ」とも言えない．人が平然と読んでいる以上「そりゃ無理だ」とも言えない．その文字をそのまま読むしかない．OCR を作る以上この問題を避けて通るわけにはいかない．

「良く似た文字をどう読むか」が本研究の１つの目的である．

もう１つの課題は「学習」である．

OCR が世に出されてほぼ半世紀．電子化が進んで紙が無くなり OCR は必要なくなると予言されてから久しい．紙のほうは予言通りではなく，むしろ増えているように見える．OCR 市場のほうはどうであろうか？ジワジワとその影響を受けているようにも見えるが，一方で OCR の性能向上もあって，その適用範囲を広げ健闘している面もある．台数ベースではむしろ拡大トレンドにある．それよりも OCR の価格低下のほうが問題で，これが金額ベースでの市場規模の停滞につながっている．

OCR の主要マーケットは規格化された帳票の大量読み取りで，比較的ていねに記載された文字が相手である．申込書のたぐいのニーズである．こうした分野では紙の利便性が並大抵で無く，電子化が進むこの世の中でも，しぶとく紙を用いたデータ・エントリは使われ続けている．

とは言うものの，大量入力，大規模物件は減少しつつあるように見え，１つ１つの物件が小形化してきている．一方で携帯カメラで文字を読むとか，数千種類もある雑多な帳票を読みたいとか，不定形なものを読み取り対象とするニーズが顕在化してきている．このような物件は小量多品種である．１件１件読み取り条件が異なり，それぞれに固有の技術的難しさがある．

この小量多品種の分野は規格化された帳票読み取りの分野に比べ，そのニーズは大きい．にもかかわらず，この分野はずっとこれまで放置されて来た．課題が１件１件の異なり開発コストが大きいにもかかわらず物件の大きさが小さいためである．ビジネスにならないのである．しかしながら OCR の市場規模を大きくしたいのならこの問題に立ち向かうしかない．

人はどんなものでも，どんな環境下でも，どんな時でも，苦もなくそこにある文字を読んでしまう．個々の物件に対する開発コストは必要ない．物件の規模も関係ない．むしろ大量読み取りの方が不得意だ．生まれたばかりの時には何もできない赤ちゃんだったのにである．確かに親や学校が文字の読み書きは教えたが，読むべき文字の様々なコンディションに対して，個々にああしろ，こうしろとは教えていない．人は勝手に学習して，「どんな状況下でも文字を読める」という能力を身に付けるのである．小量多品種の読み取りが苦手だなんて言う人の話なんて聞いたことが無い．人は学習によって「どんな文字」でも読めるようになる．何でもかんでも読める OCR を作りたい．その為のキーワードは人の学習にある．

「どんな文字でも読める OCR の学習」が本研究のもう１つの目的である．

もちろんこの２つの目的は究極の目標であり，この研究はそのための一里塚である．すぐにできるわけが無い．

1.1 研究意義

1.1.1 文字認識技術の現状

文字認識の現状について報告しよう．

国際的な文字認識の研究活動は特にアジアからの参加が増えてますます活発化する勢いにある反面，国内での学術的な面での研究はスローダウンしてきている．一方でメーカでの技術開発は維持されており，これは OCR の市場規模がほぼ現状維持と言って良い状態から来ているのであろう．

国内での学術的な面でのスローダウンの大きな原因の１つは「やれそうな事はやりつくしてしまった」ことにある．今，学会で何かアイディアを発表したとしても，たいがい過去に似たような事例が提案されている．新しく無いと言って批判されることになる．そこで一般の研究者の目を引こうとすると，よほど奇抜なものを出さないといけなくなる．また，これまでの研究で「高い認識精度が達成済」と思われている面もある．一部の研究では公開されている文字パターン・データに対して 99% を越える高い認識精度が示されており，そう思われても無理がない．実際はどうなのであろうか？

現在，製品化されている OCR では，きれいに印刷した印刷物のレベルの活字（漢字を含む）や申込書などに普通に記入するレベルの手書数字については，かなり高い認識精度が実現できる．状況にもよるが人の認識精度と比べて遜色ない程度にある．特定の高品質のカタカナ，英数字の活字認識では人の認識精度を上回った事例もある．手書の数字に関してはある程度の品質のものであれば，その認識精度は人の精度に近いレベルにある．

これに対して問題なのが手書き漢字で，英字，カナもそれについで読み取り率が低い．事例によって様々だが認識精度は 90% 〜 99% 程度で，場合によっては数十%台までに下がってしまう．

もう１つ問題なのが低品質な活字で，OCR にとって読み取りにくい活字が世の中には数多くある．どういうわけだかユーザは好んでそういう活字を読ませたがる傾向にある．と言うよりも，実は小量多品種のケースでは OCR したいデータはきちんと管理されていない場合が多く，それが低品質活字の事例の多さにつながっている．そもそもきちんと管理できるぐらいだったら，わざわざ OCR を使うような事にしないであろう．

さらにもう１つの問題はドキュメントや帳票全体にかかわることで，多様なレイアウトに対処しなければならない問題，印字の劣化や汚れ，また低品質な撮像，画像のゆがみなど，ドキュメント全体の品質が悪い場合の問題である．

このように文字認識にはまだ解決すべき課題が多く残されている．人と比べれば全然読めていないと言って良く，その意味では今後の OCR の発展にとって国内での学術的な面での研究が活発化することが望ましい．そのためには，より現場に近い低品質の文字パターン・データベースやドキュメントイメージ・データベースを公開する必要があり，また新規性に囚われることなく実用レベルでの性能が出るかどうかにフォーカスした研究が必要である．その意味ではメーカーの研究機関に今よりももっと活動の余地があるように思う．

本研究はこのような立場で行なわれている．

1.1.2 高精度文字認識の課題と解決策

本研究では高精度文字認識をターゲットとし，特に手書文字にフォーカスした実験を行なうが，ここではまずその前に文字認識の課題について，以下，手書きの場合を考察する．

手書文字認識の課題をここでは３つに整理している．

1）「大変形」

これは図 1.1 の課題１の例に示すようなものである．この例では文字が６０度近く回転してほとんど横倒しになっているもの，３の下側の文字線が下にまっすぐに伸びてしまっているもの，逆に上に大きくはてているものを例としてあげている．これは文字を重ねてみると重なる部分の少ない変形である．

2）「多様な変形」

この例を同じ図の課題２に例示した．これは文字が様々な要因によって変形しているケースで，その変形の

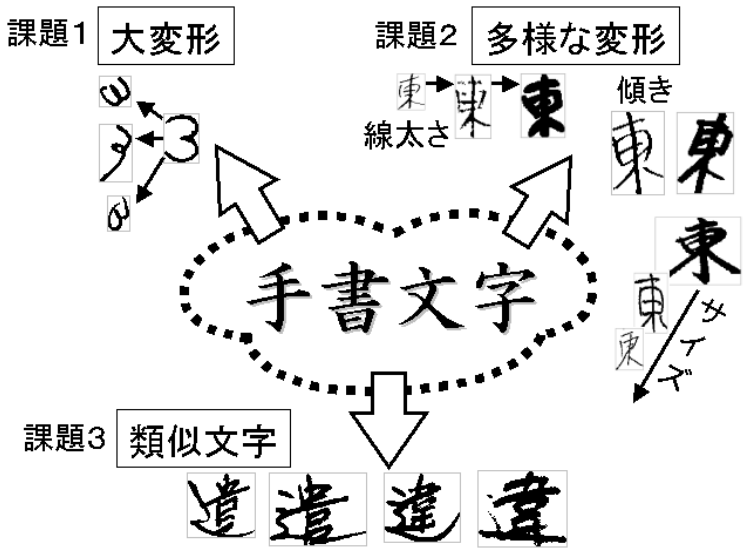


図 1.1: 手書文字認識の３つの課題

要因には例えば文字線の太さ，傾き，文字サイズなど様々なものがある．大変形でも変形の性格が決まっていれば何とか対応できるが，この図のようにあまりにも多様に変形していると対処のしかたが難しくなる．

３）「類似文字」
これは極めて似ている文字のことで，当然の事ながら識別が難しい．

これら３つの課題は手書き文字の変形として混在しており，それらすべてに１つのやり方で対応できるような方式を作ることは大変に難しい．このため，この問題に対応するには適材適所，課題に応じて得意の認識手法をうまく使って臨機応変に認識して行く考え方が必要となる．これら３つの課題に対してそれぞれ有効な認識方式があり，
大変形に「構造解析的パターン認識」，
多様な変形に「統計的パターン認識」，
類似文字に「類似文字識別方式」，
の対応関係がある．もちろん厳密な意味では無いが，おおむねこのような関係がある．大事なことは各方式にはそれぞれ一長一短があり，それらを組合わせることによって，より強力な文字認識系を構築することができるということである．

この視点で言えば本研究は文字認識全体として，構造解析的手パターン認識，統計的パターン認識，類似文字識別方式のそれぞれについて取り組んだものであり全体として文字認識精度向上に取り組んだものだある．まず構造解析的パターン認識としては新たに輪郭構造マッチング法を開発して自由手書文字認識を可能とし，また統計的パターン認識に関しては重み付き部分空間法として高性能を発揮できる新方式の提案を行なった．また類似文字識別方式についてはフィルタマスク学習を開発し，実験によりその有効性を確認した．本報告の中では特に最後の学習を用いた類似文字識別方式について詳しく報告している．

高精度文字認識に必要な様々な技術の中で，統計的パターン認識や構造解析的パターン認識にはそれぞれ有力な手法が開発されてきたが，類似文字識別に関しては相対的に研究が少なく，その性能は十分とは言えない状況であった．その意味で特に漢字やカタカナの認識においては類似文字識別は重要な課題となっている．また学習の概念は今後の文字認識にとって重要な課題である．その理由で，本報告ではこの類似文字識別に学習を導入した方式を特に取り上げてフォーカスしている．

1.1.3 類似文字識別の重要性

類似文字識別はもともと人間にとっても難しい．しかしながら特に素早く読む時などは注意深く類似文字の差を気にして読んではいないので，この場合には類似文字識別がきちんと行なわれていない可能性がある．特に注意しなければ良く似た文字に間違えた誤字を読むのに，それほどの違和感はない．その理由は人が文脈知識や単語知識を用いて総合的に文字の認識を行なっているからであり，もちろん OCR においても単語の知識を用いる後処理の技術は使われている．特に「エカタトニハヘメリロ」のような同形文字に関してはこの後処理技術は必須である．

しかしながら類似文字については，仮にそこに１文字だけ類似文字が記載されていたとしても人はかなりの確かさでそれを識別することができる．実は，注意深く読む時には類似文字識別に後処理はいらないのである．このような人の類似文字識別の能力は現在の OCR の実力と比べると遥かに強力である．OCR を人の能力に近付けるには類似文字識別の高精度化が必須である．

また実際に OCR を開発する上でも，類似文字識別を後処理に任せておけば良いという方針は経験的にはまったく不適切な考え方である．実は OCR の後処理自体も人の能力に比べると大幅に劣っているため，十分な誤読のリカバーができないのである．それ故，類似文字識別はトータルな認識精度にかなり効いてくるファクターであり，重要視すべき開発アイテムとなっている．

1.1.4 競合学習の導入による文字認識高精度化への期待

競合学習の導入

類似文字識別を実現しようとすると，その方法には２つのアプローチがある．１つは，どの類似文字にも共通の１つの認識部で対応するというもの．もう１つは，類似文字ペアまたは類似文字グループごとに異なる認識部を構築していくものである．一般に前者のアプローチはあまり例が無く，あったとしても，それは単に類似文字に強い文字認識方式とされるであろう．従って類似文字識別と言った場合には後者のアプローチを指すと思って良く，本研究でも後者のアプローチを採用している．

このアプローチは類似ペアごとに最適な識別系を用意するという考え方になるが，その場合，識別系はペアごとに人手によって作られる事が多い．しかし漢字認識のように類似ペアが多い場合，人手による方法は困難であり類似ペアごとに最適識別系を自動的に作り出すことが必要となる．

最適識別系を自動的に作り出すとは，すなわち学習である．本研究ではここに逐次学習の一種である競合学習を導入している．逐次学習とは，学習パターンの認識，それに基づく辞書更新，その更新された辞書による学習パターンの認識,... と認識＋学習を繰り返して行く方式のことである．競合学習とは，通常ある「文字」の辞書を作成する時にその「文字」の学習パターンのみを使用するのに対して，ライバルの「文字」の学習パターンをも用いる方式である．ライバルの情報を用いる競合学習は「敵を知り，己を知り」の強力な学習方式で，通常の学習より高性能であることが多い．本研究では特にこれが類似文字識別に適していると判断し，用いることとした．

仕様のない学習

「学習」の研究にも長い歴史があり，これまでにも様々な研究がなされてきている．しかしながらパターン認識の研究と同様，人間の能力に比べると遥かに水準の低いものしかできていないように感じる．

根本的な違いは我々の作った学習が「あらかじめ設計された範囲内での学習」であるのに対して，人の学習は「何を学習してしまうか見当も付かない」ところにある．子ども達を育てていると，そう言う経験をする事が多い．ちっとも設計通りに育たないのである．

この「設計された範囲外での学習」，言葉を替えれば,「仕様なき学習」は難しい課題である．ちょうど無目的に構成した大規模なニューラルネットがそれにあたるのであろうが，こうしたアプローチは今の所うまく行っていないようである．経験的にはターゲットの想定に基づいて学習システムをきちんと構成しないとうまく学習し

ないし効率も悪い．だからと言って，いつまでも「あらかじめ設計された範囲内での学習」をやっていても人間の足元にさえおよぶ事ができない．

前の節で今後の OCR の市場拡大にとって小量多品種のドキュメント，帳票の読み取りが重要である事を述べたが，この小量多品種の世界こそ「仕様の無い世界」,「しようのない物件」である．ビジネス的には筋悪のこの種の物件をうまくビジネス化できる鍵が上記の「仕様なき学習」技術である．人並とは行かないまでも，それにちょっとでも近付く技術が出てくれば，このような「しようの無い」ケースでも容易に対応できるようになる可能性がある．

本研究でもこの「仕様なき学習」を目指したいと考えている．しかしながら当面それは非現実的と言わざるを得ない．さきにも述べたように経験的にはそういうアプローチはたいがい失敗するからである．そもそも本研究には実用的な OCR の開発という課題があり，これとそれとが両立しない．とは言うものの，それでも本研究の取り組みの中で，ちょっとでも「仕様なき学習」に近付くようなアプローチを取りたいと考えた．具体的には本研究で取り組んだ学習方式の次の部分にその考え方が反映されている．

「仕様なき学習」と「仕様有り学習」のハイブリッド

本研究では学習対象のパラメータをフィルタの構造を決める数式の係数ではなく，フィルタで使われるマスクの値そのものとした．この考え方は単純でかつパラメータ量が膨大になっただけなので，学術的には後戻りした提案のような印象を与える．しかしながら，この仕様は上記の考えに基づくもので，できるだけ人為的な設計行為を無くそうとしたものである．もちろん，一方で OCR に適していると想定される設計行為もあるので，本研究で採用した方式は OCR に適した設計行為と非設計行為のハイブリッド，すなわち「仕様なき学習」と「仕様有り学習」のハイブリッドとも言える．

パターン照合以外への競合学習の導入

本研究で試みた類似文字識別だけでなく，今後の文字認識の研究では学習がキーになってくる．このため文字認識の各パートにそれぞれ学習の概念を採り入れて行く必要がある．ところが，これまでの文字認識研究に関しては特徴ベクトルと辞書ベクトルとを比較するパターン照合の分野で学習の研究が盛んに行なわれて来たものの，それ以外の分野では学習のアプローチはほとんど取られてこなかった．

本研究の重要なモチベーションの１つはこのパターン照合以外の部分への学習の導入である．その最初のアプローチとして特徴抽出部への学習の適用を試みたのである．

1.2 本研究の要点

1.2.1 本研究におけるフィルタマスク学習開発のねらい

本研究で取り上げる特徴抽出学習の具体的な手法はフィルタマスク学習である．この方式は文字パターンの特徴抽出で用いられるフィルタリングにおいて，そのフィルタのマスク値を学習する方式である．この方式の構造はニューラルネットに近いものであり，見方によればニューラルネットの一つであると解釈することも可能である．

しかしながら本研究はニューラルネット研究の延長線上の改良では無い．あくまでもニューラルネットに代表されるような学習方式の対極に位置している方式がベースとなっている．その方式とは文字認識の分野で用いられているような伝統的なパターン認識手法であり，本方式はその延長線上の方式なのである．

実際の OCR 開発を行なっていると，ニューラルネットのような学習が，結局のところ伝統的な手法にかなわないというような経験を多く持つ．実際のある OCR の実用化事例で，ニューラルネット型の OCR と従来型の OCR の読取り精度を比較したことがあった．双方かなりの開発コストをかけた OCR であり，ちょっとした実験

室レベルの話ではない．その結果，従来型の OCR の方が誤読が１桁少ないということになった．大差である．もちろんこれは一例で，一般論としてニューラルネット型が従来型よりも劣るということを示すものでは無いが,「ニューラルネットを持ってきても簡単に従来型を凌駕できるものでもない」ということを示す事例である．

本研究は伝統的な OCR 開発手法の中にニューラルネットで用いられるような競合学習方式を導入したものである．従来型の OCR の経験を全部放棄して，新たにニューラルネット型でそれを全てリプレースしようとしたものでは無い．結果的にその構造がニューラルネットに近いものとなったが，設計レベルで見た時には文字認識での経験が数多く生かされたものとなっている．伝統的な文字認識設計手法と相性の良い方式なのである．

具体的にはマスクのサイズや位置などのスペック，マスクの初期値，類似度や相違度の扱い方，位置によって異なるマスク値を用いる非一様マスクの導入，マスクを直接目視で観察できるマスクビューワの開発，など様々な部分にその考え方が反映されている．

学術的にはニューラルネット型と同じ構造という分類は可能だが，研究そのものは伝統的な OCR 設計における特徴抽出部に競合学習を導入するという，新しいチャレンジなのである．

1.2.2　本研究の意義

以下，本研究の意義をまとめておく．

文字認識への貢献：

（１）文字認識の高精度化による OCR ビジネスへの貢献
（２）高性能な構造解析的パターン認識手法と新しい統計的パターン認識手法の提案
（３）類似文字識別方式の高精度化の実現
（４）文字認識におけるパターン照合以外への競合学習導入の第一歩

学習技術への貢献：

（５）伝統的な OCR 設計手法に競合学習を導入する試み
（６）仕様なき学習を実用化する試み
（７）実際の課題に適用することによる競合学習方式の強化
（８）文字以外の一般のパターン認識技術への波及効果

1.3　研究対象となる文字認識システムの構成

1.3.1　研究分野

本研究の対象分野はパターン認識であり特に文字認識分野をターゲットとしており，この文字認識の中でも重要な類似文字識別技術を主要な課題としている．この類似文字識別を高精度化するための方法として特徴抽出技術の部分に焦点を当てて，これに競合学習を適用した．特徴抽出にはフィルタリング技術を用いた方式を採用している．したがって本研究における類似文字識別の研究は，その分野を細かい分類で言えば「類似文字識別のための競合学習を用いたフィルタリング型特徴抽出方式の自動生成」ということになる．

実際に関連する研究分野は「文字認識」,「特徴抽出」,「類似文字識別」,「フィルタリング」,「逐次学習」,「競合学習」,「ニューラルネット」などである．

次に本研究で対象とする文字認識システムの構成について説明する．

1.3.2　文字認識システムの全体構成

一般に文字認識システムは図 1.2 に示すように大きくわけて「レイアウト解析」,「文字切り出し」,「個別文字認識」,「後処理」の４つからなる．後処理は知識処理とも呼ばれるが本稿では後処理と呼ぶ．個別文字認識は１つ１つの文字を読み取る技術で通常は単に「文字認識」と呼ばれる．しかしながらこの４つのパート全部あるい

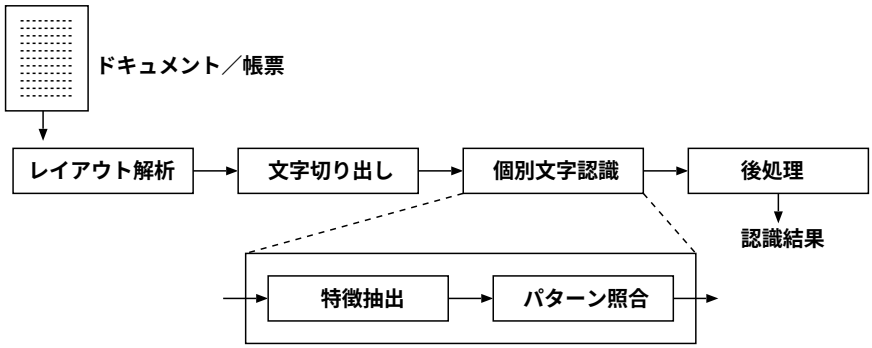


図 1.2: 文字認識システムを構成する４つの要素技術

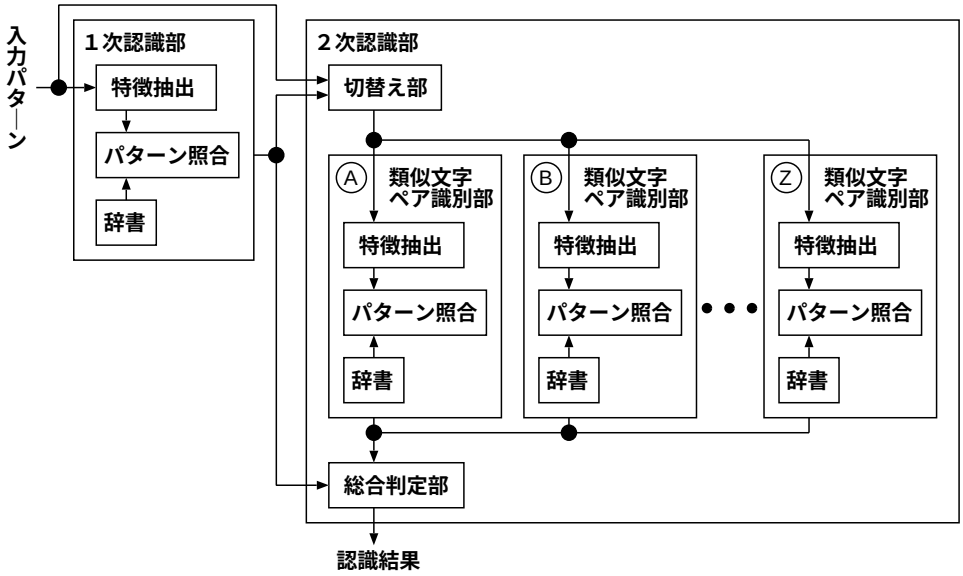


図 1.3: 類似文字識別を用いた文字認識システムの例

は一部をもって文字認識と呼ばれることも多いので，そのような場合との混同を避けるためにこの部分に「個別」の文字を付けて「個別文字認識」と呼ぶ．本研究はこの個別文字認識を対象とするものであり特に手書き文字を認識対象としている．

また，この図に示すように個別文字認識は，さらに「特徴抽出」と「パターン照合」に分かれる．特徴抽出は入力された文字の画像から特徴量を数値の組として取り出す部分である．また「パターン照合」とはその数値の組と辞書と呼ばれるデータとの比較を行なう部分である．辞書とは，やはり数値の組で文字カテゴリ（一種類の文字のこと,「０」とか「A」のことをカテゴリと呼ぶ）ごとに対応付けて作成されている．この照合において，もっとも良く照合されたと判断された辞書に対応するカテゴリを認識結果として出力する．なお，この「パターン照合」は様々に呼ばれていて固定した呼び方がないが，ここでは「パターン照合」としておく．

以下，本報告でおもにフォーカスして報告する類似文字識別についてその概要を述べる．

1.3.3 類似文字識別部の構成

類似文字識別を含む OCR システムの構成方法には幾つか有り得る．図 1.3 はその典型的な例である．認識部は２つに分かれ，１段目の１次認識部が通常の文字認識部である．ここでの認識結果を受けて２段目の２次認識部が動き出す．２次認識部では切替え部が１段目の認識結果に含まれる複数候補の中に類似文字ペアあるいは類

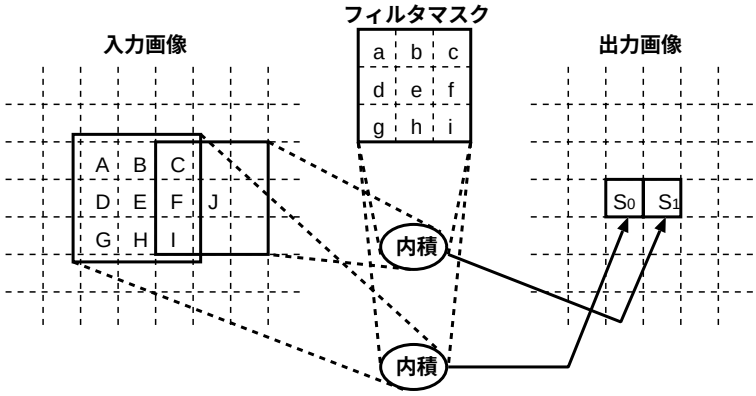


図 1.4: 畳み込み（内積）によるフィルタリングの例

似文字グループがあるかどうかを判定し，あるとすれば対応する「類似文字ペア識別部」を起動する．この図ではＡからＺまでの類似文字ペア識別部があるとしている．

この識別部は１つも起動されないこともあれば複数起動されることもあるが，いずれにしてもそれらの結果が総合判定部に入力される．総合判定部では１次認識部の結果も含めて類似文字ペア識別部の結果を総合的に判定し，最終的な文字認識結果を決定して出力する．

ここであげた認識システムは仮想のもので，本研究ではこの図のような認識システムの構築までは行なっていない．この図の中の類似文字ペア認識部が研究対象であるが，最終的には図のような形式の認識システムの中で部品として使用することを想定している．

1.3.4 学習システムの構成

類似文字識別部の学習には類似文字ペアまたは類似文字グループのみを用い，認識系としてそのペアまたはグループに特化したものを作成する．これを各ペアまたはグループごとに作成し図 1.3 における各「類似文字ペア識別部」とする．

学習手段としては逐次学習系の手法の定番である「評価関数の設定」と「降下法による最小値の探索」という形を採用した．具体的には確率降下法と呼ばれる形式を採用した．

このフィルタマスク学習を本報告では FML（Filter Mask Learning）と呼ぶことにする．

1.4 FML とは何か

1.4.1 フィルタリングによる特徴抽出

本研究で用いるフィルタリングは一般的な畳み込み演算のものである．以下の説明はこの分野では周知の事柄ではあるが一応簡単な例で説明しておく．

図 1.4 はフィルタリングの簡単な例を図示したものである．左側の格子が入力画像で桝目が画素である．画素の値および画素の名称を A〜I と J で表している．一方中央・上の格子はフィルタマスクを表している．この例では 3×3 の画素を持つフィルタマスクである．画素の中に書かれた a〜i がフィルタマスクの画素が持つ値すなわちマスク値である．なおここでも a〜i は画素の値だけでなくその名称も表すものとする．右側の格子が出力画像で桝目が画素である．この画素に記載された S_0, S_1 は入力画像と同じく画素の値および画素の名称である．入力画像の E の画素に対応する出力画素を S_0 とする．画素値 S_0 は以下の式で求める．

$$S_0 = aA + bB + cC + dD + eE + fF + gG + hH + iI.$$

つまりマスクを入力画像に重ねて対応する画素の積を計算し，マスクの画素すべてについて和を取ったものを出力画素値とするのである．画素値の組，$(A, B, \ldots, I)$ と $(a, b, \ldots, i)$ をそれぞれベクトルだと思えばこれは内積演算である．マスクはこのケースでは E の画素位置に適用されたことになるが，これはすべての画素で行なわれなくても良く，例えば n 画素おきの画素に対して実行するようにしても良い．例えば１つおきに実施したとすると出力画像のサイズは辺の長さで言って半分になる．図の例で言えば J の位置に対応する出力画像の画素が S_1 となる．

このマスク値を様々に変えることで多様なフィルタリング機能を実現できる．もっとも分かりやすいのがボカシのフィルタである．例えば，

$$\begin{matrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{matrix}$$

なるマスクはガウス関数に基づくマスクを整数化したもので，このようなマスクを用いたフィルタリングによって入力画像をボカすことができる．

線分抽出フィルタマスク　　エッジ抽出フィル タマスク

図 1.5: 線分検出，エッジ検出を行なうフィルタマスクの例

別の例を図 1.5 に示す．この例はマスクを図示したもので黒が高いマスク値，白が低いマスク値を示す．このようなマスクによって画像の線分検出やエッジ検出ができる．すなわち，このようなマスクと内積を取るとエッジや線分のある所でその値に高いピークや低いピークが現れるのである．

1.4.2 フィルタマスクを学習によって生成する

本研究で用いる競合学習について以下に簡単に説明する．

本研究で用いる学習機構を図 1.6 に示す．まず，この図の点線の中が文字認識システムで，学習終了後にこの部分だけを取り出して文字認識システムとして使用するという想定である．もちろん学習しながら文字認識システムとして使用する使い方もある．

本研究で用いた文字認識部では，フィルタリング演算によって得られた特徴抽出の出力画像はベクトルと見なされ，これは特徴ベクトルと呼ばれる．パターン照合はこの特徴ベクトルと辞書ベクトルを比較することによって行なう．認識結果に基づいて辞書の更新量とフィルタマスクの更新量が計算され，それぞれの更新に使われる．

以下，具体的に数式を使って説明する．入力される学習パターンをベクトルと見なしてこれを z で表し，特徴ベクトルを x とすると，フィルタリングにおける出力１画素分の操作は内積演算なので，特徴抽出全体はこ

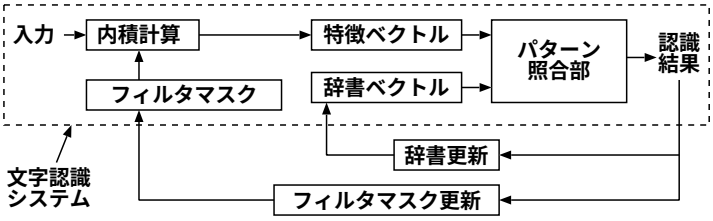


図 1.6: フィルタマスク学習機構の全体構成

れを束ねた線形変換 F で表せる．すなわち $x = Fz$ である．次にパターン照合ではユークリッド距離を用いた相違度計算を行ない，これに基づいた認識を行なう．相違度を S と記述し辞書ベクトルを φ とすると，

$$S = ||x - \varphi||^2, \tag{1.1}$$

となる．もちろん辞書はカテゴリごとに作成されており相違度もカテゴリごとに計算される．この相違度が最小となるカテゴリを答えとして出力するのであるが，ここで正解カテゴリがどれだか分かっているものとし，学習のための更新量を辞書とフィルタマスクについてそれぞれ計算する．

次にその更新量について実際の計算方法を以下に説明する．正解カテゴリに対する最小の相違度（正解カテゴリが複数ある場合も想定している）と，不正解カテゴリに対する最小の相違度を求め，それぞれの相違度を S_{ok}, S_{err} とする．この時 $d = S_{ok} - S_{err}$ は値が大きいほど認識結果が悪いということになる．この d の値を評価値に変換する単調増加関数 $l(d)$ を定義し，その値を入力された学習パターンの評価値とする．この値をすべての学習パターンについて積算したものを最終的な評価値 $J = \sum l(d)$ とする．この J をフィルタマスク F，辞書ベクトル φ の関数と見て，これを降下法によって最適化する．具体的には以下の漸化式が求まる．

$$\Delta\varphi = \pm 2\varepsilon_r \sum l'(d)(x - \varphi). \tag{1.2}$$

$$\Delta F = \mp 2\varepsilon_f \sum l'(d)(x - \varphi)z^T. \tag{1.3}$$

ここで，この式の $\pm$ と $\mp$ は上側が正解カテゴリに対する更新則，下側が不正解カテゴリ中のベストカテゴリに対する更新則である．

辞書ベクトルの初期値については特徴ベクトルの平均を用いた．またフィルタマスクについては，本研究では初期値として幾つかのマスクを用いたが基本的なものはガウス関数型のボカシのフィルタである．

学習用パターンを認識し，誤読かあるいは正解でも相違度の差が小さい場合に上記の更新量を積算していく．これを全ての学習用のパターンに実施して１セットの積算が終ることになる．その結果得られる積算された更新量を F と φ に加算する．すなわちフィルタと辞書を同時に更新する．さらにこの１セットの学習を複数回行って学習を進めていく．こうしてフィルタマスクと辞書ベクトルを更新することで誤読が少なく正解が多くなる方向にフィルタマスクと辞書ベクトルを逐次すこしずつ変更していくことができる．

なお本報告では，以下，更新のための漸化式について，簡略化のために式（1.3）における $\sum$ を省略している場合が多い．このことは方式理解の上で特にさまたげにはならないが一応注意しておく．

1.4.3　非一様マスクの導入

一般に画像処理でフィルタリングを行なう場合，マスクは１種類で画像全域で同じマスクを用いることが多い．これに対して本研究ではマスクを場所ごとに変える仕様を採用している．これは文字パターンの局所的なごく一部分に注目するようなフィルタマスクを学習しようとしているからである．このようなマスクを非一様マスクと呼ぶ．

図 1.7 は一様マスクとこの非一様マスクの例を図示したものである．この図は本研究で用いている学習によって一様マスクが非一様マスクに姿を変える様子を表現したものである．この図の中で四角形の中に黒丸あるいは黒の楕円が１つ置かれているものが１つのフィルタマスクを意味している．楕円の方が線分抽出の機能を持ったマスク，黒丸が方向性の無いボカシのフィルタである．そのマスクが縦横に並んだものは，それら各マスクを文字パターンの中にそれぞれ対応する位置に配置したものである．この場合は 5×5 の配列であり，マスクを並べたものをマスク配列と呼ぶ．左のマスク配列はすべてのマスクが同一で一様マスク配列である．右は学習後のマスク配列で右上部分に線分抽出マスクが生成されていることを示している．この学習後のマスクは非一様マスク配列である．

本研究で行なった学習実験では学習によって徐々に姿を変えていく F と φ に関し，特に F が最終的にどう変化するかを図 1.7 に図示する如く，実際のフィルタマスクをビジュアルに配置して目視でチェックしている．

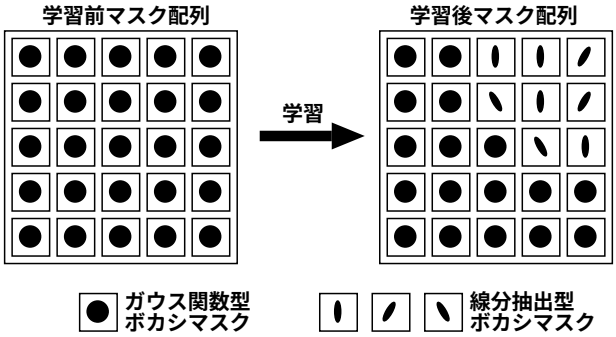


図 1.7: フィルタマスク学習でのマスク配列の変化の例

1.4.4 FML 開発のアプローチ

今後の文字認識をさらに高精度化するためには，まったく新しい方式を提案して従来方式に置き換えるという考え方ではなく，既知の提案に軸足を置いた研究開発方針を取るべきである．何故なら文字認識の世界では過去に様々な方式が数多く提案されていて，しかも実用性に対する吟味が必ずしも十分ではなかったからである．もちろん，まったく新しい方式を提案する事も重要だが，文字認識の長い研究の歴史を考慮すると新規性へのこだわりすぎは問題である．

競合学習に関して言えば，まったく新しい認識方式や学習形式を従来法のリプレースとして文字認識の世界に持ち込むのではなく，できるだけ従来型の文字認識アーキテクチャー，パラメータ，辞書データ，フィルタマスク値などを維持しつつ，標準的な競合学習を導入するというアプローチを最初に取るべきである．

文字認識における従来方式に競合学習を導入する試みは，これまであまり積極的には行なわれてはいなかった．それはそれぞれの新規性が少ないため，それらの組み合わせもまた新規性が無いと考えられていたためであろう．

本研究で用いる方式の学習構造は従来から研究されている様々な先行研究で提案されている形式と似たものである．その理由は上で述べた方針による．本研究では提案が新規かどうかではなく，競合学習の導入によって従来型の文字認識の性能を凌駕できるかどうかの可能性を探ることに軸足を置いている．そのためにはしっかりとした実用性のチェックしなければならない．

また類似文字識別の実用化にあたっては競合学習の導入は必須であり，類似文字識別に適した認識系や学習方式を具体的にどう構築するかが重要となる．そのために十分な評価・実験を実用性の観点で徹底して行なう必要がある．

そのような方針の中で，本研究はまだ基礎的な段階にあるが，将来の実用に耐えるだけの評価を行なうために十分と言えるパターンを確保して実験を行なうことも本研究の重要なポイントとなっている．

以上述べた本研究のアプローチの要点をまとめると次の 2 点になる．

（1）従来型の OCR の構造を維持しながら競合学習を導入
（2）十分なデータを用意した大規模な実験による評価

1.4.5 FML 開発における新提案

前節で述べたように本研究で用いる方式のその形式については新規性は少ないが，実験として行なっていることやその結果は新しいものであり，それが本研究の新規性である．ここではそのことについて述べる．

ビジュアライゼーションの重要性

本実験で最も重視した点はフィルタマスクのビジュアライゼーションである．こうしたことはこれまでの研究ではあまりなされていない．OCR の設計という立場では，これを視覚的にチェックすることは非常に重要であり，基礎研究としてもこれを視覚的に見られるようにすることは学習の実態やフィルタによる特徴抽出の動きを理解する上で重要である．

フィルタマスクの非一様性

非一様なフィルタマスクの導入については既に説明した通りであるが，これは一様なフィルタマスクが普通であった文字認識分野において，特に類似文字識別を開発する意味で重要な提案である．

高性能化のための改良と学習パラメータの設定法

FML の改良また学習パラメータの設定に関して以下の提案を行なった．
（1）ガボールフィルタの FML（これを一般化したものを多重化と呼ぶ）
（2）学習強度を決める方式の検討

「仕様なき学習」を目指して

前にも述べたように本研究の究極の目的は「仕様なき学習」にあり，その意味で，どのようなマスクが生成されるかはあらかじめ見込みを持たないで実験を行なっている．意図した結果では無く結果としてそうなったという実験結果である．普通の研究ではこうしたアプローチは良くないとされるが本研究ではあえてそうしたアプローチを取り，よりロバストな学習の実現を試みている．

これはある意味で生物学的な実験や観察に比較することのできるアプローチである．人に近付こうとすれば，勢いそういうことになるのかも知れない．

1.4.6 研究の位置付けと目的

ここでは本研究に関連する先行研究について述べる．本研究は個別文字認識をターゲットとしており，その中で特に FML を新たに類似文字識別に導入してその有効性を確かめた．研究は個別文字認識全体に及ぶが，先行研究の調査に関しては主要な研究対象であるこの FML について報告する．

まず文字認識に関連して類似文字識別の分野について述べ，次に本研究のターゲットである特徴抽出学習について説明する．続いて関連する学習方式として LVQ（Learning Vector Quantization）[2]，確率降下法，ニューラルネットに関し本研究との関係を説明していく．

類似文字識別の先行研究

典型的な類似文字識別は文字ペアごとにその相違点をチェックする方法で，例えば「犬」と「大」であれば右上に点があるかどうかを画像処理的にチェックすることで識別を行なうような手法である [40]．このような方式を学習によって自動的に作り出すということは難しく，実際には各ペアごとに識別プログラムを人手によって作成しなければならない．これがこのアプローチのアキレス腱であり実用上の難問であった．

とは言うものの，この種の手法は副作用が少なくまた確実に性能を上げられるので，現実に人海戦術が可能な場合にはしばしば OCR の設計現場で使われている手法でもある．しかしながら，このような手法はアドホックであるために学術的でないと見なされるためか学会発表の例は非常に少なくまた注目もされていない.「やればできるでしょ」と思われてしまうアカデミズムと現場の苦労とが相容れない部分である．

アカデミズムという立場では自動的に類似文字ペアのための識別辞書を生成するようなアプローチが好まれる．混合類似度法 [24] はそうした手法の１つでありその提案後にも様々な改良が提案されている．この種の方式では類似文字ペア間の相違部分を強調するような認識辞書を生成し，相違度もそうなるように学習を設計する．しかしながら特徴抽出部やパターン照合部はどの文字ペアに対しても同じものを使うように構成するのが普通で，その点が前で述べたアドホックな方式よりも扱いやすい利点であるのと同時にこの手法の限界を示すものとなっている．つまり文字ペア独自の特徴を反映したような特徴抽出やパターン照合は設定されないのである．その意味ではまだ改善の余地を残した方式とも言える．

本研究は認識辞書だけでなく特徴抽出部にも特定の類似文字ペア用の自動学習を導入するものであり，従来手法に比べて類似文字識別の一層の性能向上を狙うものである．

特徴抽出学習の先行研究

認識辞書の学習というテーマに比べると，これまでに特徴抽出の学習方式はあまり提案されていない．とは言うものの幾つかの先行研究 [13][14][15] の他，音声認識の分野では DFE（Discriminative Feature Extraction）と呼ばれる方式 [4],[5],[6] が知られており，これらは本研究に対する先行研究となっている．

これらの方式はいずれもフィルタを自動生成する学習方式であり，大きく分けて次の２つのタイプになる．

・１つは，フィルタを構成するための各種パラメータ値を学習によって獲得するもの．

・もう１つはフィルタに用いられるフィルタマスクの値そのものを学習する方式．

ほとんどの研究例は前者のタイプで後者のタイプは少ない．その理由は後者の方式では学習の規模が大きくなるにもかかわらず構成が単純で，しかもニューラルネットワーク（NN:Neural Network）に近い構造なので新規性が少ないことにあると思われる．学術的に手が付きにくいテーマなのである．フィルタマスク学習はもちろん後者の方式であり，新しい提案では無いのであるが文字認識での研究例は多くなく研究余地のまだ残されている方式である．

LVQ と確率降下法に関連する先行研究

本研究で用いる学習手段は MCE/GPD，すなわち MCE 基準（Minimum Classification Error Criterion）と一般化確率降下法（GPD:Generalized Probabilistic Descent）[3] に基づく学習方式を特徴抽出に適用した DFE の枠組でも説明できる．しかしながらこれまでの DFE の研究では学習対象をフィルタマスクとしない事が多い．

この MCE/GPD は甘利 [1] によって提案された確率降下法（PD:Probabilistic Descent）を改良したものだが，この PD をフィルタマスク学習に適用したものが FML だと考えることもできる．しかし，このアプローチで導出される形式は NN に近く新規性が少ない．このことが FML がこの分野でも敬遠されている理由なのであろう．なお，この PD では MCE 基準に対応するものは平均リスク，AR（Average Risk）と呼ばれているので，MCE/GPD の表記になぞらえればこれは AR/PD と表記できる．

一方 LVQ は音声認識をターゲットに提案された方式で，その後も音声以外に文字や画像などパターン認識一般に広く使われるようになっているもので，今では競合学習の１つのスタンダードとなっている手法である．この方式はしばしば NN の仲間として取り扱われる事も多い．しかしながら，この方式は PD の提案の後で提案されたものであり文献 [1] の中にはこの LVQ と同じ方式が PD の１例として記述されている．その観点では LVQ も AR/PD の仲間と言って良い．

さて，本研究で扱う FML はフィルタマスクの学習にこの PD を用いたと考えるのが最も自然で，その意味では FML を LVQ の拡張，すなわちフィルタマスクの学習を LVQ を拡張して行なったと解釈することもできる．その方向から見れば本研究は LVQ や PD の応用研究の１つとなる．

ニューラルネットに関連する先行研究

本研究で取り組む FML は見方を変えると NN と同じ構造になる．その意味で考えると FML には膨大な先行研究があることになる．これも FML が過去にあまり研究対象にならなかった理由の１つであろう．提案しても NN の分野から見れば当り前の技術と見なされるからである．

FML は NN やネオコグニトロン [12] と近い関係にある．これらの膨大な分野の中には NN の構造をフィルタと見なして，その立場で研究した事例も知られている [11].

また文字認識をターゲットにした NN の研究もあって，特に海外では実用化に成功したものも多いと言われている．しかしながら国内ではそれ程ではなく，NN の OCR での成功事例はほとんどない．国内の多くの事例では従来型の技術の改良によって高性能 OCR を実現している．また国内での NN による OCR の高性能化の事例について見れば，それらは NN と言いながらも LVQ や PD，DFE であることが多く，しかも最終的には製品にまで至っていないケースも多い．

海外での OCR 事例として NN が多く聞かれるのは何故であろうか．実は NN のアドバンテージは，実績の無いメーカでもある程度容易に性能を出せる OCR を開発できる所にある．有力メーカによる研究開発の継続性が少ない海外において，おもにベンチャー企業で NN を用いて成功した事例が多いのはこのためと考えられる．

国内では数多くの経験と実績の裏付けのもとで OCR をしっかり作り込んでおり，その性能というものは侮りがたい．地味な開発の蓄積だが，その認識精度は相当なもので NN といえども簡単に抜き去ることができない．前にも述べたように実際にある OCR の実用化事例で，NN 型に比べて従来型の OCR 方が誤読が１桁少ないケースもあった．国内では有力メーカによる OCR 開発は継続的に行なわれて来ており，国内で OCR の NN 応用事例が少ないのはこのためであると考えられる．

本研究との関係

これらの従来研究では特徴抽出部と認識部を合わせて１つのシステムと考えるものが多く，また特徴抽出部の学習自体を分析しようとするものは多くなかった．それ故，本研究で行なったような研究の事例は少なく実用レベルの研究開発という意味ではこれまでの研究だけでは不十分であった．

実用研究では個々のターゲットごとに実験データに基づいた学習プロセスの分析が必要で，本研究はその１つとして手書文字の類似文字識別をターゲットにそのような分析を行なったのである．なお文字以外の分野でも本研究で扱うようなフィルタマスクの学習に関する研究は多くなく，特に学習によってどのようなフィルタが生成されるかについての調査はあまりなされてはいない．

本研究で用いる手法は LVQ の枠組，DFE の枠組，NN の枠組，またその他の学習の枠組と比較して特にそれらと大きく異るシステムでは無い．むしろこれらの枠組の中では標準的な仕様の方式を対象としている．問題はパターン認識における類似パターン識別，とりわけ類似文字識別にけるフィルタマスクの学習が，学習しない方式と比べてどういう有効性があるのかということであり，またその学習がどう効果的に行なわれているかということである．

本研究の目的はこのような FML における学習状況の分析にある．そのための目標の１つは FML によってどの様なフィルタが生成されるのかを実際に目視で観察できるようにすることであり，その観察を通して有効性を評価する事である．もちろん FML で達成される認識精度について確認することも目標の１つである．

1.5 論文の構成

本章では本研究の概要について述べた．以降，第 2 章で本研究の記述内容を理解するための準備として文字認識，特徴抽出，フィルタリング，競合学習などについて簡単な説明を行なっている．また文字認識高精度化のための構造解析的パターン認識と統計的パターン認識の新提案についてもこの第 2 章の中で述べている．次に第 3 章では FML 導入の準備として特徴抽出のためのフィルタリング技術と画像処理におけるフィルタリングの役割について述べている．続いて第 4 章で FML の詳細についてその理論的な説明を行ない，第 5 章で具体的に

FML をどう組み上げていくかということについて詳細に説明している．第 6 章では実験報告として本研究の中で行なった実験について詳しく報告している．最後の第 7 章がまとめである．また付随する事項については付録でその詳細を記述した．

第2章　準備（文字認識高精度化とFML）

2.1　文字認識をどう高精度化するか

2.1.1　個別文字認識の高精度化

高精度文字認識を実現するためには様々な技術が必要である．大きく分けて，画像処理，レイアウト解析，行・文字切出，個別文字認識，後処理（知識処理）がその要素技術である．本研究でターゲットとするのは，その中でも最重要部と考えて良い個別文字認識の部分である．

この個別文字認識には２通りの方式，すなわち構造解析的パターン認識と統計的パターン認識があり，それぞれ一長一短がある．このぞれぞれの方式を強化して互いに補完するように用いることで文字認識の高精度化を狙えるようになる．

またこの２方式に加え，類似文字に特化した文字識別部を別途開発し，これを付け加えた．この類似文字識別は高精度化を狙う意味では重要な考え方であり，本研究では特にこの類似文字識別部分に FML を導入した．本報告ではおもにこの FML の説明を中心に行なうが，その前にまず本章では基礎となる構造解析的パターン認識と統計的パターン認識について，文字認識高精度化に向けて行なった開発の概要を述べる．

2.1.2　構造解析的パターン認識と統計的パターン認識の相補性

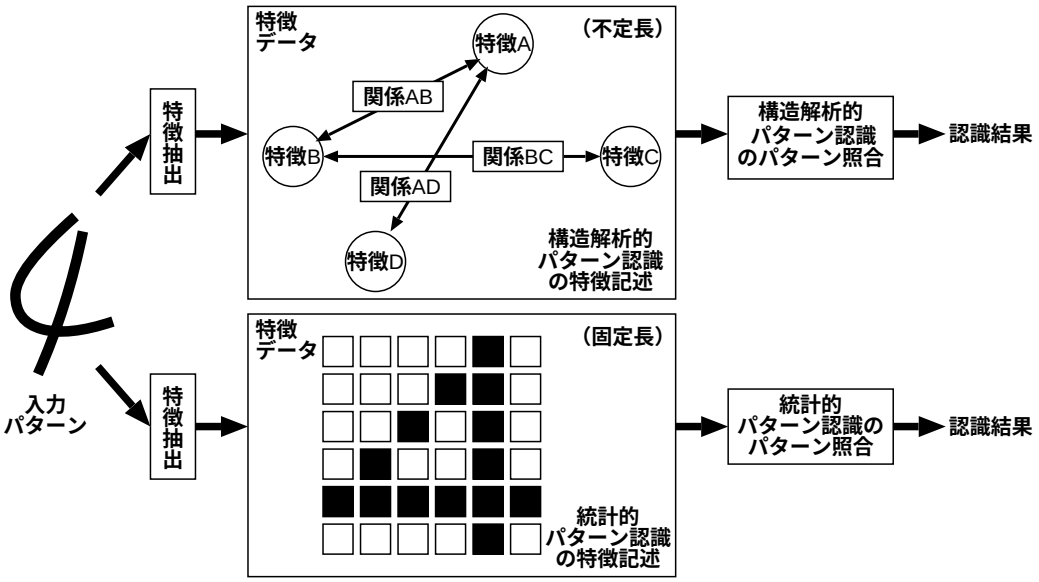


図 2.1: 構造解析的パターン認識と統計的パターン認識の比較

構造解析的パターン認識と統計的パターン認識の特徴抽出方式はこの２つでかなり異なり，それを説明するために図 2.1 に例を示す．

まず構造解析的パターン認識とは，パターンの形状や構造を複数のプリミティブとそのプリミティブ同士の関係によって表現し，辞書として記述されている同様な表現と比較することによって認識を行なう方式である．特徴データはこのプリミティブとそのプリミティブ同士の関係の記述そのものであるが，そのデータ長は不定長

であることが多い．図 2.1 の上側は構造解析的パターン認識における特徴データの例を示すものである．入力された文字「４」の端点や折れ曲がり点（特徴Ａ〜特徴Ｄ）をプリミティブとし，文字を構成する線の中にこれらのプリミティブ間を結ぶものがある場合に，プリミティブ同士の関係の記述として「関係〜」を記載している．

一方の統計的パターン認識における特徴データは固定長である．図 2.1 の下側がその例である．この図の 6×6 のタイルの並びは，白タイルが白画素，黒タイルが黒画素を意味する．入力された文字「４」からこのようなタイル配列が得られたとすれば，各タイルに 0,1 を割り振ることで固定長のデータが得られる．このような固定長のデータは特徴ベクトルと呼ばれる．

統計的パターン認識では，この固定長の特徴ベクトルが存在する線形空間の中でパターン分類を考える．線形空間の中で境界面として超平面（曲面）を決定し，その境界面のどちら側に特徴ベクトルがあるかによって分類を実施する．この境界面の決定には認識系を設計するために用いるパターン，すなわち学習パターンに関し，その分布の統計的情報が利用される．

なお実際の認識においては類似度や相違度計算という形をとって，この境界に基づく入力パターンの分類が実行される．要するに入力されたパターンから作成される特徴ベクトルが，この境界面のどちら側にどれだけ離れて存在しているかによって認識を行なうのである．

構造解析的パターン認識と統計的パターン認識における特徴抽出プロセスの重要な違いは，データ長が不定長か固定長かにある．さらに加えて重要な事実はそれらの特徴データ表現に数学的表現で言うところの「位相」が簡単に導入できるかどうかにある．分かりやすく言えば特徴データ間に簡単に距離の概念が導入できるかどうかである．

統計的パターン認識における特徴ベクトルは線形であり距離の概念が導入されている．一方，構造解析的パターン認識でも結局は辞書と特徴データの間で類似度や相違度が計算されるので，一応，距離あるいは位相に相当するの概念がある．しかしながら辞書と特徴データの形式が異なったり，データ長が不定だったり，またデータ要素が様々でそれぞれのメトリックが異なるなど，それらは簡単な数学で扱えるものでは無い．

もちろん構造解析的パターン認識でも特徴ベクトルを扱うような方式も有り得りえて，上記の説明はその点では厳密では無い．ここでの説明は単に一般的な両方式の特徴付けを述べたもので厳密な定義ではない．

なお構造解析的パターン認識でも統計的パターン認識でもその構成は「特徴抽出部」＋「パターン照合部」であり同一の構造である．この両者で特徴抽出部が異なることは前述の通りであるが，パターン照合部も大きく異なっていることは言うまでもない．

この両者は相補的な面を持っており，それぞれ得意とする文字変形が異なる．前にも述べたように構造解析的パターン認識は大きな変形に強く，統計的パターン認識は多様な変形に強い．これらを相補的に組み合わせて用いることが文字認識高精度化の開発にはかかせないコンセプトとなっている．本研究ではこの両方の方式についてそれぞれ新方式の開発を行なった．

2.2 輪郭構造マッチング法

2.2.1 輪郭構造マッチング法における特徴抽出とパターン照合

構造解析的パターン認識にも様々な方式があるが，文字認識の高精度化に寄与する高度な構造解析的パターン認識の例として，ここでは本研究の一環として行なった輪郭構造マッチング法 [40] について説明する．輪郭構造マッチング法は自由手書文字認識を目的に開発されたもので，特に文字変形の大きい場合の認識高精度化に貢献する重要な方式である．

この方式では入力パターンはまず輪郭追跡され，これによって作られる輪郭線が認識対象となる．図 2.2 はこの方式の処理の様子を図示したもので，この図の太線部分が輪郭追跡によって取り出された輪郭である．この輪郭線は短い輪郭線分に分割される．なお，この短い輪郭線分をここではセグメントと呼ぶ．図では各セグメントを区別するために数字，アルファベットを付けて表示している．

次に，この各セグメントごとにその長さ、曲率、位置、方向等の属性情報を画像処理プロセスによって計算し求めておく．図 2.2 の左側が入力パターンの輪郭データ，右側が認識のための辞書データである．この辞書デー

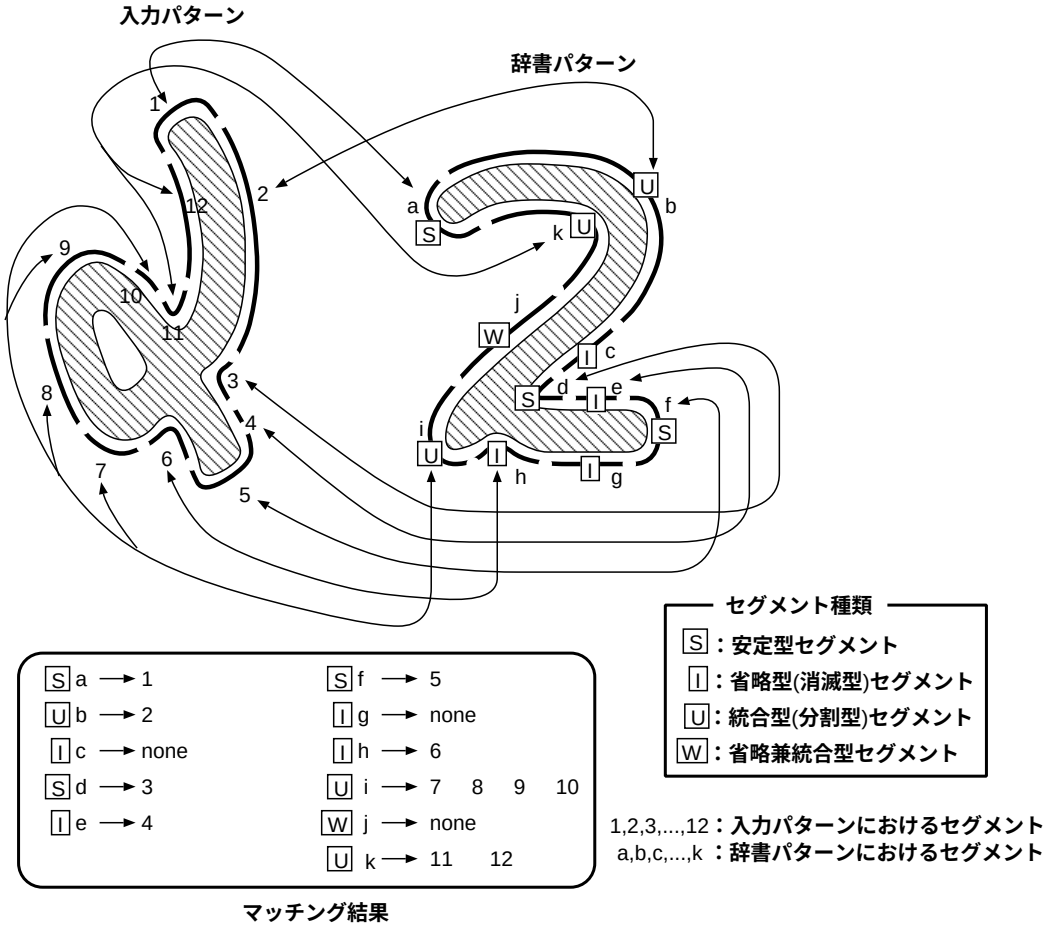


図 2.2: 輪郭構造マッチング法（構造解析的パターン認識の例）

タも入力パターンと同様に輪郭セグメントのシーケンスで記述されている．これらには前記の属性情報に関して，その値の上限と下限の値がセットされている．これらの属性情報を比較することによって，入力パターンのセグメントと辞書データのセグメントとの対応付けを行なう．辞書の各セグメントはＳ，Ｉ，Ｕ，Ｗの４種類に分けられている．

* Ｓ：安定型セグメント
* Ｉ：省略型（消滅型）セグメント
* Ｕ：統合型（分割型）セグメント
* Ｗ：省略兼統合型セグメント

辞書パターンと入力パターンのセグメントの対応関係についていえば，Ｓは１対１対応，Ｉは１対１または０対応，Ｕは１対１または多対応，Ｗは１対０，１，多対応である．これらの意味は次の通りである．Ｓは安定していて，必ず入力側にそのセグメントに対応するセグメントが１個存在することを意味している．一方，Ｉは１個存在するかまたは対応なしでも良いとするものである．Ｕは入力側の複数のセグメントが１個のＵに対応して良いというものである．ただしＵでは省略すなわち対応なしのケースは不可で，これが省略可のものがＷである．このように辞書セグメントを４種類に分けることによって高速に効果的なマッチングを行なうことが可能となる．図 2.2 の例では，最初にセグメントの位置の情報を用いて開始セグメントを決める．繁雑になるので開始セグメントを決定する手段についてはここでは説明を略するが，この開始セグメントを決定した後，すなわちこの例では入力パターンの１と辞書パターンのａが開始セグメントとして対応付けられた後，以下，図の下の囲みに示したようにｂと２の対応付けに続いて残りの対応付けが行なわれていく．最後まで対応付けが完了すると入力パターンはその辞書パターンのカテゴリであったと判断される．異なるカテゴリの場合には途中で対応付けが

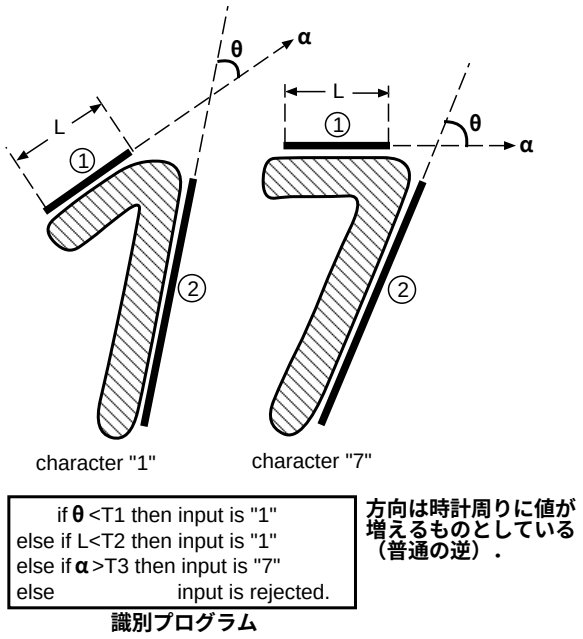


図 2.3: 輪郭構造マッチング法における類似文字識別の例

失敗するように設計されている．認識対象の文字に大きな文字変形がある場合も，この構造解析的手法によれば，それを辞書の記述として吸収することができ，高精度認識が可能となる．

2.2.2 輪郭構造マッチング法における類似文字識別法

輪郭構造マッチング法では，マッチング後に類似文字識別を詳細に行なうペアチェックと呼ばれる部分が起動される．これがこの方式の識別能力を格段に高めている部分であり付属的な機能とは言え重要な要素である．以下，図 2.3 を参考にその動作を説明する．

図 2.3 は数字「１」と「７」を区別する為の方法の例を図示したものであるが，これは例えば「０」と「６」を区別する方法とは全く異なる．ここでの問題はどのような時にどのような方法でこの詳細識別を行なうかということである．この方式で採用されている手段は次のようなものである．

上で述べたＳＵＩＷの照合を行なった結果，照合が成功した辞書パターンが何であったか，それが詳細識別を必要としているかどうか，また必要だとしてどのような種類の詳細識別が必要なのかが判定される．

このＳＵＩＷの照合で，辞書セグメントと入力セグメントの対応付けができるので，これを手掛かりに入力パターンの特定部分を指定することができる．詳細識別は辞書に記述されているが，このような特定部分の指定によって，その部分について何をどうチェックするかの記述がなされる．

図 2.3 にその識別プログラムの例が記述されている．入力パターンからＳＵＩＷの照合によって見付かる２本の棒部分について，それらの長さ，方向，なす角度を求め，それをある閾値と比較することによって「１」か「７」かを判定ししている．

要するに輪郭構造マッチング法では先行するＳＵＩＷの照合結果を用いることで，類似文字識別の段階で２本の棒の長さ，方向，なす角度を求める部分をほぼ省略している．一般にこの例に示すような類似文字識別部を個々のペアごとに手作業で作成するのは非常に手間のかかる作業であり，本方式はその作業の手間を大幅に削減しようとするものである．実際に実用レベルでのペアチェック部開発を行なってその有効性を確認している．

2.3 パターンの球面分布に基づく部分空間法の高精度化

2.3.1 超球面上のパターン分布

統計的パターン認識にも数多くの手法が存在する．部分空間法はこの分野の代表的な方式 [25] であり，これまで各種の問題に適用されてきた．一方，疑似ベイズは三重大から提案された手法 [28] でベイズ識別における分布関数のパラメータ推定においてノイズの影響を考慮することによって理論的に得られるもの [29] である．文献 [29] では，これは MQDF（Modified Quadratic Discriminant Functions） と呼ばれている．これも実用性の高いパターン認識の有力方式である．なお疑似ベイズの系統の１つである疑似マハラノビスと同じものは１９７０年に超楕円形識別関数 [26],[27] として葛城が提案している．

この部分空間法と疑似ベイズとを比較すると疑似ベイズの方が良い認識精度を示すことが知られている [39]．一方で文字認識システムの構成上，部分空間法を用いたいという要請もあり，部分空間法の性能を疑似ベイズと同様に上げられないかという課題があった．単純に考えれば疑似ベイズをそのまま使用すれば良いのであるが，実際の OCR 設計においては認識システムの仕様，ハードウェアの仕様，特徴データの種類などによって部分空間法を使用せざるを得ない事がありこのような問題が出て来るのである．

この問題に答えるべく提案したものが本研究の一環として開発した「超球面上のパターン分布」に基づく方式 [39] である．これによって部分空間法でも疑似ベイズと同等の性能のものが実現できるようになり，部分空間法の系統でも多様な変形に対する認識精度をさらに高めることができるようになった．

この考え方はパターンが超球面上に分布していると仮定してパターン認識方式を導くもので，単に部分空間法の性能を向上させるという事だけでなく，疑似ベイズと部分空間法の関係を分かりやすく考察することが可能となるコンセプトでもある．これを言葉を替えて言えば部分空間法の理論的バックグラウンドをパターン分布モデルにまで遡って探る試みと言える．疑似ベイズと部分空間法との関係は文献 [30],[31],[32] に述べられているがパターン分布モデルにまでさかのぼった関係は調べられていない．

「超球面上のパターン分布」に基づく方式では，超球面上でガウス型の分布となるある種のパターン分布を提案している [46][47][39]．円周上や球面上の分布については文献 [33] に詳しいが，本提案で用いられている分布は円周上の Dimorth-Watson 分布，球面上の Bingham 分布 と同じ形式の点対称（本報告では反対側は０にしている）の Axial 分布である．なお，このような点対称な分布と部分空間法の関係については文献 [30] で示唆されている．

パターン認識の理論解析の立場から言えば，本来その研究はパターン分布の性格を明らかにしてそのモデルを構築し，その上でパターン認識理論を展開すべきであるが本論文の立場はそれとは異り実用性に重点を置いたものである．実際に使われている実用的な種々の方式における理論的な関係を明らかにして整理することがその目的の１つである．実用化にあたっての各種実験データの分析に役立てるための研究という側面もある．その観点から部分空間法との数式的な関連が強い上記分布を分布モデルとして採用したのである．

図 2.4 はこの分布モデルを説明する図である．まず空間上のパターンを x とする．このパターン分布では $||x|| = 1$ が前提であり，その意味で x は N 次元空間内の原点を中心とした超球面 Q 上にある．図ではこれを円周 Q 上にあるとして表示している．もちろんこの Q は実際には超球面である．次に，あるベクトル m ，$||m|| = 1$ について， m に垂直な $N-1$ 次元の超平面 P を設定する．この m も超球面 Q 上にあり， P は図では右側矢印の直線で表している．P 内の $N-1$ 個の正規直交ベクトルを $\{\varphi_i\}$ とし，P 内の任意のベクトル $t(||t|| \leq 1)$ について $g(t)$ を次のように定義する．

$$g(t) = \frac{1}{A}\sqrt{1-||t||^2}\, e^{-\frac{1}{2}\sum_{i=1}^{N-1}(t,\varphi_i)^2/\lambda_i}. \tag{2.1}$$

ここで， $||t|| > 1$ では $g(t) = 0$ とする．さらに次の定数を定義する．

$$A = \int_{P,||t||\leq 1} \sqrt{1-||t||^2}\, e^{-\frac{1}{2}\sum_{i=1}^{N-1}(t,\varphi_i)^2/\lambda_i} dt. \tag{2.2}$$

この式はガウス分布（図の細い線）における次の式に対応している．

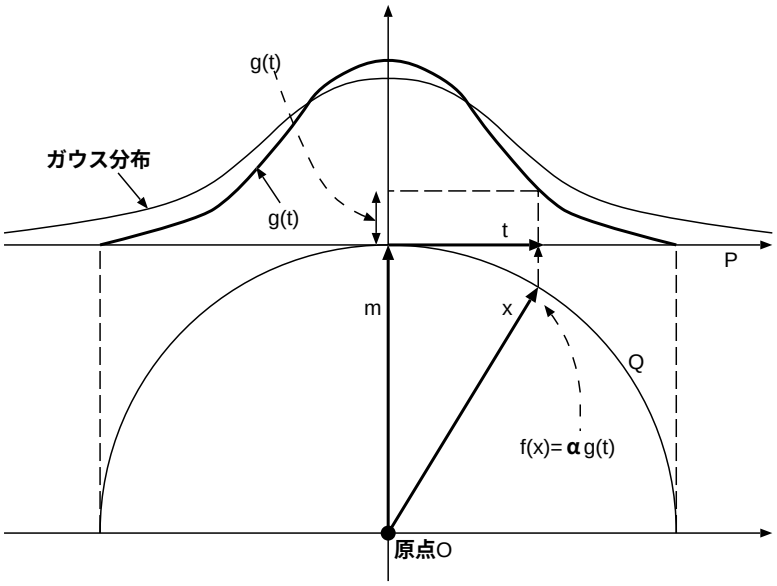


図 2.4: 球面上のパターン分布

$$\frac{1}{B}e^{-\frac{1}{2}\sum_{i=1}^{N-1}(t,\varphi_i)^2/\lambda_i}. \tag{2.3}$$

この式の B は,

$$B=\int_P e^{-\frac{1}{2}\sum_{i=1}^{N-1}(t,\varphi_i)^2/\lambda_i}dt. \tag{2.4}$$

によって求まる．もちろんガウス分布の原点すなわち $t=0$ の時の値は $1/B$ である．

次に点 x からそれに対応する分布関数の値を決定するために x から P 上に垂線をおろし，その点における $g(t)$ の値を求め，それを点 x における分布関数の値とする．すなわち $\mu=(x,m)$ と置いて,

$$t=x-(x,m)m=x-\mu m, \tag{2.5}$$

とし，Q 上で関数 $f(x)$ を次のように定義する．α を正規化するための定数として,

$$f(x)=\alpha g(t)=\alpha\frac{1}{A}\sqrt{1-||t||^2}\,e^{-\frac{1}{2}\sum_{i=1}^{N-1}(t,\varphi_i)^2/\lambda_i}. \tag{2.6}$$

ここで， $\mu>0$ の時には,

$$||t+\mu m||^2=1,\quad ||t||^2+\mu^2=1,\quad \mu=\sqrt{1-||t||^2}. \tag{2.7}$$

であることに注意しておく． $\mu<0$ の時，すなわち $(x,m)<0$ では $f(x)=0$ ，すなわち下半球では $f(x)=0$ とする．なお図 2.4 では $t=0$ の時の $g(0)=1/A$ がガウス分布（細い線）の値 $1/B$ よりも大きい事を示している．

単純に定義するのであればパターン分布を $P(x)=f(x)$ によって定義すれば良いが，球面上での積分を分かりやすくするために，パターン分布は球面に沿って薄い厚みを持っていると仮定する．すなわちパターンの球面上の分布に十分小さな幅 d を持たせ，球面上で分布の積分を行なう時はこの厚みで分布の体積積分を行ない，その後に $d\to 0$ として球面上での積分値を求める．したがってその薄い幅 d の中では，原点から点 x 上に引いた直線上の分布関数の値は x 上の値と等しいとし，また幅 d の外側では $P(x)=0$ とする．

ここで問題になるのはこのような分布を仮定した時に，分布関数のパラメータ λ_i,φ_i を推定する方法である．このパラメータは認識系を構成した時の辞書に相当する．結論から言えば，これは通常のガウス分布と同様の手

表 2.1: 疑似ベイズ系の方式と部分空間法系の方式の比較

疑似ベイズ系	部分空間法系
共分散行列 $K = \sum_\alpha (x_\alpha - m)(x_\alpha - m)^T$	相関行列 $K = \sum_\alpha x_\alpha x_\alpha^T$
分布モデル=ガウス分布	分布モデル=球面ガウス分布
↓ 誤読率最小化	
ベイズ識別 $D = \sum_{i=0}^{N-1}(x-m,\varphi_i)^2/\lambda_i + \sum_{i=0}^{N-1}\ln\lambda_i$	球面ベイズ識別 $D = -2\ln(x,\varphi_0) + \sum_{i=1}^{N-1}(x,\varphi_i)^2/\lambda_i + \sum_{i=1}^{N-1}\ln\lambda_i$
↓ 小さい固有値をノイズ成分として，k 番目以降の λ をある値 δ に固定	
疑似ベイズ $D = \|\|x-m\|\|^2 - \sum_{i=0}^{k-1}(1-\frac{\delta}{\lambda_i})(x-m,\varphi_i)^2$ $+\delta\sum_{i=0}^{k-1}\ln\lambda_i + \delta\sum_{i=k}^{N-1}\ln\delta$	疑似球面ベイズ $S = 2\delta\ln(x,\varphi_0) + (x,\varphi_0)^2 + \sum_{i=1}^{k-1}(1-\frac{\delta}{\lambda_i})(x,\varphi_i)^2$ $-\delta\sum_{i=1}^{k-1}\ln\lambda_i - \delta\sum_{i=k}^{N-1}\ln\delta$
↓ $\ln\lambda$ の項を無視	
疑似マハラノビス距離 $D = \|\|x-m\|\|^2 - \sum_{i=0}^{k-1}(1-\frac{\delta}{\lambda_i})(x-m,\varphi_i)^2$	疑似球面マハラノビス距離 $S = 2\delta\ln(x,\varphi_0) + (x,\varphi_0)^2 + \sum_{i=1}^{k-1}(1-\frac{\delta}{\lambda_i})(x,\varphi_i)^2$
↓ $\lambda_i \gg \delta$ を仮定	
投影距離 $D = \|\|x-m\|\|^2 - \sum_{i=0}^{k-1}(x-m,\varphi_i)^2$	部分空間法 $S = \sum_{i=0}^{k-1}(x,\varphi_i)^2$

法で求めることができる [39]．すなわち分布の平均ベクトルを $m' = \int xP(x)dx$ ，相関行列を $H = \int xx^T P(x)dx$ として以下のような結果が得られる．

（1）$\alpha = A/B$.

（2）$||m'|| = A/B$.

（3）λ_i, φ_i は H の固有値，固有ベクトル

（4）m も H の固有ベクトル

この結果，パターン分布から計算される H の固有値，固有ベクトルから認識系を構成できる．

2.3.2 部分空間法と疑似ベイズとの比較

前節で述べたパターン分布モデルの仮定により部分空間法，正確には疑似ベイズに似たある種の重み付き部分空間法が，最適認識方式として導かれる．重み付き部分空間法とは内積の自乗項に重みを掛けて和を取るものであり，部分空間法はこの重みを 1 としたものである．

疑似ベイズ系統の方式と部分空間法系統の方式をそれぞれ対応付けて表 2.1 に示す．

この表で N をパターン空間の次元，学習データを x_α ， その平均ベクトルを m ，φ_i を共分散行列または相関行列の固有ベクトル，λ_i をそれに対応する固有値とする．D は相違度，S は類似度である．

疑似ベイズ系統，すなわちガウス分布型では x は正規化されない．部分空間法系統では x は正規化されている．ただし x が正規化されていないケースでは x のかわりに $x/||x||$ とすれば良い．

これまでは，この表の右側部分については部分空間法しか知られていなかった．これに対して球面ガウス分布を導入することによってこの表のすべて埋めることができるのである．

この表において理論的に最初に求まる識別関数がそれぞれベイズ，球面ベイズであり「誤読率最小化」と記述した段の下に示した．これに対して小さい固有値をノイズ成分の影響としてある値 δ に固定したものが疑似（球面）ベイズでその次の段に示した．さらに固有値が各カテゴリで共通の値を取ると考えて式の一部を無視した形式が疑似（球面）マハラノビスとなり，それをその次の段に示した．さらに $\lambda_i \gg \delta$ と仮定して，各内積の

自乗の項に付いている重みを１にしたものが最終段に示した投影距離と部分空間法である．

認識実験を行なってみると，その認識精度の結果についてはこの図の縦方向の差が大きいのに対して，横方向の差は少なく，その意味でガウス分布型と球面ガウス分布型の両者が良く対応付いていることが分かる．球面疑似ベイズの性能は疑似ベイズのそれと良く似ており，その結果，部分空間法系の方式でも疑似ベイズの水準の性能が達成できることが確認できた．

球面ガウス分布では入力ベクトルのノルムは１でなければならないが，これはある特定のパターンでは無意味なことではない．もっとも典型的な例は特徴ベクトルをボカシたパターンやパターンそのものとした時で，パターンが平行移動の変形やそれに近い変形を受ける場合はそのノルムはほぼ同じ値となる．

2.4 類似文字識別を高精度化する FML

2.4.1 FML の導入

高精度文字認識の実現にあたって，ここまでは構造解析的パターン認識と統計的パターン認識のそれぞれについて新提案の内容を述べて来た．本節以降では，この構造解析的パターン認識と統計的パターン認識の研究開発に引き続いて行なった類似文字識別の高精度化の試みについて述べる．この課題に対しては先に述べたように FML を提案・導入し実験でその有効性を確認した．また本報告では特にこの FML にフォーカスしていることもすでに述べた通りである．次章以降この FML について実験も含めて詳しく述べる．

次章以降の FML の説明の前に，次の節でこの FML に学習を導入した背景として，構造解析的パターン認識と競合学習の関係について述べておく．

2.4.2 FML における競合学習導入の背景

FML における特徴抽出は統計的パターン認識のタイプである．その理由は統計的パターン認識での特徴データが固定長でベクトルだからである．そこでは線形空間の数学が使えるために競合学習の導入が容易なのである．

一方，輪郭構造マッチングで取り組んだ構造解析的パターン認識でも競合学習は重要である．輪郭構造マッチングの説明の中で述べたように，その開発においては類似文字識別機能の作成は手作業であった．輪郭構造マッチングではその作業を軽減するための仕組みがが導入され，それによって高性能文字認識が実現できたのであるが，残念ながらこの方式でさえ類似文字識別部の開発にはかなりの手間がかかっていた．さらなる高精度化を狙う意味ではこのような構造解析的パターン認識に対しても競合学習の導入が必須である．それが無い事は，近年の学会や業界において構造解析的パターン認識の研究が停滞している１つの理由だと思われる．

しかしながら，この構造解析的パターン認識の分野は数学的にスッキリしていないことが多く，競合学習の枠組を導入することは大変に難しい．これまでにも幾つかの研究例 [23] はあるが，その数は多くなく実用レベルのものは無いと言って良い．また構造解析的パターン認識に競合学習を導入する課題には，パターン照合部の学習と類似文字識別部の学習の２つの課題があるが，実用レベルでは両方必要であると考えられ，それはこの問題を一層難しい研究課題にしている．

実は FML はかなりの困難が予想されるこの構造解析的パターン認識＋競合学習の課題に対し，その前に取り組むべき特徴抽出＋逐次学習として取り上げたものである．まず難しい課題へ向かうための第１歩として考えられた研究テーマである．その意味では FML 開発の契機は輪郭構造マッチング法のペアチェック部の中にある．

第3章　フィルタマスクの基礎検討

FML ではフィルタリングを特徴抽出として用いているが，このフィルタリングは画像処理の基本技術の１つでもある．これは画像処理や画像理解はちろん文字認識でも重要な役割を果たしており，FML 開発においても基盤をなす技術である．このフィルタリングの機能の本質を理解するためには単に特徴抽出という狭い分野だけでなく，広く画像処理の分野で用いられているフィルタリング技術について調査研究をしておく必要がある．

そこでこの FML 開発に先だってその基礎となるフィルタリング技術の調査およびノウハウ蓄積のため，文字認識の特徴抽出におけるフィルタリングと画像処理におけるフィルタリングに関し，調査と研究を行なった．以下はその結果についての報告である．

3.1　特徴抽出のためのフィルタリング

3.1.1　線形フィルタ

フィルタについての説明は第１章で述べた通りであるが，ここではそのフィルタマスクの形状について，画像や文字における特徴抽出で用いられる典型的な事例について説明する．

濃度特徴／ボカし／飯島理論

式（3.1）はフィルタマスクの値を決める式であり，A はフィルタの強度を決める係数，x, y はフィルタマスクの中心位置からそれぞれ x, y 方向の画素位置，σ_x, σ_y はボケのパラメータ，f はその時の画素の値である．この関数はガウス関数なのでガウス型マスクと呼ぶ．この式で $\sigma_x = \sigma_y$ とすると等方性のフィルタとなる．

$$f = Aexp(-\frac{1}{2}(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2})). \tag{3.1}$$

これは最も一般的なマスクで，これを用いた特徴抽出は濃度特徴と呼ばれる．またこの種のマスクによるフィルタリングはボケとも呼ばれ，その考え方の背景となっているものが飯島理論 [35][34] である．飯島理論の中に含まれているスケールスペースの概念が重要で，これは式（ 3.1）における σ を変える考え方であるが，実装の場面ではフィルタリングにおけるマスクサイズを変えることにも相当している．実はフィルタリングにおいてマスクサイズを変えることは重要な問題で，これは次節で述べる画像処理の例にもあてはまり，また FML 開発にもあてはまる．

線分抽出型／メキシカンハット

式（ 3.1）において $\sigma_x \neq \sigma_y$ とするとマスクパターンは等方性でなく縦に長いか，横に長い楕円形のマスクになる．これは縦線抽出，横線抽出の機能を持ったマスクで，さらにこの楕円の長軸を傾けることによって様々な線方向を抽出するマスクが作れる．

図 3.1 はメキシカンハット型とガボールフィルタ型のマスクの例で，２次元のマスクの断面を見たものである．横軸は座標値，縦軸はマスク値である．メキシカンハットは中央の値が高く，そこから両側に向かって値が低くなり，いったんマイナス値になった後なだらかにゼロに近付いていくものである．ガボールの場合は周辺で値が波打っているのが特徴である．この図の断面は楕円の短軸で切断した場合を想定したものである．すなわち画像の中の線分の方向がこの断面に垂直であったとして，その中心がこの関数の中央にある時にフィルタリング

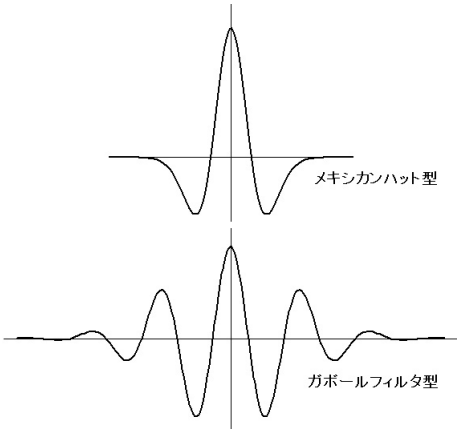


図 3.1: より鮮明な線方向抽出ができるマスクの例（断面）

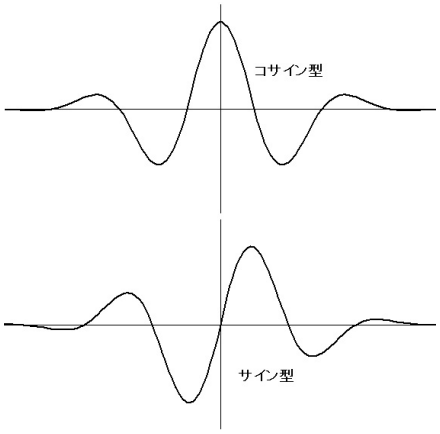


図 3.2: ガボールフィルタの例

出力が最大となる．そして，その線が左右にずれると徐々にフィルタリング出力は下がっていく．すなわち，このマスクはこのマスクの中央にある線分を見付ける性質を持っているのである．

なおメキシカンハット型のマスクの式は１次元で書くと，

$$f = A(1 - \{\frac{x}{\sigma}\}^2)exp(-\frac{1}{2}\frac{x^2}{\sigma^2}), \tag{3.2}$$

であり，積分すると 0 になる．一方ガボールは，

$$f = A\cos(2\pi\frac{x}{u})exp(-\frac{1}{2}\frac{x^2}{\sigma^2}), \tag{3.3}$$

である．この関数はガボールフィルタで用いられるマスクの内，中央の値が高いコサイン型のものである．

このように周囲でマイナスの値を取るこのようなマスクでは線分抽出はより鮮明になる．このような関数はウェーブレットと呼ばれ画像処理，画像理解の分野で多用されている．

このような線分を抽出するマスクは同時にその方向も検出している．従って様々な方向を検出するマスクを用意しないとすべての線分を抽出することができない．つまり線分抽出型マスクでは複数のマスクを用いることが必要となる．このようにマスクを掛ける１つの部分領域に対して複数のマスクを用意することを本報告では多面化と呼ぶことにする．なお本研究の FML に関してはこの多面化の提案およびその実験を行なっている．

ガボールフィルタ

ここではガボールフィルタについて簡単に説明しておく．ガボールフィルタの式を１次元で記述すると，

$$\begin{aligned} f &= A\cos(2\pi\frac{x}{u})exp(-\frac{1}{2}\frac{x^2}{\sigma^2}), \\ g &= A\sin(2\pi\frac{x}{u})exp(-\frac{1}{2}\frac{x^2}{\sigma^2}), \end{aligned} \tag{3.4}$$

となる．これらを図示したものが図 3.2 である．この図の下側がサイン型の関数で，この図に示すように中央で 0 の値を取り，その両サイドでピークを取る．三角関数と同じくピークはコサインとサインで 1/4 周期ずれている．

ガボールフィルタではこのコサインとサインの両方のマスクで部分領域との内積計算を行なう．そしてその結果を a, b とする時，フィルタリング出力値を $\sqrt{a^2 + b^2}$ によって計算する．

このようなフィルタを線分の書かれた画像に用いるとどうなるであろうか？コサイン型マスクの中央より少し右にずれた線分があったとする．するとコサイン型では内積値が下がるが，サイン型での内積値は上昇してくる．したがって，この時のガボールフィルタの出力はそれらの自乗和の平方なので，全体としては線分がずれて

もフィルタリングの出力値は落ちないようになる．なぜなら線分がコサイン型マスクの中央にあった時にはサイン側の内積値は 0 だったからである．

このようにガボールフィルタでは線が移動してもその出力値の下がり方が緩慢であることが分かる．すなわち線分の移動がその線分方向に対して垂直である場合に，そのような移動を横移動と呼ぶことにすると，ガボールフィルタは線分の横移動を伴うような画像変形に強いということになる．文字画像の中には線分が多く，またその横移動も多い．このガボールフィルタが文字認識で有効であることの理由はこれによると考えられる．

なお本研究の FML に関してはこのガボールフィルタを適用した提案と実験も行なっている．本報告ではその方式を多重化と呼んでいる．

3.2 疑似濃淡表現におけるフィルタリング

ここでは特徴抽出とは異なる別の事例として，２値画像による疑似的な濃淡画像表現技術におけるフィルタリングの調査と研究を行った．このような文字認識や特徴抽出でない画像処理におけるフィルタリングの研究調査は，知見を広めノウハウを蓄積しておく意味で重要であり，よりロバストでより適用範囲の広い FML 開発に貢献するものである．

この節ではその調査研究に関する報告を行なう．

3.2.1 疑似濃淡表現とは何か

黒と白の２値による疑似濃淡表現とは多階調出力ができないドット出力装置に濃淡画像を疑似的に出力する場合に用いられる方法で，ファクシミリ，複写，写真製版，画像処理装置の出力部等に応用される．この２値濃淡疑似表現の基本アイディアは従来のどの方式にも共通していて，それは次のような考え方である．すなわち，ある小領域における黒面積の占める割合をコントコールすることによって，その小領域の濃淡を表現するという考え方である．このとき，この黒面積の占める割合はその小領域の濃度と見なされ，この濃度を原画に近づけることにより濃淡画像の疑似表現が行われる．

この考え方にはフィルタリング技術を適用することができる．その方法とは２値化画像の小部分領域においてボカし型のフィルタマスクを掛け，その結果得られる値が原画像のそれに近付くように２値化を行なうというものである．平たく言えば２値化した画像をボカして見た時に，元に戻っていれば良いという考え方である．このことから，この課題はフィルタリング技術との関連が深いことが分かる．

3.2.2 疑似濃淡表現の従来方式

ディザ法

まず，この分野における従来方式を説明する．図 3.3 はディザ法と呼ばれる方式を説明したものであり，上段が網点と呼ばれる方式，下段が組織ディザと呼ばれる方式である．各段の（A）がマスクで,（B）がそれによって生成されるパターンの例である．

ディザ法では，この例で言えば原画像は 4×4 のブロックに分割され，そのそれぞれのブロックに対してディザのマスク（図の（A））が適用される．マスクの値は閾値で，原画像の画素の値がその閾値を越えていれば黒，越えていなければ白というように２値化を行なう．これによれば，もし画素の値がすべて -0.5 であれば総ての画素が白になり，0.5 であれば閾値が 0 の位置のみ黒になる．また，画素が 1.5 であれば閾値が 0 と 1 の位置のみ黒になる．以下同様にして最後に画素の値が 15.5 の時に全画素が黒くなる．この様子を図の（B）に示している．

上段はだんだん黒丸が太る形で網点と呼ばれる．網点は一般の印刷物で用いられる最も一般的な２値化手法である．これに対して下段に示すものは点が格子状に並ぶベイヤーのパターンと呼ばれるマスクであり，これに基づく方式は組織ディザ法と呼ばれている．

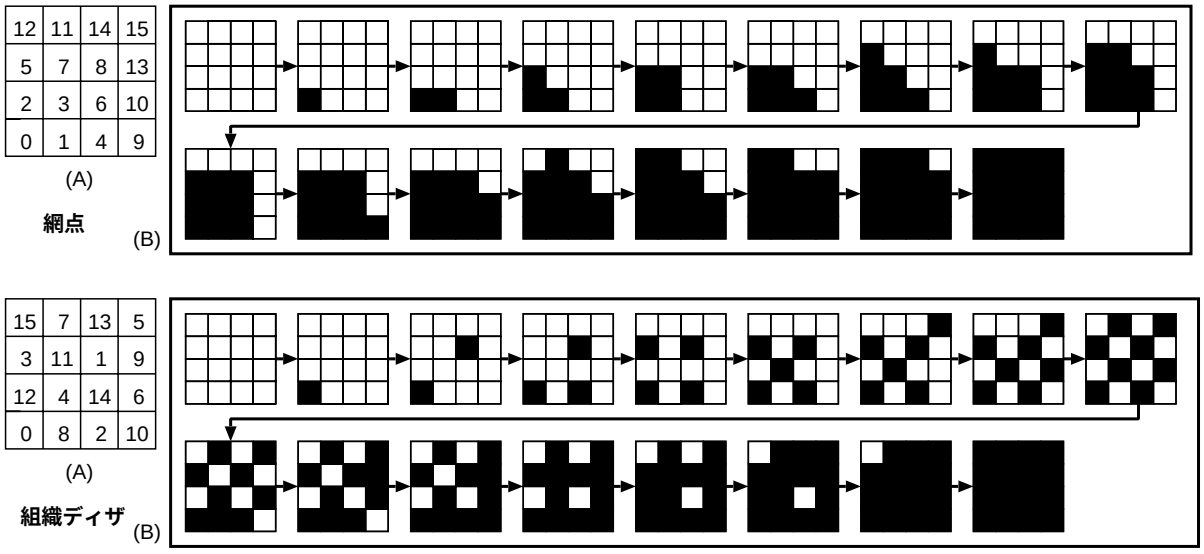


図 3.3: ディザ法におけるマスクパターン（A）と２値化で生成されるパターン（B）

A	B	C	D	E
F	G	H	I	J
K	L	X	∗	∗

1	3	5	3	1
3	5	7	5	3
5	7	X	∗	∗

図 3.4: 平均誤差最小法，平均濃度近似法におけるマスクパターン（左），実際の重み係数の例（右）

ディザ法は次のようにも言い替えられる．すなわちディザ法では，ディザ・パターンに -1 を掛けたものを原画像に加算し，つまり原画像からディザ・パターンを減算し，固定閾値 0 によって２値化している．この考え方は印刷業界や漫画家が使うスクリーン・トーンと同じ考え方である．これは濃度が変化するパターンを持ったスクリーンを濃淡画像に重ね，これを極端なコントラストで撮影することによって濃淡画像を２値化する方法である．一般にこのような手法もフィルタリングと呼ばれるが，本研究で対象とする内積に基づくフィルタリングとは異なるので注意が必要である．

平均誤差最小法

この方式では２値化した後の画素の値と２値化前の画素の値の差を残しておいて，その差の平均が小さくなるように２値化を行なうという考え方である．

図 3.4 左にこの計算のためのマスクパターンを示す．図のXが処理対象画素で，処理は左から右，上から下の横ラスタスキャンとし，図のA〜Lの画素が処理済の画素である．２値化処理においては入力画素値を修正した値 f'_i を求めるものとする．ここで i は画素番号で，$0 \sim n-1$ を画素A〜Lに対応付ける．画素Xの番号を n とする．もちろんこの図のケースでは $n = 12$ である．図のA〜Lで定義された重み係数を w_i，入力画像の画素値を f_i，２値化画像の画素値を g_i として以下の式で２値化を行なうのが平均誤差最小法である．なお図の右側は実際の重み係数の例である．

$$f'_n = f_n + \frac{\sum_{i=0}^{n-1} w_i (f'_i - g_i)}{\sum_{i=0}^{n-1} w_i}. \tag{3.5}$$

$$f'_n \geq 1/2 \Rightarrow g_n = 1. \tag{3.6}$$
$$f'_n < 1/2 \Rightarrow g_n = 0.$$

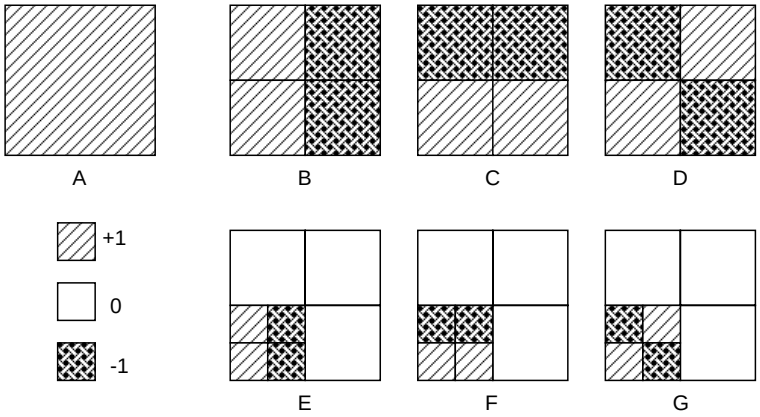


図 3.5: Harr 直交関数系

この式で，$f'_i - g_i$ が２値化誤差で，この誤差の平均を入力画素値に加算したもの，すなわち入力画像の濃度値にこれまでの誤差分を含めたものが f'_n である．この修正された誤差分を含めて２値化を行なうことで，過去の誤差分を近傍の画素における２値化の時にできるだけ解消するようにしているのである．

この方式では出力画像に独特の模様が出るのが欠点であるが，見た目には組織ディザよりも鮮明な疑似濃淡画像が得られる．

3.2.3 多段分割量子化法

ここでは前節で述べた従来方式に対し，本研究に先行する予備的な研究の中で開発した多段分割量子化法 [36][37] について述べる．前に述べた組織ディザ法においては内積あるいは畳み込みによるフィルタリングは用いられていないが，この多段分割量子化法ではそれが使われている．また，生成される疑似濃淡画像の性質は組織ディザ法のそれと良く似ているのが特徴である．以下その方式の説明である．

画像は $2^n \times 2^n$ の画素からなるとし，入力画素の値は 0 から 1 の間にあるとする．この時，この画像の２値化画像全域における黒点の数を入力画像の画素値の合計値を四捨五入して決定する．次に，この全画像を縦横それぞれ等分に分割，すなわち４分割する．そしてこの４領域にさきに決まった黒点を配分する．配分の基本はそれぞれの領域における入力画素値の合計を四捨五入して決めるが，数が合わない時は何らかのルールに基づいて調整する．こうして各領域の黒点数をＦＩＸする．

さらにこれらの各領域を４分割し，その各領域それぞれに割り当てられた黒点を再配分する．

これを繰り返して最終的に分割された領域が１画素になるところで処理を終了する．この方式によって総てのスケールにおける平均的な濃度を入力と出力とで合わせることができる．

この方式はフィルタリングの考え方で説明することもできる．分割された領域において，その領域と同じサイズのマスク値がすべて１のマスクを用意し，入力画像にこれを掛けて出力を求めるとこれは濃度の積分，すなわち平均濃度と画素数の積になる．一方，このマスクを２値画像に掛けると黒点数になる．つまり，この方式では入力画像と２値画像のそれぞれをフィルタリングし，その結果における両者の差が少なくなるように２値化を実行していると解釈することが可能なのである．

このように多段分割量子化法はフィルタリングの考え方に近い．重要なことは，ここではスケールの概念が導入されていることである．スケールの粗いところで優先的に近似操作を行ない，徐々にスケールを細かくしている．このような考え方はスケールスペースと呼ばれていて画像処理の重要な概念である．フィルタリングにおいてもスケールスペースの概念が重要であることはこの例からも分かる．

なお，この多段分割量子化法は図 3.5 に示す Harr 直交関数系に基づく２値化方式としても解釈できる．実際の２値化の例として後に示す疑似濃淡表現の出力画像例はこの Harr に基づくものである．

3.2.4 平均濃度近似法

前に述べた平均誤差最小法そのものはフィルタリングでは無いが，実は式（3.5）を変形するとフィルタリングの形に書き替えられる．この式の右辺の f'_i に，それぞれその f'_i を求める式を繰り返し代入して行くことで，

$$f'_n = f_n + \sum w'_i(f_i - g_i) \tag{3.7}$$

なる形式の式に書き替えられる．ただし，この場合の積和部分は過去のすべてにさかのぼったものになる．２値化は $|f'_n - g_n|$ を最小にする g_n を選ぶことで行なわれるので，上の式は，

$$|(f_n - g_n) + \sum w'_i(f_i - g_i)| \tag{3.8}$$

を最小化することと同じである．この式は，まさに 1 と w'_i からなるマスクによるフィルタリングの結果すなわち，$f_n + \sum w'_i f_i$ と $g_n + \sum w'_i g_i$ とをできるだけ近付けるように２値化するということである．

もし式（3.7）を計算式として用いた方式を考えたとすると，現実問題としては無限の広さのマスクを用いることになる．これは有限で打ち切らざるを得ない．その方式が本研究の一環として開発を行なった平均濃度近似法 [38] である．

この方式を式で表すと式（3.5）において， 右辺の f'_i を f_i に置き替えたものと形式的には同じであるが，具体的には以下の形の評価式で表される．

$$|\frac{\sum_{i=0}^{n} w_i(f_i - g_i)}{\sum_{i=0}^{n} w_i}| = |\frac{\sum_{i=0}^{n} w_i f_i}{\sum_{i=0}^{n} w_i} - \frac{\sum_{i=0}^{n} w_i g_i}{\sum_{i=0}^{n} w_i}|. \tag{3.9}$$

２値化はこの値を最小にするように g_n の値を決めることにより行なう．なお重みは図 3.4 右のものと同じで画素Xの位置の重みは 9 である．

この方式の出力結果はほぼ平均誤差最小法と同じになる．この方式には濃度が白に近い所や黒に近い所で，かつエリアが広い場合，平均誤差最小法に比べて微妙な濃度変化を表現できないという欠陥がある．しかしながら処理速度が早いこと，ハードウェア化した場合の規模が小さいこと，２値画像を印字するドットの大きさや形状を考慮した２値化方式に拡張できる点がこの方式の長所である．なお上記の欠点についてはマスクを大きくするなど幾つかの改善策がある．

3.2.5 各方式による２値表現結果の比較

図 3.6 に各方式の実例をあげる．これにより，組織ディザ法と多段分割量子化法，また平均誤差最小法と平均濃度近似法が良く似ていることが確認できる．

（A）原画

（B）網点

（C）多段分割量子化法

（D）組織ディザ

（E）平均誤差最小法

（F）平均濃度近似法

図 3.6: 疑似濃淡画像方式による結果の比較

3.2.6 FML 開発に向けての開発指針

ここまでの画像処理における特徴抽出法の検討や 2 値画像による疑似濃淡表示法の検討を通してフィルタリング技術の重要性を確認した．ここで，この調査・研究を通して得られた知見を以下にまとめておく．

・マスクサイズ

マスクサイズは大きければ大きいほど良いと予想される．もちろん濃度変化の激しい部分は小さいサイズの方が良いが，これは大きいサイズのマスクの周囲を 0 にすることで実現できる．当然サイズは処理量との兼ね合いで適切なサイズに決める必要があることは言うまでもない．本研究ではこれを 7×7 とした．

・マスク値

マスク値を決める関数としてガウス関数が妥当であると思われる．ボカしの程度は実験的に決める必要があるが，図 3.4 のものや，画像の特徴抽出に良く使われる，

$$\begin{matrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{matrix}$$

などが参考になる．

・非線形性

特徴抽出した結果の値は実験室レベルでは $-\infty \sim \infty$ で良いが，実用上はこれを整数化しなければならない事が多い．すなわち特徴値に上限と下限を設けるということである．経験的にはこうすることで性能が向上することもしばしばであり，このようなサプレス変換の導入は特徴抽出では一般的である．このサプレス変換は非線形変換であり FML でも導入すべきものと考えられる．

・非一様性

本報告では述べなかったが，疑似濃淡画像表示の研究を通してエッジ部とフラット部で異なる方式を用いる重要性についての知見も得られている．このことは画像中の位置によってマスクを変える考え方が重要であることを示しており，本研究ではこれはマスクの非一様性に反映させている．

・多面化と方向性

Harr 直交関数系をフィルタマスクとして見ると，これは 1 つの部分領域に複数のマスクを設定している例になっている．それぞれ縦，横，斜めの線分抽出マスクである．これに限らず画像における特徴抽出フィルタリングではフィルタマスクとして線分方向検出機能を持ったものは良く用いられる．これは FML でも取り入れるべきコンセプトである．

・ガボールフィルタ

画像処理においてガボールフィルタは良く使われる有効な手法である．この方式に FML を適用することは重要であると判断される．

・スケールスペース

多段分割量子化法で見るように当然のことながら複数のマスクサイズ，あるいは 複数の σ を用いるスケールスペースの考え方をフィルタリングに持ち込むことは有効である．当然 FML に用いることも可能であるが，今回の研究では基礎的な調査を優先したのでこれについては研究範囲内に含めず今後の課題の 1 つとした．

第4章　フィルタマスク学習の提案

4.1　特徴抽出学習と特徴抽出フィルタ学習

4.1.1　特徴抽出学習の考え方

フィルタマスク学習について述べる前に，フィルタマスク学習を含む特徴学習という概念について，ここでは簡単に触れることとする．

図 4.1 に特徴抽出学習の全体構造を示す．この図で「入力」は入力画像であり，これが認識対象となる．この入力画像は「特徴抽出部」に入力され，ここで特徴が取り出されて特徴データとして出力される．この特徴データと，あらかじめ設定されている辞書データとの比較を行なうことで認識を実行する．辞書データはカテゴリごとに設定されており，比較は「パターン照合部」で行なわれて，もっとも照合度の高いカテゴリを正解として出力する．ここまでが認識システムであり標準的なパターン認識システムの１つである．

次に学習部分であるが，この図の「辞書更新」と「特徴抽出更新」が学習部分を表している．認識結果に基づいて辞書データと特徴抽出パラメータを更新することによって逐次学習を行なっていく．学習に用いるパターンを学習パターンと呼ぶが，この学習パターンすべてを使って辞書と特徴抽出パラメータを更新し，次にまた，この更新された辞書と特徴パラメータによって同じ学習パターンによる認識を行ない，その結果に基づいた更新を再び実施する．これを繰り返して学習を継続する．終了条件は特に定めていないが認識精度が向上しなくなったと判断したところで学習を終了する．

この更新した辞書と特徴抽出パラメータを用いた認識系を最終的に学習後の認識システムとして FIX する．なお，これに対して出来上がった特徴抽出部のみを取り出してパターン照合部が異なる別の認識システムに適用するという考え方があるが，この場合は学習した時とパターン照合部が違うので認識精度が落ちると予想される．

本方式では学習対象は特徴抽出部なので辞書学習を行なう必要は無いのであるが，辞書を元のままにすると更新した特徴抽出部から出力される特徴データと辞書データの乖離が大きくなってうまく学習が進まない．このため辞書も同時に学習する必要があるのである．もちろん特徴抽出更新のたびに辞書を更新するのではなく，その時に作成し直すという方法もある．しかしながら実験的には良い結果が出なかったので本研究ではそのアプローチは取らなかった．

以上は特徴抽出方式をフィルタに限定しない特徴抽出学習一般の説明である．しかしながら実際にはフィルタ学習以外の特徴学習方式というものは，これまでにあまり提案された事がない．フィルタ型以外の手法にはアドホックなものが多く，学習機構を入れるのが難しいからである．本研究でも特徴抽出方式としては学習機構を入れやすいフィルタ学習を用いている．次にその特徴抽出フィルタ学習について述べる．

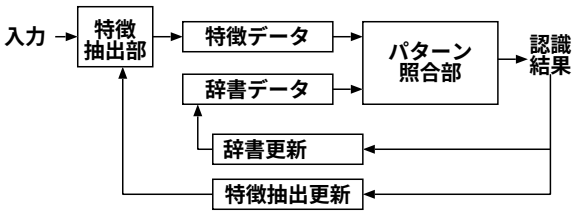


図 4.1: 特徴抽出学習機構の全体構成

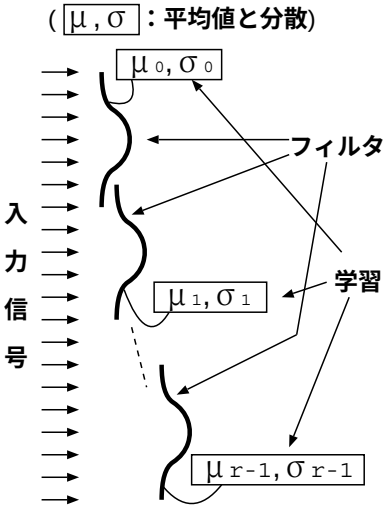


図 4.2: フィルタパラメータ学習（FPL）

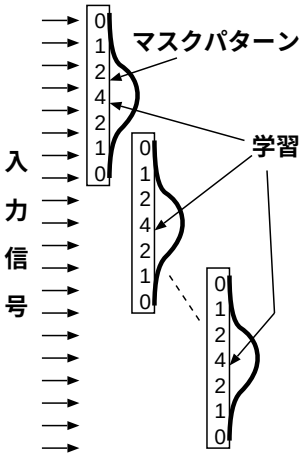


図 4.3: フィルタマスク学習（FML）

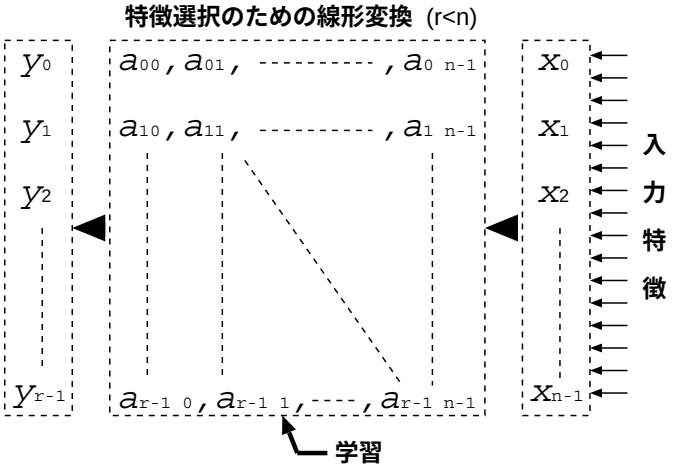


図 4.4: 特徴選択学習（FSL）

4.1.2 特徴抽出フィルタ学習とその種類

特徴抽出フィルタ学習には３つの種類がある．以下，それらを簡単に紹介する．

（１）フィルタパラメータ学習 （FPL） [4],[5],[6]

これは特徴抽出部に設定されたフィルタを数式によって表現し，その数式パラメータを学習するものである．ここではこれを FPL（Filter Parameter Learning）と呼ぶ．図 4.2 にその例を示す．この例は入力が１次元ベクトルの場合のもので，フィルタマスクをガウス関数 $exp(-(x-\mu_i)^2/(2\sigma_i^2))$ としたものである．この時，μ_i, σ_i を学習対象とするのがこの方式の考え方である．

（２）フィルタマスク学習 （FML） [41][42][43] [48][49][50]

これはフィルタのマスクパターンの値そのものをパラメータと見なし，それを直接学習する手法で，本研究で扱う手法である．図 4.3 にその例を示す．この例のフィルタの構成は図 4.2 と同じであるが，ここでは学習対象はマスクの値そのもの（図 4.3 の 0124210）である．

（３）特徴選択学習 （FSL） [7] [8][9][10]

これは特徴抽出機構そのものを学習するのでは無く，特徴抽出後の特徴ベクトルの次元圧縮を目的とした線形

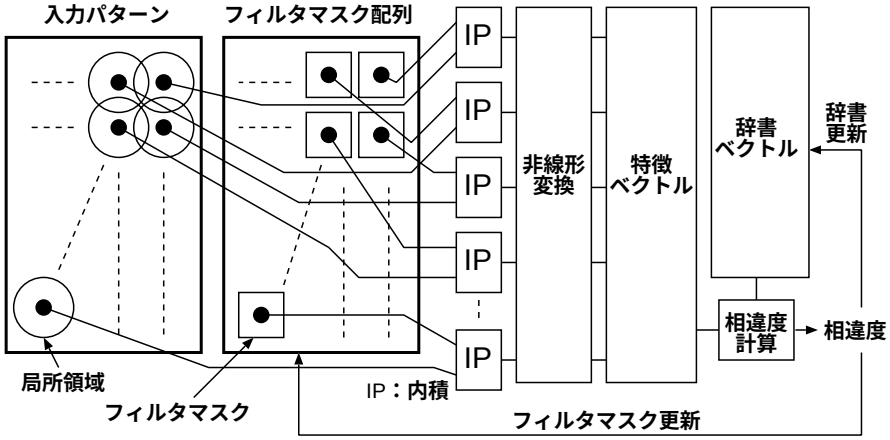


図 4.5: フィルタマスク学習機構の全体構成

変換行列を学習対象とするものである．ここではこれを FSL（Feature Selection Learning）と呼ぶ．図 4.4 にその例を示す．この図は，線形変換によって n 次元の入力特徴ベクトルが r 次元 $(r < n)$ に圧縮された様子を示したものである．上記の（1）（2）では図 4.1 の入力はイメージそのもの，あるいはボカしたものであるが，この構成では入力には特徴ベクトルを用いる．また,（1）（2）で採用されているような局所性は無い．ここでの局所性とはフィルタマスクの値が定義されるエリアが，画像全体の中で一部の局所領域に限定されているという意味である．図 4.1 に記載した特徴抽出の部分はこの構成では特徴選択機構に対応し，これとは別に特徴抽出が外側にある形になる．この方式の計算上の形式は（2）と同じであるが実用上は異る処理系である．

本研究が対象とする FML は（1）の FPL に比べてパラメータの数が圧倒的の多いので，より高い性能が期待できる．それにもかかわらず，これまであまり研究された来なかった理由は学習処理量が大きくなることや，形式が単純でニューラルネットに類似していることから学術的な興味があまり持たれなかったためと考えられる．本研究ではその点にはこだわらず実際の OCR 製品で役に立つ手法として本方式を調査対象とした．

4.2 フィルタマスク学習（ＦＭＬ）の基本構造

4.2.1 ＦＭＬの枠組

図 4.5 に FML の構成を示す．この図の入力パターンの中に複数の局所領域（○で示す）を設定し，その局所領域内の画素の値からなるベクトルと，局所領域ごとに設定されたフィルタマスク（□で示す），すなわち重みベクトルとの内積（IP で示す）を計算し，これを非線形変換して特徴ベクトルの各要素値とする．このフィルタマスクを集めたものがこの方式で学習対象とする特徴抽出パラメータであり，以下ではこれを「マスク配列」または単に「マスク」と呼ぶ．

特徴ベクトルは辞書ベクトルとの距離計算が施され，それに基づいた認識が実行される．この認識結果に基づいたフィードバックがフィルタマスクと辞書ベクトルに対してあり，これによってフィルタマスクと辞書が更新される．

これは前にも述べた事だが，このシステムで特徴的なのはフィルタマスクが各局所領域で異る非一様性である．これまで，文字認識の世界ではフィルタマスクはどの局所領域でも同じものを用いていた．別の言いかたをすれば認識系の中にフィルタマスクは１つしかなかった．その理由の１つは計算機リソースが昔は十分でなかったということである．この問題は現在ではまったく問題にならない状況であるが，それでも実際に各部分領域のマスクをどう設計するかという課題が残っている．本研究はこの課題に対する１つのアプローチでもある．

次に，このフィルタマスクをどう学習するかについて説明する．

4.2.2 認識精度評価式と降下法

まず一般論として最適な認識パラメータを競合学習で求める手法について述べる．これは確率降下法（PD）[1] を簡略化した考え方であるが，その元となる PD については後で述べる．なお，この PD とはパターン認識における逐次学習手法の１つであり，音声認識の分野に拡張適用されて MCE/GPD として知られているものである．この方式は文字認識にも適用されている．

本研究における学習の考え方では，まず認識精度を評価する認識精度評価関数を設定する．この認識精度評価関数とは認識精度の悪さを表現する数値であり，たとえばある認識パラメータに対する誤読率などがそれにあたる．

競合学習とは認識結果が正解の場合と，不正解の場合の両方の結果を用いて学習を行なうことであり，上記の認識精度評価関数の中にはその両方の結果に関する情報が盛り込まれるようにする．

認識パラメータとは認識系の中で用いられる各種数値データで，例えば辞書データや特徴抽出フィルタマスクのマスク値などがそれにあたる．

学習パターンを x, 認識精度評価値を J ，認識パラメータを Λ_p ，この認識パラメータを並べたベクトルを Λ と書き，認識精度評価関数をこの Λ の関数として $loss(\Lambda, x)$ とする．これらによって認識精度評価式を以下のように記述する．

$$J = \sum_x loss(\Lambda, x) \tag{4.1}$$

この式で J を最小にする Λ を探すことが学習の課題である．かなずしも最小値になるわけでは無いが，そのための解法として最急降下法を用いる．従って Λ を逐次変更する漸化式は以下で与えられる．更新後のベクトル及びベクトルの要素値を $\Lambda^{new}, \Lambda_p^{new}$ として，

$$\Lambda_p^{new} = \Lambda_p + \Delta\Lambda_p, \tag{4.2}$$

$$\Delta\Lambda_p = -\varepsilon \frac{\partial J}{\partial \Lambda_p} = -\varepsilon \sum_x \frac{\partial loss(\Lambda, x)}{\partial \Lambda_p}. \tag{4.3}$$

ここで ε は学習強度を決める係数．この漸化式をベクトルで表現すると，

$$\Lambda^{new} = \Lambda + \Delta\Lambda, \tag{4.4}$$

となる．

しかしながら，これでは実際の計算ができないのでもう少し現実的な形式を次に示す．

まず相違度 S を求める関数を定義する．この相違度は各カテゴリごとに計算され，カテゴリ名を c とした時 S_c と書き，その関数を $g(\Lambda_{c;p}, \Lambda_{a;p}, x)$ とする．ここで $\Lambda_{c;p}$ はカテゴリ c に依存して決まるパラメータで例えばカテゴリ c の辞書パラメータがそれにあたる．また $\Lambda_{a;p}$ は各カテゴリに共通のパラメータで特徴抽出パラメータなどがそれにあたる．ある学習パターン x に対する正解カテゴリの相違度を $S_{ok} = g(\Lambda_{ok;p}, \Lambda_{a;p}, x)$ とし，不正解カテゴリの相違度を $S_{err} = g(\Lambda_{err;p}, \Lambda_{a;p}, x)$ とする．この時 x に対する認識精度評価関数をある単調増加関数 $l()$ を準備し，$d = S_{ok} - S_{err}$ として $l(d)$ と定義する．この中で d は値が大きい程，正解の相違度が高く不正解の相違度が低いので，d は認識精度の悪い程度を表している．$l()$ は単調増加関数なので $l(d)$ もまた認識精度の悪い程度を表している．この $l(d)$ を学習パターン全体について積算したものを最終的な認識精度評価関数とする．すなわち，

$$J = \sum_x l(d) = \sum_x l(S_{ok} - S_{err}) = \sum_x l(g(\Lambda_{ok;p}, \Lambda_{a;p}, x) - g(\Lambda_{err;p}, \Lambda_{a;p}, x)), \tag{4.5}$$

である．ここで， Λ_p を $\Lambda_{a;p}$ や $\Lambda_{c;p}$ を表すものとして， J を Λ_p で偏微分をすると，

$$\begin{aligned}\frac{\partial J}{\partial \Lambda_p} &= \sum_x l'(d)(\frac{\partial S_{ok}}{\partial \Lambda_p} - \frac{\partial S_{err}}{\partial \Lambda_p}) \\ &= \sum_x l'(d)\{\frac{\partial g(\Lambda_{ok;p}, \Lambda_{a;p}, x)}{\partial \Lambda_p} - \frac{\partial g(\Lambda_{err;p}, \Lambda_{a;p}, x)}{\partial \Lambda_p}\},\end{aligned} \tag{4.6}$$

なので，これにより Λ_p の更新量は，

$$\Delta\Lambda_p = -\varepsilon \sum_x l'(d)(\frac{\partial g(\Lambda_{ok;p}, \Lambda_{a;p}, x)}{\partial \Lambda_p} - \frac{\partial g(\Lambda_{err;p}, \Lambda_{a;p}, x)}{\partial \Lambda_p}) \tag{4.7}$$

となる．この $\Delta\Lambda_p$ は S_{ok} と S_{err} の項を加算したものであるが，このそれぞれを分離して別々に加算するとすれば，上記更新量は S_c を S_{ok} または S_{err} を表すものとして，

$$\Delta\Lambda_p = \mp\varepsilon \sum_x l'(d)\frac{\partial S_c}{\partial \Lambda_p} = \mp\varepsilon \sum_x l'(d)\frac{\partial g(\Lambda_{c;p}, \Lambda_p, x)}{\partial \Lambda_p}, \tag{4.8}$$

と書ける．さらに１パターンごとの更新量を表記すると $\sum_x$ が取れて，

$$\Delta\Lambda_p = \mp\varepsilon l'(d)\frac{\partial S_c}{\partial \Lambda_p}, \tag{4.9}$$

となる．この式で，ok ， err はまとめて c と記述され，$\mp$ の上側は正解カテゴリ，下は不正解カテゴリに対する更新量である．ただし S が相違度で無くて類似度の時は，$d = S_{err} - S_{ok}$ と置かなければならないので上記の符号は $\mp$ では無く $\pm$ となる．

なお本研究の実験で用いている方式は，この更新量を全学習パターンに対して積算し，その後，一括して更新操作を行なう方式を採用している．その意味では式（ 4.9）のように $\sum_x$ を省略することができない．しかしながら $\sum_x$ を式に残すとやや繁雑になるのと，また一般にこの種の漸化式では $\sum_x$ は省略されていることが多いので，本報告ではほとんどの場合に $\sum_x$ を省略した１パターンごとの更新形式を用いていることを注意しておく．

これが認識精度評価関数と降下法による学習の一般形である．ここで説明した方式は PD と本質的に同じものであり，ここで定義した認識精度評価値 J と $l()$ は PD では平均損失，損失関数とそれぞれ呼ばれている．なお PD を拡張した MCE/GPD では J に相当するものは MCE と呼ばれている．そこで本論文では以降， J を平均損失， $loss()$ を認識精度評価関数，$l(d)$ を損失関数と呼ぶことにする．

なお本研究で用いた学習方法を PD と呼ばなかった理由は，厳密には方式に異なる部分があるからである．その違いとは学習が１パターンごとの更新か，あるいは一括しての更新かの違いにある．しかしながらこれは本質的な違いでは無いので，俯瞰的に言えば本方式を PD を特徴抽出に適用したものと考えて良い．また，これも厳密には異るので正しくは無いが，MCE/GPD を特徴抽出に適用したものは DFE[4] と呼ばれるので，本方式をこの DFE の一種であると見なすこともできる．

4.2.3　基礎的なＦＭＬの更新式

上で述べた一般的な形式に FML を適用することによって得られる更新式を以下に示す．フィルタマスクが定義されている局所領域について，番号 i の局所領域に対応するマスクをベクトル形式で $f^{(i)}$，その要素値を $f_j^{(i)}$，局所領域の入力パターンの画素値を並べたベクトルを $z^{(i)}$，その要素値を $z_j^{(i)}$，その内積値を $\hat{x}_i$，非線形関数を $\rho(\hat{x}_i)$，出力特徴ベクトルの要素値を x_i とすると，$x_i = \rho(f^{(i)T} z^{(i)})$ と書ける．一方，認識部については，x_i を並べて作った特徴ベクトル x と辞書ベクトル φ の相違度をユークリッド距離の自乗で計算し，最も相違度が小さいものを正解として出力することによって認識を行う．特徴ベクトル x の次元を n とし，φ_i を φ の要素として相違度を次式で表わす．

$$S = \|x - \varphi\|^2 = \sum_{i=0}^{n-1}(x_i - \varphi_i)^2 = \sum_{i=0}^{n-1}(\rho(\sum_j f_j^{(i)} z_j^{(i)}) - \varphi_i)^2. \tag{4.10}$$

従って，φ と $f_j^{(i)}$ の更新式は $\hat{x}_i = \sum_j f_j^{(i)} z_j^{(i)}$ なので，

$$\frac{\partial S}{\partial \varphi_i} = -2(x_i - \varphi_i) \tag{4.11}$$

$$\frac{\partial S}{\partial f_j^{(i)}} = \frac{\partial S}{\partial x_i}\frac{\partial x_i}{\partial \hat{x}_i}\frac{\partial \hat{x}_i}{\partial f_j^{(i)}} = 2(x_i - \varphi_i)\rho'(\hat{x}_i) z_j^{(i)}, \tag{4.12}$$

である．よって，

$$\Delta\varphi_i = \pm 2\varepsilon_r l'(d)(x_i - \varphi_i). \tag{4.13}$$

$$\Delta f_j^{(i)} = \mp 2\varepsilon_f l'(d)\rho'(\hat{x}_i)(x_i - \varphi_i) z_j^{(i)}, \tag{4.14}$$

となる．これらの式で，$\varepsilon_r, \varepsilon_f$ は学習の強度を決める定数である．

ここでベクトルの要素をベクトルに置き換えることにより，上の式は以下のように若干スッキリした形になる．

$$\Delta\varphi = \pm 2\varepsilon_r l'(d)(x - \varphi). \tag{4.15}$$

$$\Delta f^{(i)} = \mp 2\varepsilon_f l'(d)\rho'(\hat{x}_i)(x_i - \varphi_i) z^{(i)}. \tag{4.16}$$

さらにこの式を簡略化するために次の記法を導入する．まず k を入力画像中の画素番号とし，それに対応する局所領域 i の画素の画素番号を j とする．この関係がある時に $k = R_i(j)$ と表し，

$$z_k = z_j^{(i)} \quad (k = R_i(j)), \tag{4.17}$$

とする．この時，

$$F_{ik} = \begin{cases} f_j^{(i)} & \text{if there is } k; k = R_i(j)) \\ 0 & \text{else} \end{cases} \tag{4.18}$$

と置くことで，フィルタリングの式は，$\hat{x}_i = \sum_k F_{ik} z_k$ となる．これにより，マスク学習の更新式は，

$$\Delta F_{ik} = \mp 2\varepsilon_f l'(d)\rho'(\hat{x}_i)(x_i - \varphi_i) z_k. \tag{4.19}$$

となる．次に $\rho'(\hat{x}_i)$ を対角要素として並べた対角行列を考え，これを $\rho'(\hat{x})$ と表し，フィルタマスクの更新式を辞書の更新式と合わせてベクトルで表すと，以下のように更にスッキリした形になる．

$$\Delta\varphi = \pm 2\varepsilon_r l'(d)(x - \varphi). \tag{4.20}$$

$$\Delta F = \mp 2\varepsilon_f l'(d)\rho'(\hat{x})(x - \varphi) z^T. \tag{4.21}$$

ここで上の式変形における $(x - \varphi)z^T$ について説明しておく．z^T の T は転置である．一般に，ベクトル x とその n 個の要素 x_i と，ベクトル y とその m 個の要素 y_j があった時，$x_i y_j$ について，i を 0 から $n-1$ まで縦方向に変化させ，j を 0 から $m-1$ まで横方向に変化させ，この $x_i y_j$ を縦横に並べた行列を A とする．すると $A = xy^T$ と書ける．上の式変形はこれに基づいた記述となっている．

この式は 4.1.2 の「特徴抽出フィルタ学習とその種類」で説明した（3）の「特徴選択学習」の形式と同じであり，フィルタリングを線形変換として理解する上で直観的に分かりやすい表現である．

次に本研究における φ, F の初期値設定について述べる．本研究では初期フィルタとして文字認識で通常用いられるフィルタマスクの値を用いた．また初期辞書ベクトルについては，このフィルタを用いた特徴抽出によって得られる特徴ベクトルを平均して作成した．初期値としては一般には何を使っても良いが，実用上は初期値設定は重要であり学習後の性能に大きな影響を与える．その意味で初期値としてはできるだけ性能の良い既知のパラメータを与えることが好ましく，本研究でもそのポリシーに従っている．

ニューラルネットなどでしばしばランダムな値が初期値として用いられるが，そうする事によって初期段階の性能を低く押えて学習効果をより一層際だたせることができるものの，実用上の観点からは好ましくは無い．

本手法においてフィルタマスクの更新を行なわない，すなわち辞書学習のみを行なう手法は LVQ と同じである．実際，式（4.20）は LVQ の更新式と同じ形式である．それ故，本方式は LVQ の拡張とも見てとることができる．つまり LVQ に特徴抽出フィルタマスク学習を付け加えた手法とも考えることができる．本研究の目的はより高性能なパターン認識方式を見つけ出していくことにあり，全く新規な手法を提案しようというものでは無いので，その意味でパターン認識の有力手法である LVQ をベースとして用いたのである．またフィルタマスク学習の構成については，次に述べる確率降下法が LVQ を解析的に説明できるコンセプトであることから，確率降下法の考え方に従った．

4.3 確率的降下法およびニューラルネットとの関係

4.3.1 確率的降下法

ここで，本研究と関係の深い確率的降下法（PD）について述べる．PD と関係が深い方式に LVQ があり，パターン認識の分野ではこの LVQ が競合学習手法の１つとして良く知られている．この方式は PD によって説明することができ，従って LVQ は PD の典型的な１具体例と考えることができる．

発表の順序から言うと，PD は LVQ よりも先に提案されており，また辞書ベクトルに対する更新式は LVQ と同じものが PD の論文に代表例として記述されている．したがって歴史的には，LVQ を PD の１具体例と見ることは自然であるが，一般には PD よりも LVQ やまたその同系統の方式である SOM（Self Organizing Map）の方が良く知られていて，LVQ が PD の具体化であるいう認識は多くない．そのような観点からは，すなわち LVQ 側から見れば，PD を，LVQ を一般化あるいは拡張した LVQ 的手法の１つと言うこともできる．

以下，PD について述べる．PD では，まず認識パラメータ Λ を有する認識系を定義する．入力されたパターンを x として，そのパターンに対応して何らかの数値を与える関数 $g(\Lambda, x)$ を定義する．この数値は認識に用いられるもので，この数値に基づいて次の損失関数が定義される．すなわち認識誤りが多い場合に高い値，認識誤りが少ない場合に低い値を取るように，損失関数 $l(g(\Lambda, x))$ を定義する．そして x の出現確率を $p(x)$ として，損失関数の値を全ての入力パターンについて積分したものを評価値 J とする．

$$J = \int l(g(\Lambda, x))p(x)dx. \tag{4.22}$$

ここで認識系の学習を，認識系が関数 $g()$ とパラメータ Λ で構成されているとして，その Λ を求めることとする．すると，この J を最小にするパラメータ Λ を求めることが，すなわち認識系の学習と言うことになる．

PD では，この Λ を求める手段として最急降下法を用いる．すなわち，Λ の要素を Λ_p ，その更新量を $\Delta\Lambda_p$ とした時，次に示す式で Λ を更新して J を最小値に近づけ Λ を最適化する．

$$\Delta\Lambda_p = -\varepsilon\frac{\partial J}{\partial \Lambda_p}. \tag{4.23}$$

しかしながら，この式では J を全入力パターンに対して求めなければパラメータの１回の更新ができないので，処理時間が膨大になるという問題がある．そこで入力１パターンごとに更新を行うように更新式を以下のように変更する．

$$\Delta\Lambda_p = \frac{\partial l(g(x, \Lambda))}{\partial \Lambda_p}. \tag{4.24}$$

このようにしても J を最小にすることができるというのが PD の考え方であり，１パターンづつ学習しても，確率的な意味で全部を一括して学習した場合と同じになるという考え方が PD として主張されていることである．この方式が確率的降下法と呼ばれているのはこの理由による．

一方で，この１パターンづつ学習を行うという PD の考え方は実装上はかえって問題があり，処理速度が遅くなったりプログラムが複雑化するケースがあって，必ずしも使い易いものではない．この PD が考えられた 1967 年当時は CPU の処理速度が現在に比べると数桁違いに遅かった時期であるから，辞書更新をすべてのパターンの処理を終えてから行なうというのでは，ほとんど学習が進まない状況であったので，PD の考え方には

意味があったのである．しかしながら現在の CPU パワーでは例えば，実際に文字認識などで用いられる学習パターンをすべてを学習するのにそれ程の時間はかからない．むしろ，このような状況で 1 パターンづつパラメータを更新するのは，かえってオーバーヘッドが大きいのである．したがって現在の CPU の場合には PD の「確率的」という部分は実質的には不要ということになってしまう．

本研究で用いているこの部分の実装法は全パターン一括，すなわち全パターンについての更新量を積算して，そのあとで一括してパラメータの更新を行っている事は前に述べた．このように本研究の手法は厳密には PD では無いのであるが，本質的には PD の範疇の方式である．

以下に損失関数について説明する．確率降下法の提案で示された損失関数の例は次のようなものであり，本研究でも用いていて，この損失関数に基づく評価値 J は平均リスク（Average Risk）と呼ばれている．

認識対象パターンをベクトル x で表し，c をカテゴリ名、Λ_c をそのカテゴリに対応した辞書とする．認識対象パターンと同じカテゴリ名を ok，認識対象と異るカテゴリの中で最高（相違度で言えば最小）の認識結果となったカテゴリ名を err で表す．すなわち，ok が正解カテゴリ， err が不正解カテゴリである．この時，それぞれのカテゴリの相違度 S_c を次のようにして求める．ここで $g(x, \Lambda_c)$ は相違度を計算する関数である．

$$S_{ok}(x) = g(x, \Lambda_{ok}) \tag{4.25}$$

$$S_{err}(x) = \min_{c \neq ok} g(x, \Lambda_c) = g(x, \Lambda_{err}). \tag{4.26}$$

損失関数は $l()$ を単調増加関数として次のように定義される．

$$l(x) = l(S_{ok}(x) - S_{err}(x)). \tag{4.27}$$

これにより評価関数は，学習パターン全体を Ω で表して次のようになる．

$$J = \int_{x \in \Omega} l(x)p(x)dx = \int_{x \in \Omega} l(s_{ok}(x) - s_{err}(x))p(x)dx. \tag{4.28}$$

確率降下法では，この J を偏微分して更新量を求めるかわりに $l(x)$ を偏微分する．本研究では先に述べたように学習 1 回は全パターン分一括なので偏微分は J に対するものであるが，この計算自体は本質的にはどちらも同じで，いずれも更新量を求めるための偏微分は特に区別することなく $l(x)$ に対して行なうことになる．したがって 4.2.2 で示したものと同じ計算により更新量は以下のように表現される．

$$\Delta\Lambda_p = \mp\varepsilon l'(d)\frac{\partial S}{\Lambda_p}. \tag{4.29}$$

ここで，$\mp$ の $-$ は正解カテゴリに対する更新量，$+$ は不正解カテゴリに対する更新量である．

次に相違度としてユークリッド距離を用いた場合の更新式を求める．認識パラメータとして標準ベクトル m を用意して，これにより相違度を

$$S = ||x - m||^2, \tag{4.30}$$

で定義する．この場合も 4.2.2 と同じ計算により更新式は，

$$\Delta m = \pm 2\varepsilon l'(d)(x - m), \tag{4.31}$$

となる．これは LVQ と同じ形式であり $2l'(d)$ が LVQ の窓関数に相当している．

4.3.2 ニューラルネットとの関係

図 4.6 は FML と NN の関係を示したもので，上段が FML で下段が 3 層の NN である．FML の入力パターンが NN の入力層，フィルタマスクが入力層と隠れ層の間の重み係数，内積＆非線形変換が隠れ層，辞書ベクトルが 2 番目の重み係数，距離計算が出力層に相当している．

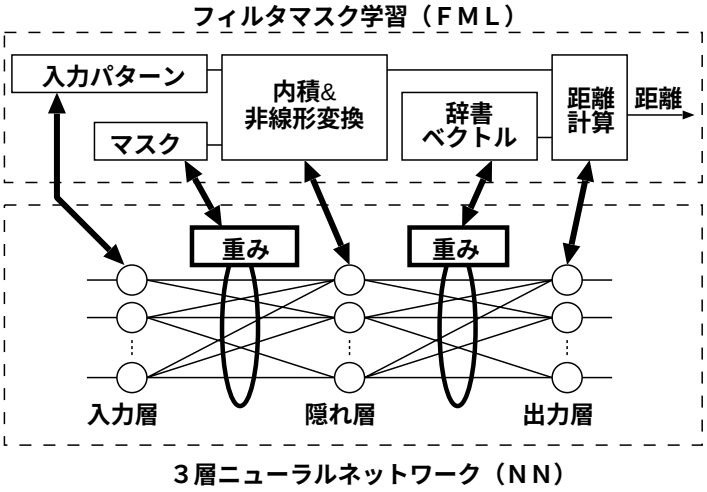


図 4.6: FML と NN の関係

典型的な NN では各ユニットの計算は内積で，出力結果の評価計算は教師信号とのユークリッド距離である．これに比べると FML では最終ユニットの計算がユークリッド距離であり，出力結果の評価は PD の平均リスクに基づくものである．

NN のユニット間結合に全結合と局所結合がある．全結合とはある層のユニットが隣の層のすべてのユニットに接続されているもので，局所結合とはそれが一部に限定されているものである．本研究で扱う方式のように１つ１つのユニットが画素に対応し，画素の位置関係に基づいてユニット同士が影響しあう系では，局所的な結合が妥当である．一般的に課題がフィルタの時は局所結合であり本研究での仕様も局所結合である．

また一様重みか非一様重みかという選択肢がある．一様重みとは局所領域の場所によらずその重み係数がすべて同じものの事であり FML の一様マスクに対応する．非一様重みとは重み係数が局所領域によってそれが異るように設定されたもので FML の非一様マスクに対応する．

一様重みはパターンの状況が場所に依存しないことが条件で，その場合に少ない学習パターン数でも学習が局所領域の数だけ倍増されるので，学習効果が増大するという特長がある．またマスクパターンの格納スペースを小さくできることも利点である．

しかしながら本研究では画像内の場所によってパターン識別の状況が変わるケースを対象としているので，この仕様に関しては非一様重みを採用している．

これは前にも述べた事だが，NN の分野で非一様重みが珍しく無いのに対して文字認識の分野では非一様重みの例は少ない．その理由は，最近ではそんな事は無いが，メモリ容量の増加と，そもそもその重みをどう設計するかという問題があったからである．

NN の立場から本研究の目的を述べると，上記のような NN の仕様において実際のパターン認識での効果はどうか，ユニット間結合の重みの姿が学習によってどう変化するかを観察・確認するという事になる．

以上は FML を NN として見た時の仕様であり，まとめると以下のようになる．

（１）３層構造
（２）最終段のユークリッド距離計算
（３）平均リスク型の評価関数
（４）局所結合
（５）非一様重み

4.4 内積型類似度に基づくＦＭＬ

上記の NN の仕様で「最終段のユークリッド距離」を通常の NN のように内積に置き換えるとどうなるであろうか？もし，その場合でも状況が変わらなければ評価関数の違いがあるものの NN でも本研究で見られたような現象が起こっている可能性が高いと推測できる．

NN で最終段を内積にするということは，FML ではユークリッド距離を内積演算に置き換える事に相当している．ユークリッド距離と内積演算は同じ性格の認識手法であることから [39]，それらに基づく FML の結果も似ていることが予想される．そこで，ここでは内積型の類似度演算に基づく FML について述べることにする．

4.4.1 内積演算を用いた正規化無視型ＦＭＬ

内積演算による類似度は単純類似度とも呼ばれ，正規化を考慮しない場合，すなわち辞書ベクトルも入力ベクトルも正規化されている場合の形式は以下の通りである．

$$S = (x, \varphi)^2 = (\sum_i x_i \varphi_i)^2. \tag{4.32}$$

この時，

$$\frac{\partial S}{\partial \varphi_i} = 2(x, \varphi) x_i, \tag{4.33}$$

$$\frac{\partial S}{\partial f_j^{(i)}} = 2(x, \varphi)\varphi_i \frac{\partial x_i}{\partial \hat{x}_i} \frac{\partial \hat{x}_i}{\partial f_j^{(i)}} = 2(x, \varphi)\varphi_i \rho'(\hat{x}_i) z_j, \tag{4.34}$$

なので，これに対応する更新式は次のようになる．なお，ここでは S は相違度では無く類似度なので，式（4.9）の符号を逆にしたものを適用している．

$$\Delta\varphi_i = \pm 2\varepsilon_r l'(d)(x, \varphi) x_i. \tag{4.35}$$

$$\Delta F_{ij} = \pm 2\varepsilon_f l'(d)\rho'(\hat{x}_i)(x, \varphi)\varphi_i z_j. \tag{4.36}$$

さらに，これらをベクトル表現して，

$$\Delta\varphi = \pm 2\varepsilon_r l'(d)(x, \varphi) x. \tag{4.37}$$

$$\Delta F = \pm 2\varepsilon_f l'(d)\rho'(\hat{x})(x, \varphi)\varphi z^T, \tag{4.38}$$

となる．なお式（4.32）では正規化が考慮されていない．従って式（4.37）を適用した後に φ を正規化する必要がある．

4.4.2 内積演算を用いた正規化考慮型ＦＭＬ

一方，正規化を考慮するケースについては，

$$S = \frac{(x, \varphi)^2}{\|x\|^2 \|\varphi\|^2}, \tag{4.39}$$

が類似度の式であり，更新式は，

$$y = \frac{x}{\|x\|}, \quad \psi = \frac{\varphi}{\|\varphi\|}, \tag{4.40}$$

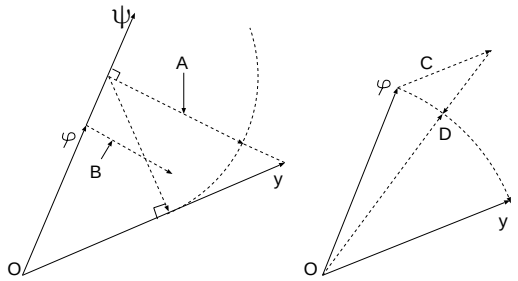


図 4.7: 正規化考慮型の辞書更新（左）と無視型の辞書更新（右）

と置いて以下のようになる．

$$\Delta\varphi = \pm 2\varepsilon_r l'(d)\frac{1}{\|\varphi\|}(y,\psi)\{y-(y,\psi)\psi\}. \tag{4.41}$$

$$\Delta F = \pm 2\varepsilon_f l'(d)\rho'(\hat{x})\frac{1}{\|x\|}(y,\psi)\{\psi-(y,\psi)y\}z^T. \tag{4.42}$$

この式のの導出に関する詳細は付録 A 参照.

認識システムに降下法を適用するための計算では類似度を偏微分するが，その時，式（4.32）ではなく式（4.39）を使うとどうなるかを，ここで考えてみよう．式（4.39）を使うと入力パターンが超球面に垂直な方向に変動しても類似度は変わらないので，垂直方向の偏微分値は 0 となる．従って，その偏微分に基づいて求まる更新ベクトルには超球面に垂直な成分が無い．すなわち更新は超球面に接する方向に移動して行われることになる．このことを図 4.7 で説明しよう．

図 4.7 は辞書ベクトルの更新について，正規化を無視した場合と正規化を考慮した場合の違いを示している．左が正規化を考慮した場合で，正規化された入力ベクトル y に対して正規化されていない辞書ベクトル φ と，これを正規化した ψ を示している．この図で，ベクトル A が式（4.41）における $(y,\psi)(y-(y,\psi)\psi)$ を示し，B が φ の更新ベクトル $\Delta\varphi$ である．従って更新後のベクトルは $\varphi+B$ となる．これは O を中心とする超球面に接する方向である．

一方，右は正規化を無視した場合で，更新ベクトルは y と並行なベクトル C である．更新後のベクトルは $\varphi+C$ であるが，これがさらに強制的に正規化されて D になる．

この両者を比較すると更新方法は全く異ることが分かる．しかしながら正規化した後の D と $\varphi+B$ で比較すると，それほど違わないようにも考えられる．この差は実験レベルで検討すべき課題であるが，経験的には特に問題無い可能性も高いと考えられる．

4.4.3 正準化

正準化 [44][45] は文字のツブレに強い前処理とされており，単純類似度を用いる時に良く使われる手段であるので，ここで正準化を行なった場合の FML について検討しておく．

正準化は複合類似度法 [18] が提案された時に，その一部として発表されたものでボケ変換に強い前処理とされている．文献 [18] で提案された正準化は一般形式で表現されており直観的には理解しにくいが，文献 [19] にその直流成分に対する考え方が分かりやすく書かれている．また，この正準化に確率降下法を適用する手法については文献 [20] に報告がある．

正準化とは要素値が全部 1 のベクトルを $\vec{1}$ と表記し，入力ベクトルを ζ，出力ベクトルを φ として以下ような変換を言う．

$$\varphi = \zeta - \frac{1}{n}(\zeta,\vec{1})\vec{1}. \tag{4.43}$$

正準化とは入力特徴ベクトルからそのベクトルの要素値の平均を各要素値から減算する処理である．こうする事によって各要素値の平均は０となるので，正準化は，ひらたく言えば直流成分を除くコンデンサーのような役割を持つものである．

この処理を導入しないと内積計算結果に直流成分に相当する量が常に加算された状態になり，類似度計算結果がその分の下駄を履きダイナミック・レンジが小さくなるという問題がある．実用上は高々１～２ビットのことなので今の時代には問題無いことかも知れないが，かつては深刻な問題であった．また正準化によれば，パターンの分布は広がる方向に変化するので，そもそも認識精度改善に役立つという考え方もある．

次に正準化がつぶれ文字に強いという性質について若干考察する．一般に入力文字が大幅にツブれると，そのストロークは大きく太ることになり，結果としてパターンの平均値が大きく上昇する．逆にカスれさせると平均値が下る．このような平均値すなわち直流成分の大幅な変化が類似度に与える影響は大変に大きい．それ故，この様な変形の変動を吸収するという正準化の効果は容易に推測することができる．

一方で平均値が変化しないボカしやカスれの操作ではどうなのであろうか．正準化の提案ではこのようなケースでもボカしやカスれによる誤差をキャンセルでき認識精度向上に役立つとされている．

正準化手法そのものについては上記のようなケースも含め，様々なバリエーションについての効果は実験的にはまだ十分に確認されていないと思われる．しかしながら上記したような明らかな効果があるので，この処理を単純類似度での計算に用いるケースは多いと考えらる．その意味で本研究では正準化についても検討することとした．

なお，ここでは正準化として元の特徴ベクトルの要素値から一定値を減算する方式のみ説明した．オリジナルの提案である一般的な正準化としては，カテゴリをまたがった平均パターンを減算するなど他の手段もある．

4.4.4　内積演算を用いた正準化考慮型ＦＭＬ

ここでは正準化と正規化を両方考慮した系を，入力パターンを z ，辞書ベクトルを ζ として以下のように定める．

$$
\begin{aligned}
&y = \frac{x}{\|x\|}, \quad x = v - \frac{1}{n}(v, \vec{1})\vec{1}, \quad v = \rho(\hat{v}), \quad \hat{v} = Fz, \\
&\psi = \frac{\varphi}{\|\varphi\|}, \quad \varphi = \zeta - \frac{1}{n}(\zeta, \vec{1})\vec{1}, \\
&S = (y, \psi)^2.
\end{aligned} \tag{4.44}
$$

この系で ζ と F の更新式を求めると，

$$\Delta\zeta = \pm 2\varepsilon_r l'(d)\frac{1}{\|\varphi\|}(y, \psi)\{y - (y, \psi)\psi\}, \tag{4.45}$$

$$\Delta F = \pm 2\varepsilon_f l'(d)\rho'(\hat{v})\frac{1}{\|x\|}(y, \psi)\{\psi - (y, \psi)y\}z^T, \tag{4.46}$$

となる．これらは形式的には正準化を考慮しない式（4.41）,（4.42）と同じになる．

このように正準化を考慮する場合も，正規化を考慮する場合と形式的には同じ更新式になる．もちろん正準化の場合には正準化操作が入っているので，更新量を求める実際の計算が異なることは言うまでも無い．

ここでの式の導入に関する詳細も付録 A に記述した．

第5章　フィルタマスク学習の実現

5.1　文字認識への応用

本研究の最終的な目標は文字認識精度の向上にあり，その目的で FML の性能評価と実験的な調査を行なった．しかしながら本来 FML は文字認識だけでなくその他の分野，音声認識や画像認識などに応用することができる．そしてそのような様々な分野に適用するためには，それぞれ固有の認識系および学習システムが必要となる．FML を実際のアプリケーションに組み込む時，核となる学習方式は同じであるけれども，前処理や認識系など周辺部分がその課題ごとに異って来る．文字認識に適用する場合もそれ固有の構成が必要となる．ここでは FML を文字認識に適用するためのシステム構成について説明する．

5.1.1　全体構成

図 5.1 に文字認識に FML を適用した学習システムを示す．本システムは「特徴抽出部」「認識部」「学習部」の３つのパートからなる．特徴抽出部はさらに「１段目フィルタ」「大きさ正規化」「２段目フィルタ」「非線形変換」からなる．

入力される学習パターンは予備的なフィルタリング処理として１段目のフィルタでボカしの処理が行なわれる．ここで若干パターンがボカされているが，これは次の大きさ正規化処理の入力として，２値画像よりも多値画像の方が良いのではないかという考えに基づいている．つまりその方がよりスムースな正規化画像が得られるのではないかということである．なお，大きさ正規化の前にボカしを入れることの効果については未確認で，これはまた別の調査研究の課題と考えている．今回は未確認のまま効果があるのではないかという推定に基づいてボカしの処理を入れている．

次にこのボカされたパターンは大きさ正規化されて，すべて同じ大きさに揃えられる．そしてその後に２段目のフィルタリングで本格的な特徴抽出が行なわれる．この２段目で使われるフィルタマスクが学習対象になる．最後に非線形変換を通して得られる値が特徴ベクトルの要素値として出力される．

こうして作成された特徴ベクトルは認識部に入力され，そこで文字認識が実行される．認識部では文字種（以降，カテゴリと呼ぶ）ごとに辞書ベクトルが設定されていて，この辞書ベクトルと特徴ベクトルの間で相違度（または類似度）が計算され，その結果，相違度が最小となる（または類似度が最大となる）カテゴリが認識結

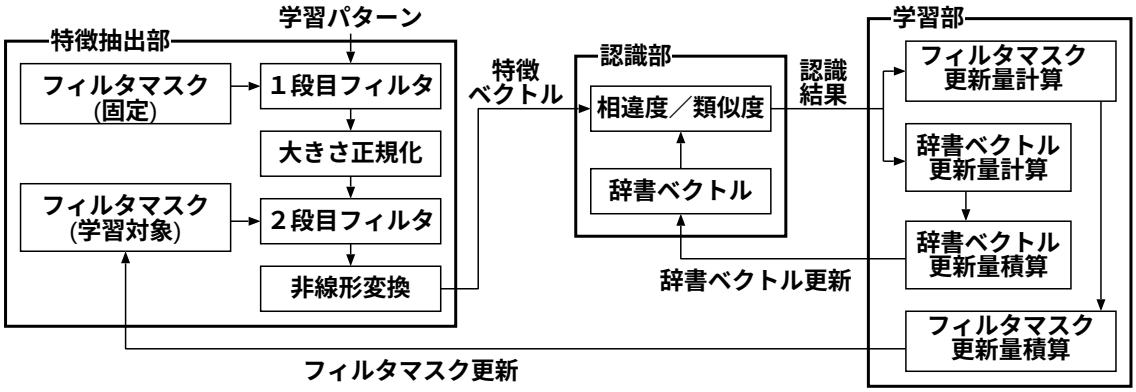


図 5.1: FML を文字認識に適用した場合の学習システム

果として出力される．ここまでが学習を含まない文字認識システムである．

次に学習を含めたシステムとしての動きを説明する．学習を行なう場合，各学習パターンにはその学習パターンの「答え（そのパターンのカテゴリの名称）」が付属して入力される．この答えに基づいて正解カテゴリの中の最小相違度（または最大類似度）と不正解カテゴリの中の最小相違度（または最大類似度）となるカテゴリ名とその相違度（または類似度）が認識結果として出力される．正解とは辞書ベクトルのカテゴリ名と入力パターンのカテゴリが一致する場合のことで，不正解は一致しない場合のことである．この認識結果と，繁雑になるのでその線は図示していないが，入力パターン，特徴ベクトル，辞書ベクトルが学習部に入力される．これらに基づいて辞書ベクトルの更新量とフィルタマスクの更新量が計算される．これらの更新量はすべての学習パターンについて計算されて積算され，それぞれが学習部の中に蓄積される．すべての学習パターンの積算が終了したところで，この積算された更新量がそれぞれフィルタマスクと辞書ベクトルに加算され１回分の学習が終了する．

学習開始にあたって初期フィルタマスク，初期辞書ベクトルが必要となる．これらを作成するためのパートも当然必要となるが図 5.1 には特に図示していない．

認識部ではすべての学習パターンの認識を行なうとともに評価用のパターンの学習も行ない，適宜その認識率が出力される．当然，評価用パターンにも正解カテゴリ名が付属していて，それに基づいて認識率が出される．学習回数を横軸に取り，この認識率をプロットしたグラフを観察することによって学習状況を確認することができる．学習終了はこの観察に基づいて決定される（自動ではなく，目視による決定）．

5.1.2 大きさ正規化

本研究で取り上げる FML では特徴データは数値からなるもので固定長である．ここで固定長とはパターンによらず特徴データの要素数が一定の場合を言い，不定長とはそれがパターンによって変動する場合を言う．一般に文字認識をターゲットとする場合，特徴データとして不定長のものを扱わなければならないという強い要請は無いので，特徴データを固定長，すなわち特徴ベクトルを用いて認識系を構築することがほとんどである．これに対して音声認識では不定長の特徴データを扱うケースが多くあり，実際に確率降下法を音声認識に適用した MCE/GPD は不定長のケースにも対応している．しかしながら不定長の方式を用いるとシステムが複雑になり見通しも悪くなるので，可能であれば固定長の方が扱い易い．特に線形空間の議論に基づく様々なパターン認識研究の成果を利用できることは固定長の大きなメリットである．そのような理由で本研究でも特徴データを固定長としている．

このため本研究の FML では入力された文字画像の大きさを正規化している．すなわち縦横の大きさが同じになるように拡大・縮小処理を施す前処理を，２段目のフィルタリング処理に先だって行なうようにしている．大きさ正規化自体は文字認識技術のいわばスタンダードで様々な手法があるが，本研究では以下に述べる比較的ていねいな，言葉を替えれば処理量の多い方式を導入している．本研究の目的である FML の性能を調査する中で，前処理の悪影響をできるだけ減らす目的でそうしている．

入力パターンの画素番号を (i,j) とし，i を横方向，j を縦方向の番号とする．画像サイズを N,M とし，$i=0,...,N-1$，$j=0,...,M-1$，とする．一方，出力画像のサイズを n,m とする．ここで入力および出力画像の座標系を次のように設定する．入力画像の横軸で言えば左端の位置を 0，右端の位置を N とする．各画素内の位置は，画素の切れ目の位置を整数値として，１画素の長さが 1 になるように，実数値で表す．縦は上を 0 ，下を M とする．出力画像も同様に定義する．

この時，出力画像の画素番号 (i,j) の画素値を以下のようにして決める．まず，

$$
\begin{aligned}
x_{st} &= Ni/n,\ \ x_{en} = N(i+1)/n \\
y_{st} &= Mj/m,\ \ y_{en} = M(j+1)/m
\end{aligned}
\tag{5.1}
$$

のようにして $x_{st}, x_{en}, y_{st}, y_{en}$ を定義する．これは出力画像の画素 (i,j) に対応する入力画像中の矩形領域の座標値である．この矩形領域における黒の濃度を積算し，その値を出力画像の画素値とする．もちろん入力画素の

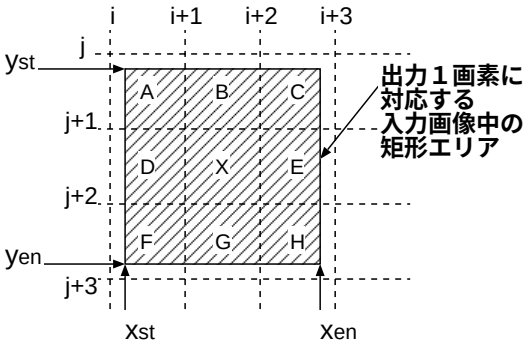


図 5.2: 大きさ正規化における面積計算

１画素が全部矩形領域に含まれる時はその画素値 f を黒の濃度として積算する．一方，入力画素の一部が矩形領域に含まれる時は，（含まれる画素の割合）× f を黒の濃度として積算する．

図 5.2 はこの処理の様子を示したもので，点線が入力画像の画素の境界線，i から $i+3$ 番目の縦線と j から $j+3$ 番目の横線が示されている．ちなみに番号は 0 オリジンでそれぞれの境界線は全部で $N+1$ 個 と $M+1$ 個 ある．点線で囲まれた部分が入力画像の１画素である．一方，出力側の１画素に対応する矩形領域を図の斜線で示している．その矩形領域の入力画像内での座標値が $x_{st}, x_{en}, y_{st}, y_{en}$ で与えられている．出力画素の画素値は斜線部分の中の入力画像の画素値を f とする時，f ×（その部分の面積）を積算したものである．すなわち画素 X については入力画像の画素値が f であった場合に面積 f を積算し，画素 A〜H についてはその画素値が f であった場合に斜線で塗られた部分の面積に f を掛けたものを積算する．画像が全部黒の時の値を１にしたい場合には，矩形領域の面積で上記の黒画素面積を割れば良い．

この方式は画像処理における座標変換の一般的な手段であるが，面積計算をする時にバグを入れることが多いので注意が必要である．入力画像の右端の位置または下側の位置に x_{en}, y_{en} が来る時が問題である．単純にプログラミングすると，この時に存在しない画素すなわち (N, j) や (i, M) にアクセスするようなコードを書いてしまうことが多い．これがバグの原因となる．これに対しては右端や最下位置の画素の計算について，x_{en}, y_{en} から適当な小さい値を引いておく等なんらかの対処が必要である．

このような大きさ正規化の処理により，大きさが同じでグレー・レベルの文字パターンが得られる．この画像を用いて次のフィルタリング処理が行なわれる．

5.1.3 フィルタリング

本方式では最初に述べたようにフィルタリングは２段階で行なっている．大きさ正規化前の１段目は３×３のマスクを用いており入力画像のすべての画素においてフィルタリングを行なっている．すなわち出力画像のサイズと入力画像のサイズは同じである．２段目では７×７のマスクを用いており画像のサイズを（縦または横の長さで）およそ１／３にしている．

本方式の１段目では縮小を行なっていないが，一般に２段のフィルタリング・システムを用いる場合，１段目でも画像のサイズを縮小することが多いので，ここで若干その事について触れておく．たとえば１段目でサイズを（長さで）１／２にしたとする．そうすると１段目のフィルタの出力画素１個分は１段目の入力画像の画素４個分に相当することになる（長さで言えば２個分）．また２段目のマスクの１画素は１段目の出力画像の画素に対応しているので，入力画像の画素で言えば，画素４個分（長さで言えば２個分）に相当している．したがって，この２段のフィルタで，２段目のフィルタの２倍のサイズ（長さ）の領域をカバーしていることになる．

フィルタリングの処理時間はフィルタサイズ（長さ）の自乗すなわちフィルタマスクの画素数に比例する．それ故，このようにフィルタリング処理を２段に分けることでマスクの総画素数を減らすことができ，結果的に処理の高速化をはかることができる．

これを具体的な例で見てみよう．仮に入力画像のサイズを 79×79 とし，１段目が 3×3 のマスク，２段目

が 7×7 のマスクだとする．１段目はサイズを１／２にし，２段目ではサイズを１／３にするものとする．１段目ではサイズは１／２なのでマスクを掛ける回数は入力画像の画素数の１／４となり，具体的には 40×40 回となる．従って，ここでのフィルタマスクの画素の総数は $(3 \times 3) \times (40 \times 40)$ である．２段目はこれをさらに１／３にするのでマスクを掛ける回数は 14×14 回となる．従って２段目でのフィルタマスクの画素の総数は $(7 \times 7) \times (14 \times 14)$ である．この両者を合わせると $24,004$ となる．

一方，上記の２段フィルタに相当する１段フィルタを仮に作ったとすると，１段目で画像サイズが１／２になるので，２段目のマスクサイズの約２倍が１段フィルタのマスクサイズとなる．具体的には 15 である．また，出力の画素数は上記の２段フィルタと同じなので 14×14 となり，そのフィルタマスクの画素の総数は $(15 \times 15) \times (14 \times 14) = 44,100$ となる．

このケースでは２段のフィルタでは１段のものに比べて処理時間がほぼ半分の 54% である．

このような考え方は文字認識のフィルタリング操作では良く使われるものであり，本研究でもその手法を導入することは可能であったが，今回の実験では１段目は大きさ正規化をスムースに行なうことのみを主目的と考えて１段目での画像サイズの縮小は行なわなかった．その理由はより基礎的な構成での実験を最初に行なうべきと考えたためであるが，実用時にはこの種の検討が必要となる．

FML はこの２段目の７×７のフィルタマスクに対して行なう．１段目については固定のフィルタマスク値を用いている．

本方式ではフィルタリング後に非線形変換を導入している．特徴ベクトルの各要素値に非線形変換を入れると効果があることが文字認識の分野では知られている [21]．この文献の例では平方根関数が使われていて必ずしもシグモイドでは無いのであるが，本研究では次の理由でシグモイドを用いた．

一般に特徴データは$-\infty$から$+\infty$までの値を取るが，そのままでは不都合なのでこの値を上限下限でサプレスすることが OCR の設計では良く行なわれる．特に特徴値を整数として処理する系を設計する時にはこの処理は必須である．本研究ではこのサプレス関数に似た連続関数としてシグモイドを用いている．平方根関数を導入する以前に，より基礎的なシグモイドによるシステムを検討すべきであると考え，今回は平方根関数は扱わずに今後の課題とした．

5.1.4 認識部

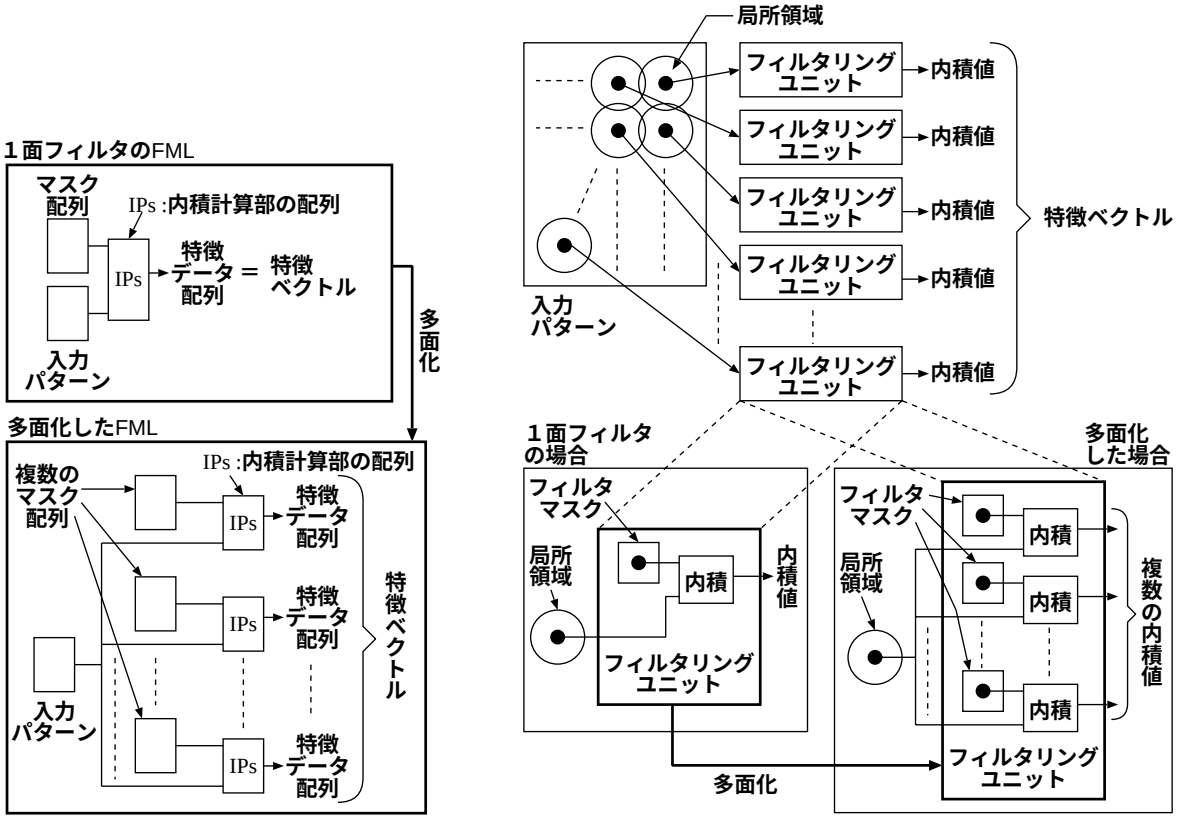
本研究では認識部としてユークリッド距離による識別系と単純類似度によるものと２通りのものを用いている．この認識部は相違度や類似度を計算するための辞書ベクトルを各カテゴリに対応させて持っている．この辞書ベクトルも学習対象の１つである．この認識部は認識シミュレーションの機能を備えていて学習パターンや評価用パターンのすべてを認識してそれらについての認識率を計算して出力することができる．また次の学習部で必要となる正解カテゴリや不正解カテゴリに関するデータ，すなわち相違度最小（類似度最大）となる正解カテゴリ，不正解カテゴリ及び，それぞれの相違度（類似度）を出力する．

システム構築上はこの認識部は次に述べる学習部の中に組み込む形でも作成できるが，本研究では学習なしの認識部のみからなる文字認識システムを容易に構成できるようにすることと，学習システムの組替えを容易にする目的で認識部と学習部とはそれぞれ独立に作成した．

5.1.5 学習部

本システムの学習部にはこれまでに述べて来た FML の機構を導入し，辞書ベクトルとフィルタマスクの値を学習するようにしている．入力は認識部からの認識結果の他，学習パターン・データそのものや，特徴ベクトル，辞書ベクトルなどである．これらに基づいてフィルタマスクの更新量と辞書ベクトルの更新量のそれぞれが計算され，積算されて蓄積される．すべての学習が終ったところで，その積算された更新量を辞書ベクトルとフィルタマスクに加算する．

より正確に言うと，ここへの入力は学習パターンそのものでは無く特徴抽出部の大きさ正規化のフィルタリングの出力，すなわち２段目のフィルタリングの入力データが用いられる．しかしながら大きさ正規化後のデータ



（A）多面化したマスク配列　　（B）多面化のユニット表現

図 5.3: FML 多面化の２つの表現.

サイズはかなり大きく，処理時間的にはそのデータをファイル渡しで学習部に送るよりも学習パターンそのものを学習部に入力するようにして，学習部の中で大きさ正規化と１段目フィルタリングを行なうようにした方が処理時間的に有利な場合がある．実際に本研究では後者の状況となり，そのためファイル渡しの方法は採らなかった.

5.2　FML の高度化

ここでは，これまでに述べた FML の標準的なシステムをさらに高度化するための２つの拡張法について述べる．１つめが「多面化」であり，もう１つが「多重化」である．まず「多面化」について述べる.

5.2.1　多面化による FML の高度化

FML における多面化（MFML: Multiple FML）とは１つの局所領域に対応するフィルタマスクを複数用いる事である．その様子を図 5.3（A）に示す．この図の（A）の上側部分は１面のフィルタマスクしか無い例で，これまでに説明してきた形式である．この例では局所領域が縦横に並んだ入力パターンを１つの矩形で表現していて，それを「入力パターン」と記述している．一方，その局所領域に対応してマスクを縦横に並べたものをやはり１つ矩形で表現していて，それを「マスク配列」と記述している.「IPs」は内積演算がやはり縦横に並んだ配列で，上記の局所領域の画素値とフィルタマスクのマスク値の内積を計算し，それを特徴データとして出力している．特徴データはマスクの数と同数あってマスク配列に対応しているので，その意味で「特徴データ配列」と記述しており，これが特徴ベクトルとなる.

この１面フィルタの形式を多面化した図が図 5.3（A）の下の部分である．多面化では１つの局所領域に対し

て複数のフィルタマスクが対応付くのであるが，この図では複数のマスク配列を用意して，これらと１つの入力パターンとの間で特徴データ配列を計算出力するように記述している．複数の特徴データ配列はまとめられて１つの特徴ベクトルとして出力される．したがって n 個の配列がある時の特徴ベクトルの次元は１面フィルタにおける次元の n 倍になる．

この方法を別の表現法で表したものが図 5.3（B）である．本稿ではこの表現をユニット表現と呼ぶことにする．この図の上の部分は FML の一般形式をユニット表現で図示したものである．入力パターン上で縦横に並んだ局所領域の画素それぞれは，その局所領域に対応して用意してあるフィルタリングユニットに入力される．フィルタリングユニットは，その局所領域の画素に基づいてフィルタリングの計算を行ない結果を出力している．図ではこの出力を内積値として記述しており，それらをまとめたものが特徴ベクトルになる．図ではフィルタリングユニットを縦１列に並べたが，これも縦横に並んでいると考えて良く，その意味でこれをユニット配列と呼ぶこととする．

この表現において１つのフィルタリングユニットの中身を書き表したのがその下の図で，左側が１面フィルタの場合で右が多面化した場合のものである．

１面フィルタの場合にはフィルタリングユニットには１つのフィルタマスクが保持されていて，１つの局所領域の画素が入力されるとこれとマスク値との間で内積計算が施され内積値が出力される．

一方，多面化した場合にはフィルタリングユニット内には複数のフィルタマスクが保持されていて，１つの局所領域の画素が入力されると，これとそれぞれのマスク値との間で内積計算が施され，それら複数の内積値が出力される．

ユニット表現は多面化した複数のマスクについて，同じ局所領域に対応付くもの同士を関連付けて見る場合に適している．一方，図 5.3（A）に示すマスク配列による表現はマスク配列をパターンとして見るのに適している．本稿ではマスク配列をパターンとして見ることが多いので多くの場合マスク配列による表現を使う．

このフィルタマスクを複数枚使う考え方は従来から良く用いられる考え方である．もっとも良く使われる考え方は文字線の方向性を出すために，文字画像の局所部分における方向性を抽出するフィルタを複数方向分用意する考え方である．メキシカンハット型のフィルタやガボールフィルタなどがそうである．

また解像度の異るフィルタマスクを用いる考え方もある．一般にはフィルタリングを行なう時に画素を間引いて解像度を落すことが普通に行なわれるが，この考え方はこの間引きの度合を変えたフィルタリングを複数，同じ場所で行なう考え方である．これは厳密にはフィルタマスクを複数枚持つのとは異るが，複数のフィルタリングを行なってそれぞれの特徴ベクトルを合体して大きな特徴ベクトルにすると言う意味では同じ考え方である．

このように多面化自体は新規な考え方では無いが，それが FML の枠組の中でどうなるのか，どう有効なのかを調査する事がこの拡張を行なう目的である．

ここで多面化した時の辞書ベクトル更新式およびフィルタマスク更新式を求めておく．これらは基本的に１面の時と同一であるが多面化するために若干サフィックスが変わる．

局所領域を i ，面番号を j とする時，そのマスクを $f^{(i,j)}$ と書き，これと局所領域 i の画素からなるベクトル z_i とにより，各マスクとの内積値を $f^{(i,j)T}z_i$ と記述する．この内積値を $\hat{x}_{ij}$ と書いて，$x_{ij} = \rho(\hat{x}_{ij})$ の変換により特徴ベクトルの要素値を求める．辞書ベクトルの要素値を φ_{ij} とする時，相違度は，

$$S = \sum_{ij}(x_{ij} - \varphi_{ij})^2 \tag{5.2}$$

となる．また特徴ベクトルは $x = (x_{ij})$ ，辞書ベクトルは $\varphi = (\varphi_{ij})$ と書く．この記法によってフィルタマスク更新量は次のようになる．

$$\Delta f^{(ij)} = \mp 2\varepsilon_f l'(d)\rho'(\hat{x}_{ij})(x_{ij} - \varphi_{ij})z^{(ij)}. \tag{5.3}$$

なお辞書ベクトルの更新式は式（4.15）と同じである．

5.2.2 多面化 FML における学習の対称性の問題

FML を多面化する場合，各マスクの値が初期値において同じであると，その後の学習においてそれぞれのマスクの更新量の計算の状況はすべて同じなので，結果としてマスクの更新量はすべて同一となる．すなわち，それぞれのマスクはまったく同じように学習される．これを学習の対称性と呼ぶことにする．この状況だと学習が進んでも得られるマスクはどれも同じということになってしまい，これでは多面化する意味が無くなる．従ってこれを避ける必要があるが，そのための手段についてここで若干の説明をしておく．

例えば学習時にランダムノイズをマスク値に加えるとか，あるいは学習強度を決めるパラメータや，窓関数のパラメータなどをマスクによって変えるなどの対策が考えられるが，以下，本研究で試した２つの手段について述べておく．

１つは初期値として異るマスクを用いる考え方である．極めて単純で妥当な考え方であるが初期値をどうするかは考えなくてはならない．

もう１つはディレイド・スタートという方式である．この方式ではマスクの初期値はすべて同じ値とする．しかしながら複数マスクの学習を同時に行なわず最初は１つのマスクのみ学習する．ある程度学習の進んだところで２つめのマスクの学習を開始する．もちろん，１つめのマスクの学習は継続している．この２個のフィルタの学習がある程度進んだところで３番目のマスクの学習をスタートさせる．以下同様にこのような処理を順次行なう．

以上は学習に非対称性を持たせる２つの手法であるが，それでも各マスクに対する更新は似通ったものになって，出来上がったマスクは冗長性の大きなものになってしまうという懸念が残る．このような懸念に対しては相互抑制の考え方を導入することができる．

相互抑制の考え方の一例は，例えば A，B の２つのマスクがあった時，A の更新時に本来の A の更新に加えて（B ×微小な係数）を減算し，逆に B の更新時に B の本来の更新に加えて（A ×微小な係数）を減算するような処理である．これによって A と B は似ていない方向に学習が進むことになる．相互抑制の考え方には他にも様々な手段が考えられるが，いずれにしても多面化したマスクの中で，マスク間に同じ要素が無くなる方向に学習を進める手法である．次にその一例として直交化の手法を説明する．

5.2.3 多面化 FML における直交化の手法

直交化とは，できるだけ各マスク同士のなす角度が直交するように制御して学習させる考え方である．各マスク同士のなす角度とはマスク値を並べたものをベクトルと見なした時の角度である．このための更新式を決定するために，あらたな認識精度評価関数を設定し，それに基づいて更新式を求めることとする．

もともとの認識精度評価関数を LF ，局所領域 k の i 番目のマスクを $f^{(ik)}$ ，マスク面数を s として，直交性を考慮した認識精度評価値 J を以下のように定める．

$$J = LF + J_0, \quad J_0 = \mu \sum_k \sum_{i=0}^{s-2} \sum_{j=i+1}^{s-1} \frac{(f^{(ik)}, f^{(jk)})^2}{\|f^{(ik)}\|^2 \|f^{(jk)}\|^2}. \tag{5.4}$$

この式は各マスクのなす角のコサインの自乗の和を μ 倍して評価値に加えたものである．これによって μ の重みに応じた直交化が学習時に行なわれることになる．

この式を $f^{(ik)}$ の要素値 $f_p^{(ik)}$ で偏微分することによって更新量を求める．なお以下では LF の部分の偏微分は省略する．

$$\begin{aligned}
\frac{\partial J_0}{\partial f_p^{(ik)}} &= \mu \sum_{i \neq j} \{ \frac{\partial}{\partial f_p^{(ik)}} \frac{(f^{(ik)}, f^{(jk)})^2}{\|f^{(ik)}\|^2 \|f^{(jk)}\|^2} \}, \\
&= \mu \sum_{i \neq j} \{ \frac{2(f^{(ik)}, f^{(jk)}) f_p^{(jk)}}{\|f^{(ik)}\|^2 \|f^{(jk)}\|^2} + \frac{(f^{(ik)}, f^{(jk)})^2}{\|f^{(jk)}\|^2} (-\frac{1}{\|f^{(ik)}\|^4})(2 f_p^{(ik)}) \}
\end{aligned}$$

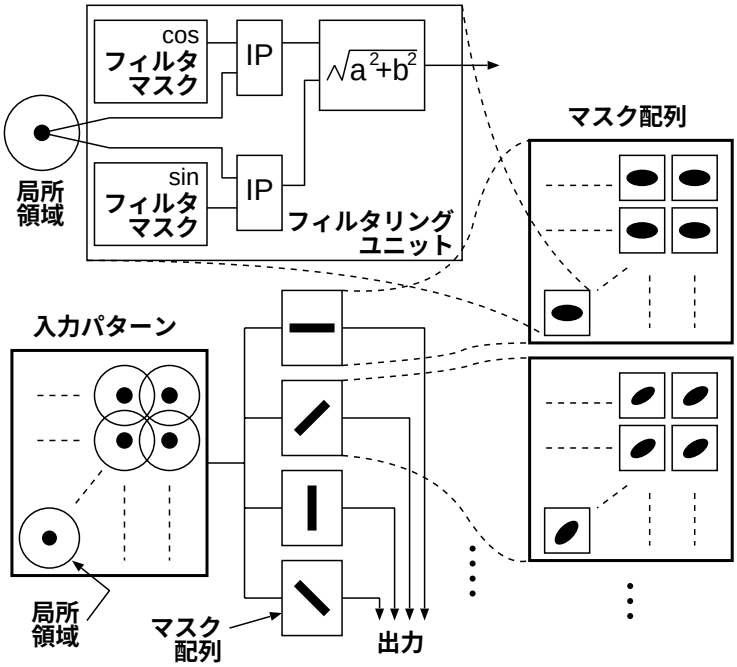


図 5.4: 多重化した FML の構造の例

$$
\begin{aligned}
&= \mu \sum_{i \neq j} \frac{2(f^{(ik)}, f^{(jk)})}{\|f^{(ik)}\|^2 \|f^{(jk)}\|} \{ \frac{f_p^{(jk)}}{\|f^{(jk)}\|} + \frac{(f^{(ik)}, f^{(jk)})}{\|f^{(jk)}\|} (-\frac{1}{\|f^{(ik)}\|^2})(f_p^{(ik)}) \} \\
&= 2\mu \sum_{i \neq j} \frac{1}{\|(f^{(ik)}\|} \frac{(f^{(ik)}, f^{(jk)})}{\|f^{(ik)}\| \|f^{(jk)}\|} \{ \frac{f_p^{(jk)}}{\|f^{(jk)}\|} - \frac{(f^{(ik)}, f^{(jk)})}{\|f^{(ik)}\| \|f^{(jk)}\|} \frac{f_p^{(ik)}}{\|f^{(ik)}\|} \}. \qquad (5.5)
\end{aligned}
$$

ここで,

$$
P = \frac{(f^{(ik)}, f^{(jk)})}{\|f^{(ik)}\| \|f^{(jk)}\|}, \qquad (5.6)
$$

と置くことで,

$$
\frac{\partial J_0}{\partial f_p^{(ik)}} = 2\mu \sum_{i \neq j} \frac{P}{\|f^{(ik)}\|} \{ \frac{f_p^{(jk)}}{\|f^{(jk)}\|} - P \frac{f_p^{(ik)}}{\|f^{(ik)}\|} \}, \qquad (5.7)
$$

となる．従ってこれに関する更新量は,

$$
\Delta f_p^{(ik)} = -2\varepsilon_f \mu \sum_{i \neq j} \frac{P}{\|f^{(ik)}\|} \{ \frac{f_p^{(jk)}}{\|f^{(jk)}\|} - P \frac{f_p^{(ik)}}{\|f^{(ik)}\|} \}, \qquad (5.8)
$$

であり，通常の更新に加えてこれによる更新を行なえば良い．これによって直交化を加味した多面化 FML が可能となる．

5.2.4 多重化による FML の高度化

ここではもう１つの拡張法である「多重化」について説明する．図 5.4 はその例を示したものである．この手法では新たにフィルタリングユニットを多面化の時とは別の形で導入する．ここでのフィルタリングユニットは図に示す通り，複数のマスク,（cos フィルタマスク）と（sin フィルタマスク）と非線形関数 ($\sqrt{(a^2+b^2)}$) を有している．１つのフィルタリングユニットは１つの局所領域に対応しており，その局所領域の画素値からなるベ

クトルと各マスク値からなるベクトルの間で内積計算（図の IP）を行ない，その結果得られる２つの数値の自乗和のルートがフィルタリングの結果として出力される．

このフィルタリングユニットは縦横に並んでマスク配列を構成している．さらにこれが多面化されて，この図の例では縦方向線分抽出機能を持ったものに続いて，斜め右上，横，斜め右下の４つの面を持つものとして表現されている．各マスク配列の出力は合体されて最終的に特徴ベクトルとして出力される．以降このマスク配列の番号を面番号と呼ぶことにする．この例では，0 から 3 である．

なお，ここでのフィルタリングユニットの使い方は多面化の時と異なるが，分かり易さのためにそうしている．多面化の所ではフィルタリングユニットは１つの局所領域に対応付くものとしていたが，ここでは１の出力に対応付くものとしている．

この多重化と多面化の違いは，多重化ではフィルタリングユニットに複数のマスクを持ち，それらの出力を非線形関数を通して１つにまとめ１個の数値として出力しているところにある．

次にこの場合の更新式を求めよう．

まずフィルタリングの式を以下のように定義する．局所領域を i ，面番号を j ，フィルタリング・ユニット内のマスク番号を k とする時，そのマスクを $f^{(ijk)}$ と書き，局所領域 i の画素からなるベクトルを z_i と書く．すると各マスクとの内積値は $f^{(ijk)T}z_i$ となる．ここでフィルタユニットのマスク数を t としてフィルタリングユニット内の非線形関数を $h(P_0, P_1, ..., P_{t-1})$ と定義すると，フィルタリングユニットの出力は

$$\hat{x}_{ij} = h(P_0, P_1, ..., P_{t-1}), \quad P_k = f^{(ijk)T}z_i, \tag{5.9}$$

となる．この値を $x_{ij} = \rho(\hat{x}_{ij})$ によって変換し特徴ベクトルの要素値とする．辞書ベクトルの要素値を φ_{ij} とする時，相違度は，

$$S = \sum_{ij}(x_{ij} - \varphi_{ij})^2, \tag{5.10}$$

となる．また特徴ベクトルは $x = (x_{ij})$ ，辞書ベクトルは $\varphi = (\varphi_{ij})$ と書く．この記法によって各更新量は次のようになる．

$$\begin{aligned}
\frac{\partial S}{\partial f_p^{(ijk)}} &= 2(x_{ij} - \varphi_{ij})\frac{\partial x_{ij}}{\partial f_p^{(ijk)}} \\
&= 2(x_{ij} - \varphi_{ij})\rho'(\hat{x}_{ij})\frac{\partial \hat{x}_{ij}}{\partial f_p^{(ijk)}} \\
&= 2(x_{ij} - \varphi_{ij})\rho'(\hat{x}_{ij})h^{(k)}(P_0, P_1, .., P_{t-1})\frac{\partial P_k}{\partial f_p^{(ijk)}} \\
&= 2(x_{ij} - \varphi_{ij})\rho'(\hat{x}_{ij})h^{(k)}(P_0, P_1, .., P_{t-1})z_{ij}.
\end{aligned} \tag{5.11}$$

ここで， $h^{(k)}(P_0, P_1, .., P_{t-1})$ は $h(P_0, P_1, .., P_{t-1})$ に対する P_k による偏微分である．これによりフィルタマスクの更新式は，

$$\Delta f^{(ijk)} = \mp 2\varepsilon_f l'(d)\rho'(\hat{x}_{ij})h^{(k)}(P_0, P_1, .., P_{t-1})(x_{ij} - \varphi_{ij})z^{(ij)}, \tag{5.12}$$

となる．なお辞書ベクトルの更新式は式（4.15）と同じである．

次に図の例のように $P_0 = a$，$P_1 = b$ として平方根関数を用いた場合の更新式を計算しておく．

$$\hat{x}_{ij} = \sqrt{a^2 + b^2}, \tag{5.13}$$

なので，

$$\frac{\partial \hat{x}_{ij}}{\partial a} = \frac{a}{\sqrt{a^2 + b^2}} = \frac{a}{\hat{x}_{ij}}, \tag{5.14}$$

となり，

$$\Delta f^{(ija)} = \mp 2\varepsilon_f l'(d)\rho'(\hat{x}_{ij})(a/\hat{x}_{ij})(x_{ij} - \varphi_{ij})z^{(ij)}, \tag{5.15}$$

となる．b についても同様である．これは，式（4.16）に $(a/\hat{x}_{ij})$ を挿入した形である．

上記に示したように本研究における FML の多重化では，フィルタリングユニットの具体例として 2 つのマスクと平方根関数からなるものを導入した．この関数はガボールフィルタに適合するように決めたものである．実際にガボールでは式（5.13）を用いており，その意味ではガボールは本方式でのフィルタリングユニットの一例となっている．また非線形関数として平方根関数が使われているが，この非線形関数がこのフィルタリングユニットの重要な性質となっている．これが線形だと学習全体は基本構成のものと同じになってしまう．この方式をここでは多重フィルタマスク学習と呼び CFML（Compound FML） と略記する．

本研究で用いた CFML はガボールを学習対象としているので，これは FML の拡張というだけでなくガボールの学習方式としても解釈できる．そのようなガボールの学習方式の例としては文献 [15] がある．なお，この先行研究については学習対象がマスクでは無く数式パラメータである点，学習方法，ターゲット，構成などが本研究で扱う方式と異っている．

5.3 FML の実装

5.3.1 特徴抽出部の実装

前の節で述べたように特徴抽出部の 1 段目のフィルタでは 3×3 のマスクを用いており，間引きなし，すなわち入力画像と出力画像のサイズは不変である．このフィルタの重み係数は入力パターンを若干ボカすものでガウス関数から求めたものである．また大きさ正規化については出力サイズを 40×40 にしている．次に 2 段目のフィルタリング処理について述べる．

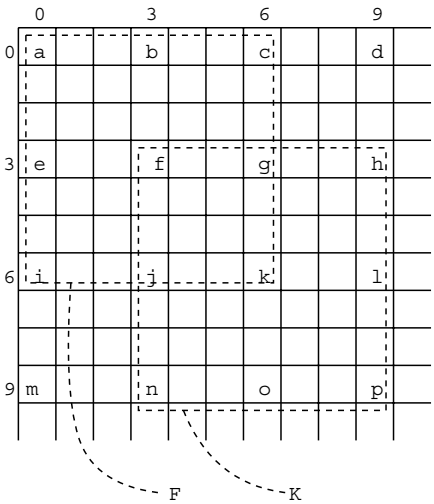


図 5.5: フィルタサイズの配置（左上部分）

図 5.5 はフィルタの配置の左上部分を示すもので図の格子で囲まれた矩形が画素を表す．フィルタはこの図で a〜p の画素位置を中心にしたフィルタサイズ 7×7 のものを用いる．点線で表示した矩形がフィルタの大きさを示すもので，f の位置のフィルタが F，k の位置のフィルタが K である．この図で示すように正規化画像のたとえば x 軸方向の画素位置を 0 から 39 とした時に，0 から 39 の位置まで 3 個目ごとに, y 方向も同様に 0 から 39 の位置まで 3 個目ごとにフィルタを設置していく．画像からはみ出す部分は入力画像の画素値を 0 として計算している．この結果 14×14 の特徴ベクトルが得られる．

フィルタマスクの初期値はガウス関数から求めた総和が 1 となる値である．

フィルタリングによって得られた特徴値は次の非線形変換によってサプレスした．

$$\rho(x) = 1/(1 + e^{(b-ax)}). \tag{5.16}$$

ここで a, b を $a = 9.4, b = 2$ としたが，これはこの関数の入力が実際の文字パターンの例で 0 〜 0.42 の時，出力が 0.12 〜 0.88 となるように設定したものである．

5.3.2 認識部の実装

認識部としてはユークリッド距離計算に基づくものと，単純類似度に基づくものを両方実行できる文字認識シミュレーションソフトを作成した．このソフトでは認識結果としての認識率の出力の他，次の学習に使用するための認識結果，すなわち各パターンごとの ID 番号，答えコード，正解カテゴリの最小相違度（最大類似度），不正解カテゴリの最小相違度（最大類似度）を出力する．

5.3.3 学習部の実装

学習部では入力パターン，特徴ベクトル，認識結果を読み込み，各学習パターンに対する辞書ベクトル更新量の計算とフィルタマスクの更新量の計算を行ない，それぞれを積算し蓄積する．全学習パターンに対する処理が終ったところで辞書ベクトルとフィルタマスクの更新を行なう．

なおここで用いられている損失関数はシグモイドであり，これとその微分すなわち窓関数は次式で表される．

$$l(d) = 1/(1 + e^{-\delta d}), \quad l'(d) = \delta l(d)(1 - l(d)). \tag{5.17}$$

学習で用いたパラメータには数種類あるが重要なのは式（4.15）すなわち辞書学習での ε，式（4.16）すなわちフィルタ学習での ε，式（5.17）における δ である．これらの値は予備実験として各パラメータ値を変えながら簡単な実験を試行しつつ，その実験結果を参考に決めたものである．辞書学習での ε を ε_r，フィルタ学習での ε を ε_f とした時，必ずしも $\varepsilon_f = \varepsilon_r$ ではうまく行かない．これは辞書パラメータによる式（4.5）の偏微分値とフィルタのパラメータによる偏微分値とで，その数値の桁数（オーダー）が違うことがあるからである．これについては第 6 章の中で詳しく述べる．

高速化のため，$l(d)(1 - l(d))$ の部分が 0.01 以下の場合は学習せずにスキップするようにした．また実験では微分した関数の頭に付いている δ は式（4.15）や式（4.16）における ε の中に含めるようにしている．

前にも述べたように各更新量は全学習パターンに対して積算され，その後，一括して更新操作が行なわれる．本研究の実験において一括方式を採用する理由は処理速度を上げる目的にある．この一括処理と 1 パターンごとに更新する方式との差については実験していないが同程度と推定している．しかしながら厳密にはこの違いの調査も必要であり，それは今後の課題と考えている．

なお，これも前述したが本報告における漸化式のほとんどは 1 パターンごとの更新形式で記述されており，その点は実験と同じで無いことを注意しておく．

第6章　文字認識における学習実験

ここでは文字認識をターゲットとする FML の学習実験について報告する．この実験では，まず基礎的な実験としてユークリッド距離に基づく FML の実験を行ない，引続きその拡張として多面化 FML，多重化 FML の実験を実施した．その実験の認識対象は，
（1）類似文字ペア「ボ」と「ポ」，同じく「間」と「問」の２つのペア
（2）「門構え」を持つ漢字からなる類似文字グループ
の２種類である．類似性の強い文字セットとして（1），類似性が比較的弱い文字セットとして（2）をとり上げた．

この実験結果から分かったことを概略以下にまとめておく．まず基礎的な FML での実験結果からは FML に学習効果があること確認できた．さらに「多面化」,「多重化」の拡張によって順次，認識精度が向上して行くことも確認した．多重化 FML に関しては実用レベルにある従来方式と比べても遜色無いレベルにあることが確認でき，類似文字識別に限定すれば FML は「高精度文字認識」構築のための部品として十分な能力を持っているということが分かった．特に「ボポ」,「間問」のような類似性の強い文字セットに関しては従来方式よりも優位性があり十分な実用性を確認することができた．またこの FML を一層改良することによって，さらに良いものにできる可能性についても見通しが得られた．その意味で類似文字認識レベルを越えて一般的な文字認識レベルでも相当な実力を発揮できると考えられる．

以下その実験に関して詳細を述べる．

6.1　実験環境

6.1.1　実験システム

実験システムはすべてソフトウェアで構成した．コーディングはすべてＣ言語でＯＳは linux である．処理の流れについては全体構成を FML の構造に合わせ，ごく自然な形でシステムを作成した．高速化のためにフィルタリング部と学習のコア部分に特殊なコーディングを用いたが，それ以外の部分については全体的に分かりやすさを優先して開発を行なった．

FML を構成する「特徴抽出部」「認識部」「学習部」に対応させ，それぞれ独立のモジュールとして「特徴抽出プログラム」「認識プログラム」「学習プログラム」を開発した．そして学習全体の流れをシェル・スクリプトで記述し学習シミュレーションに柔軟性を持たせるようにした．

図 6.1 にその概要を示す．なお図 6.1 に示すもの以外にもフィルタマスクの初期化ツールや実験サポートツールなど，様々なソフトを開発しているが詳細はここでは省略する．

図 6.1 に示すように文字パターンには学習用とテスト用があり，それぞれファイルになっている．これらを入力として特徴抽出プログラムは特徴ベクトルを作成し，それぞれファイルとして出力する．この特徴抽出処理ではフィルタマスクが格納されたファイルが使用される．

認識プログラムはこれらの特徴ベクトルのファイルと辞書ベクトルの格納されたファイルを読み込み，認識実験をそれぞれの特徴ベクトルに対して行なう．認識実験の結果，得られる認識性能（誤読率）はテキストデータとしてファイルに出力される．学習回数ごとに得られるこのデータは蓄積されグラフ表示などに使われる．

学習用パターンに関する認識結果のデータはファイルに出力され，これが次の段の学習プログラムで使用される．この学習プログラムは全学習パターンを用いた学習の結果として，辞書ベクトルのファイルとフィルタマスクのファイルを更新し，１回の学習を終了する．全体としてはこの処理の流れを繰り返し実行することで逐次学習を進めて行く．

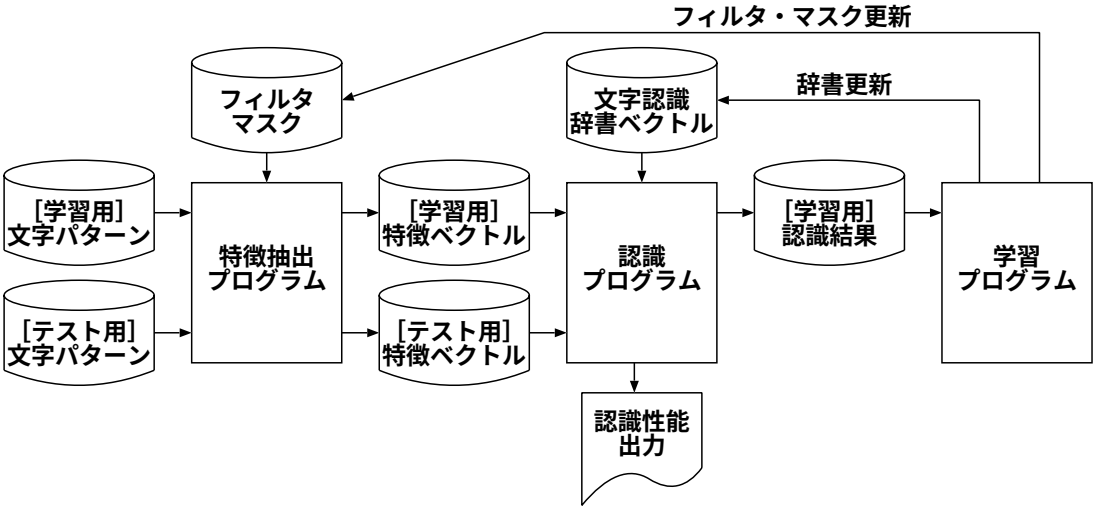


図 6.1: 実験システム

使用した計算機は PenIV 2.8G ～ 3.4G 程度のＣＰＵを持つＰＣで，学習時間については１回の学習に数秒～数十秒程度であった．なお実際の処理時間にについては実験結果のいくつかについて，その節の中で報告している．

6.1.2　実験で用いた文字パターン・データ

実験で用いた文字パターン・セットは次の２種類である．

（１）手書き類似文字ペア（「ボ」と「ポ」のペア,「間」と「問」のペア）

「ボポ」学習用：2,089 文字，評価用：2,017 文字

「間問」学習用：2,000 文字，評価用：1,300 文字

（２）手書き類似文字グループ（門構え）

学習用：16,000 文字，評価用：10,476 文字

それぞれの文字パターンの例を図 6.2（A）,（B）に示す．

これらの文字パターンはそれぞれ今回の実験のために手書で記入したものを２値化イメージで収集したものである．手書の品質としては通常の丁寧さとし，申込書などに記入する際の丁寧さを基準に記入してもらった．

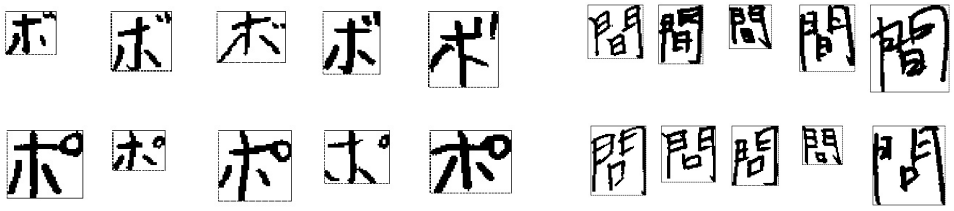
類似文字ペアのセットは，両パターンの間で異る部分が非常に小さいエリアに限定されているようなものである．「ボ」「ポ」は右上部分,「間」「問」は中央下部が異るだけで他は同じパターンのペアである．

類似文字グループは複数個の文字からなるグループで，どの文字もその一部分に共通のパターンを含んだものである．今回の実験ではすべての文字が「門構え」を含む文字のグループを採用した．

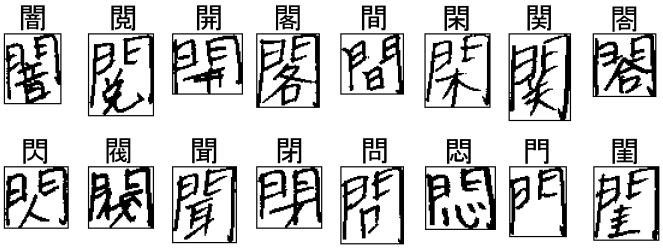
これら類似文字ペアや類似文字グループは，FML がターゲットとする類似文字識別の能力を評価するために選択したものである．

6.2　ユークリッド距離に基づく FML の学習実験

ここでは FML の基本的な形であるユークリッド距離に基づく形式のものについて実験した結果を報告する．実験は準備したパターン・セットすべてについて行なった．以下それらについて順に説明していくが，まず各パターン・セットの実験で共通に用いられた初期フィルタマスクについて説明する．



（A）



（B）

図 6.2: 実験で用いたパターン：　（A）「ボ」「ポ」,「間」「問」,　（B）「門構え」

6.2.1　実験で用いた初期フィルタマスク

図 6.3（B）は初期フィルタマスク，すなわち学習前のマスクの様子を示す図で縦横に 14 個づつマスクが配列されており，各マスクは総て同じものである．その各マスクは 7×7 の画素で構成されており，その画素の値の大きなものを濃く，マイナスの大きいものを薄く表している．値がゼロの時の濃さは図 6.3（B）の各マスクにおける周辺部分の濃度である．この図の（A）は（B）における 1 個のマスクを見易くするために拡大したもので，明るさを調整して濃い目に表示している．

この初期マスクの重み係数は式（6.1）のガウス関数で求める．式（6.1）において A はフィルタの強度を決める係数，x, y はフィルタの中心位置からそれぞれ x, y 方向の画素位置，f はその時の画素の値である．

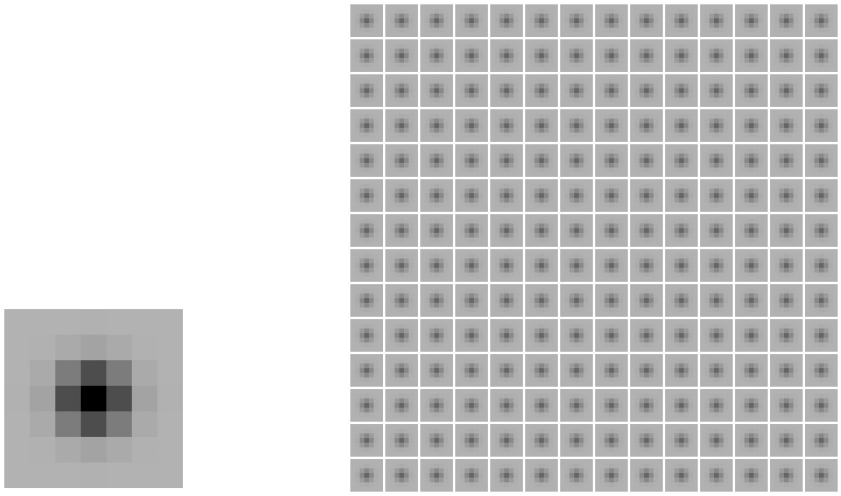
$$f = Aexp\{-\frac{1}{2}(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2})\}. \tag{6.1}$$

実験で用いた式（6.1）のパラメータは $\sigma_x^2 = 0.8$，$\sigma_y^2 = 0.8$，$A = 0.19896$ である．なお，この実験では A をフィルタマスク値の合計が 1 になるように決めた．実際の値を以下に示す．

0.00000	0.00006	0.00038	0.00072	0.00038	0.00006	0.00000
0.00006	0.00134	0.00874	0.01633	0.00874	0.00134	0.00006
0.00038	0.00874	0.05700	0.10650	0.05700	0.00874	0.00038
0.00072	0.01633	0.10650	0.19896	0.10650	0.01633	0.00072
0.00038	0.00874	0.05700	0.10650	0.05700	0.00874	0.00038
0.00006	0.00134	0.00874	0.01633	0.00874	0.00134	0.00006
0.00000	0.00006	0.00038	0.00072	0.00038	0.00006	0.00000

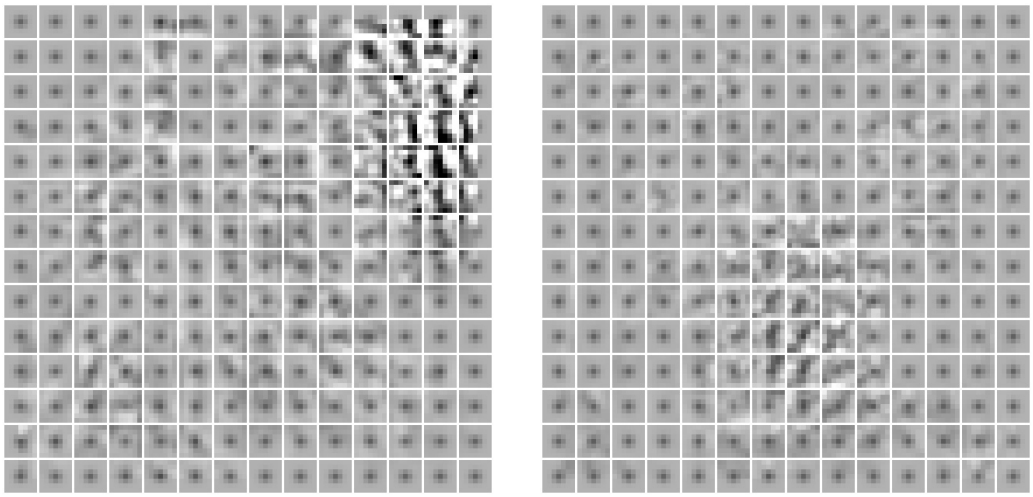
6.2.2　類似文字ペア「ボポ」を対象とする FML の学習実験

最初に類似文字ペアの手書カタカナ「ボ」,「ポ」の識別に本方式を適用した例について報告する．



（A）　　（B）

図 6.3: （A）ガウス関数型初期マスクの拡大表示　　（B）ガウス関数型初期マスク配列



（A）　　（B）

図 6.4: （A）「ボポ」学習後マスク配列　　（B）「間問」学習後マスク配列

図 6.3（B）が学習前のマスク配列，すなわち初期マスク配列の様子を示す図である．図 6.4（A） は学習後のマスク配列である．

この図から「ボ」と「ポ」の違う部分，すなわち右上部分にマスクの初期値とは異なるマスクが形成されていることが分かる．できたマスクは縦線抽出型のものがほとんどである．それ以外の部分については右上部分に比べて変化は少ない．この結果からこの学習方式がパターンの違いを強調するマスクを形成することが確認できる．

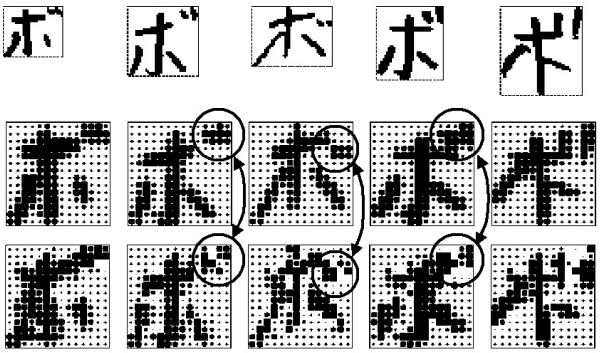


図 6.5: 「ボ」（上段）原パターン　（中段）特徴ベクトル／学習前　（下段）特徴ベクトル／学習後

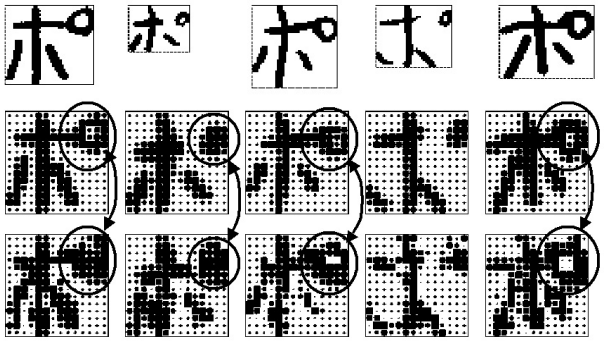


図 6.6: 「ポ」（上段）原パターン　（中段）特徴ベクトル／学習前　（下段）特徴ベクトル／学習後

図 6.5，図 6.6 は「ボ」「ポ」について上から順に，原パターン，学習前のマスクで得られた特徴ベクトル，学習後のマスクで得られた特徴ベクトルを表示したものである．なお，この特徴ベクトルの表示では１画素の濃度を黒丸の大きさで表現している．

学習前のパターンと学習後のパターンを比べると学習後は全体的にノイジーであり，また，あたかもハイパスフィルタを通したかのように，つまり高域が強調されているように見える．

個々のパターンで見ると例えば「ボ」の２，３，４個目のパターンで，学習前では２つの点が接触して１つになっているのに対して，学習後では分離していることが分かる（丸で囲んで示す）．また「ポ」については１，２，３，５個目のパターンで，学習後の半濁点が学習前よりも大きくなっていることが分かる．穴の大きさについては潰れて無くなってしまったもの，大きくなったもの，小さくなったものなど様々であった．いずれにしても半濁点部分が強調されているように見える（丸で囲んで示す）．

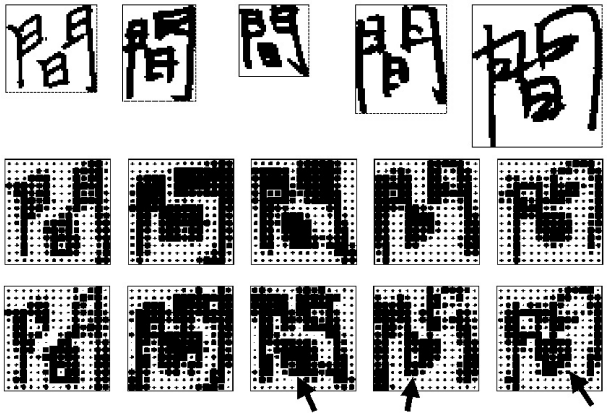


図 6.7: 「間」（上段）原パターン　（中段）特徴ベクトル／学習前　（下段）特徴ベクトル／学習後

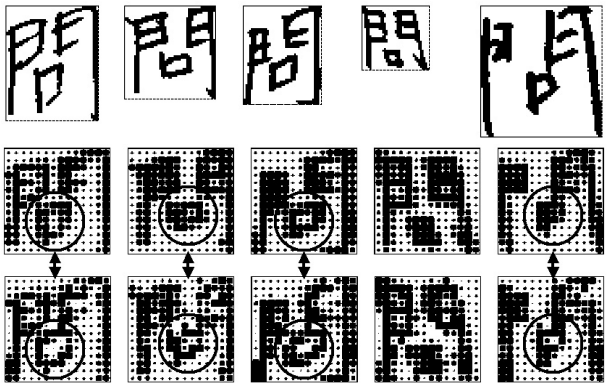


図 6.8: 「問」（上段）原パターン　（中段）特徴ベクトル／学習前　（下段）特徴ベクトル／学習後

6.2.3 類似文字ペア「間問」を対象とする FML の学習実験

次に手書漢字「間」と「問」の識別についての実験を行った．

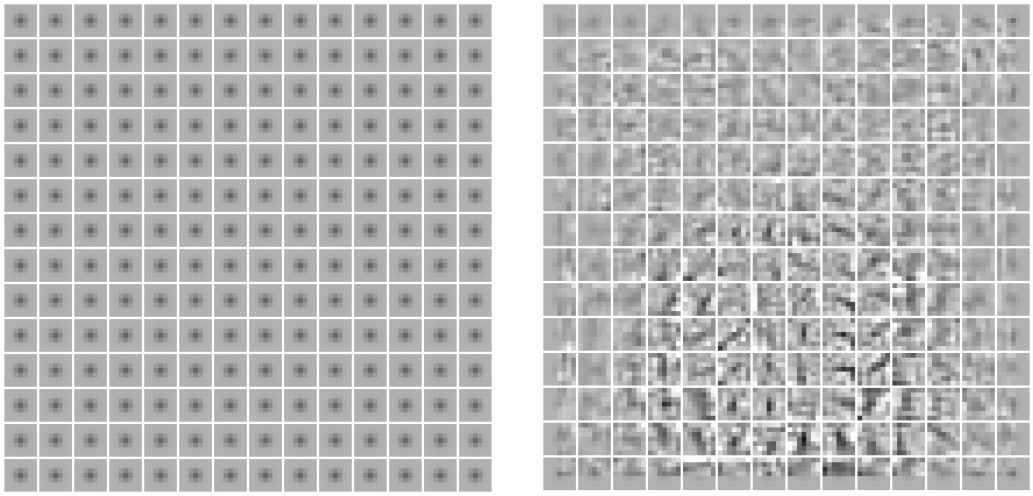
図 6.3（B）が学習前のマスク配列，すなわち初期マスク配列の様子を示す図である．図 6.4（B）が学習後のマスク配列の様子を示す図である．

この図から「間」と「問」の違う部分，すなわち中央やや下部分にマスクの初期値とは異なるマスクが形成されていることが分かる．この部分では右上から左下，あるいは垂直方向の線分抽出フィルタ，あるいは孤立点や端点抽出フィルタらしきものが生成されている.「間」の「日」の真中の棒の端点を検出するような機能が生成されているのではないかと思われる．

図 6.7，図 6.8 は「ボポ」と同様でそれぞれ原パターン，学習前の特徴ベクトル，学習後の特徴ベクトルを表示したものである．

こちらのケースでも学習後の特徴ベクトルはノイジーで特徴フィルタは高域強調的な性格である．また「間問」の違っている部分，すなわち中央下部については学習前よりも強調されているように見える．

個々のパターンで見ると例えば「間」の３，４，５個目のパターンで学習前では「日」の部分がまったく潰れてしまっているが，学習後ではわずかかながら若干の穴があいていることが分かる（矢印で示す）．また「問」の１，２，３，５個目のパターンについて見ると，学習前のパターンよりも学習後のパターンの方が「口」の穴が大きくなっていることが分かる（丸で囲んで示す）．



（A）　　（B）

図 6.9: （A）ガウス関数型初期マスク配列　　（B）「門構え」学習後マスク配列

6.2.4 類似文字グループ「門構え」を対象とする FML 実験

次に手書漢字「門構え」の類似文字グループの識別について実験を行った．

図 6.9（A）は学習前のマスク配列,（B）は学習後のマスク配列の様子を示す図である．なお，この図では表示を見やすくするために（B）は（A）に比べてコントラストを下げて表示している．すなわち実際に FML によって生成されたマスク配列はこの図のものよりコントラストが強い．

この図から「門構え」の違う部分，すなわち中央やや下部分にマスクの初期値とは異なるマスクが形成されていることが分かる．主に縦線抽出型のマスクが多く形成されていることが見てとれる．

個々のパターンの特徴ベクトルのチェックについてはこの実験ではカテゴリ数が多いので省略する．

6.2.5 認識精度の評価と結論

以上の実験で得られた認識精度をまとめると図 6.10 の右側の表のようになる．表の数値は評価用データによる誤読率でリジェクト無しの条件である．なお本報告での認識精度の数値は特にことわりの無い限り，総べて評価用データにおけるリジェクト無しの誤読率である．図 6.10 の左側はこれをグラフ表示したものである．

本来このような認識精度の評価は厳密な統計分析に基づいて有意性の判定を行なうべきであるが，文字認識の分野ではそうした厳密な評価が行なわないことがしばしばである．特に実用に近い開発でその傾向が強い．

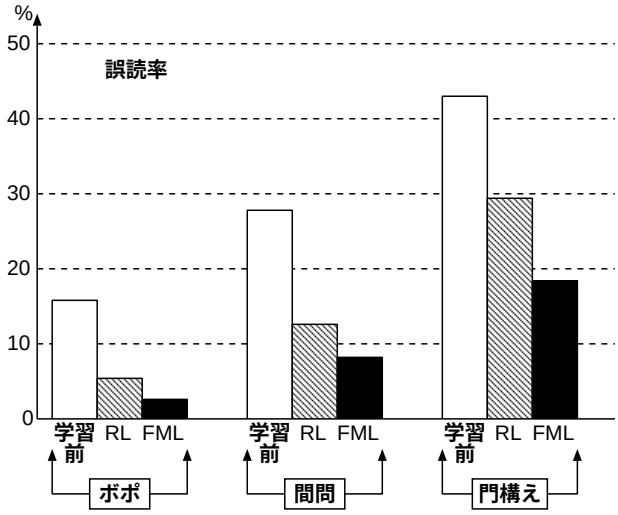
これは評価用パターン・データをＦＩＸできないことに起因している．確かに単に実験という意味ではデータをＦＩＸすれば良いのであるが，実際のＯＣＲが使用されている現場ではそのデータが妥当であるかどうかが誰にも分からない．多分，妥当であろうという程度のことである．実際には，評価用データとは異なる様々なデータが，大量に読み取られ，時間によっても変化するし，場所によっても，機械によっても変化する．その意味で実際に実用性のある方式を開発する場合には，ある固定的なデータに厳密に依存しすぎる判断はかえって判断ミスにつながるリスクがある．むしろ厳密に評価せずにおおざっぱに評価する方が安全である．

学術的にはデータを固定して厳密に評価すれば美しい結果は得られるが，そのデータが妥当かどうかが分からないという宿命を考えると，実用性を意識した研究開発では経験にたよった認識精度の評価を行なわざるを得ない．

本研究でもその意味で厳密な統計評価を行なわなかった．もちろんそうする事は可能だが，厳密な調査にはかなりのコストがかかるので本研究ではそれよりもパラメータを振った実験を広範囲に行なう方を優先した．

さて，そのような経験的な評価に基づく結論であるが，この実験結果の評価を以下に述べる．

図 6.10 から本方式が文字認識の性能向上に貢献していること，すなわち学習がしっかり行なわれていること



字種	学習前	RL	FML
ボポ	15.9%	5.4%	2.7%
間問	27.8%	12.7%	8.2%
門構え	43.0%	29.5%	18.4%

図 6.10: 認識精度の比較（誤読率）

が確認できる．表の RL はフィルタマスク学習を行わず，辞書のみを学習した場合の結果で LVQ と同じである．このことから FML による認識システムが LVQ のものよりも高精度である確認できる．この結果から FML によって類似パターンの違いを強調するマスクが生成されるということが認識性能に役だっているということも推測できる．

なおそれぞれの学習回数を表 6.1 に示す．これら回数はパラメータを振って実験した中で評価用データで最高の認識率となった学習回数である．以下その他の実験でも同様のポリシーで学習回数を決めた．

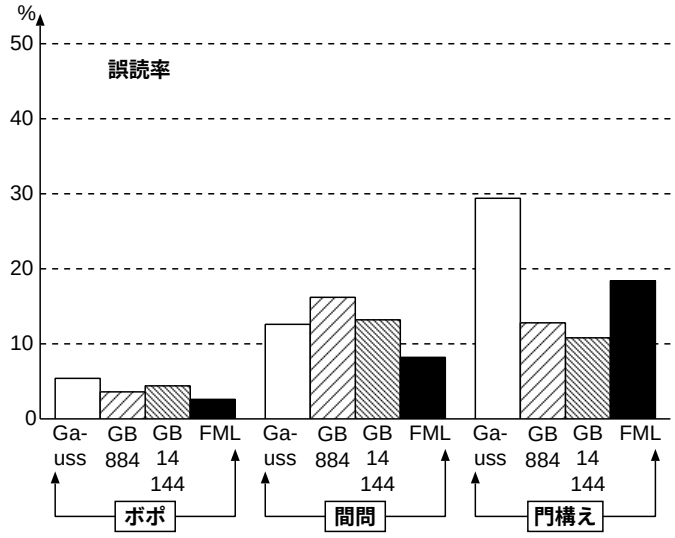
表 6.1: 学習回数

字種	RL	FML
ボポ	2,510	3,050
間問	830	1,430
門構え	400	5,760

認識性能面での学習効果のチェックをガボールフィルタ [16] と比較することで行った．ガボールフィルタとして２種類のパラメータのものを用いた．１つは 8 × 8 の局所領域で 4 方向のもの，もう１つは 14 × 14 の局所領域で 4 方向のものである．認識方式としては LVQ を用いた．この LVQ はシグモイドの微分を窓関数として用いた方式のものである．この 14 × 14 は実用化されている文字認識の分野ではやや大きいサイズであるが，「門構え」の認識に関しては良好な認識性能を得ることのできるサイズであったのでこのサイズも実験に加えた．なお，これらのガボールフィルタのマスク値を決めるパラメータは，数字やカタカナなどの一般的な文字認識実験を通して得られた最適値である．

認識結果を図 6.11 に示す．この図の「Gauss+LVQ」はガウス関数マスクと認識部に LVQ を用いた結果，「GB884+LVQ」「GB14144+LVQ」はガボールフィルタと LVQ の結果,「FML」は FML での認識結果である．FML は本来，学習のためのシステムであるが，表の「FML」ではその認識部の性能を示している.「Gauss+LVQ」は FML において辞書学習のみを行ったのと同じなので，これと「FML」を比較することでマスク学習が有効に行われたかどうかをチェックすることができる．なおこの「Gauss+LVQ」の結果は図 6.10 の RL と理論上は同じだが，シミュレーション・ソフトやパラメータが違うために若干結果が違っている．図 6.10 のところで述べたのと同様，図 6.11 の「Gauss+LVQ」と「FML」を比較することで FML の効果が確認できる．

次にガボールフィルタとの比較について述べる.「ボポ」や「間問」のケースでは FML の結果はガボールのそれよりも良い．これにより FML によるマスクは類似文字が良く似ている場合にはガボールよりも有効であると



方式	Gauss +LVQ	GB884 +LVQ	GB14144 +LVQ	FML
ボポ	5.5%	3.8%	4.5%	2.7%
間問	12.7%	16.1%	13.2%	8.2%
門構え	29.5%	12.9%	10.9%	18.4%

884：8 × 8，4 方向， 14144：14 × 14，4 方向

図 6.11: ガボールフィルタとの比較結果（誤読率）

推定される．一方で「門構え」ではガボールよりも精度が悪くなっている．

このことから類似性がゆるやかな場合にはさらなる改良が必要と推定される．実証という意味ではもっと数多くの例が必要であるが，以上は方式検討のための見通しというレベルでの結論である．

もちろんすべてのケースについての実験ではないので，あくまでも推定ではあるが本実験によって次のようなことが予想される．

（1）FML によって類似文字の違いのある部分を強調するフィルタマスクが生成される．
（2）FML は類似文字間の認識精度を向上させるという意味で有効である．
（3）FML を類似文字識別方式として見た時に LVQ よりも認識面で優れている．
（4）一部分が良く似た類似文字ペアについては FML による特徴抽出はガボールフィルタよりも性能が良い．
（5）類似性が多様な文字グループについては FML はガボールフィルタよりも性能が悪い．

この（5）を受けた改良策として後で述べる「多面化」と「多重化」の実験を行ったのであるが，この改良策の報告の前に基礎的な確認事項として，内積型の FML すなわち認識部に内積を用いた方式についての実験について次に述べる．

6.3 内積型フィルタマスク学習

6.3.1 内積型とユークリッド距離型 FML の比較

ここまでの実験で用いてきた基礎的な方式，すなわちユークリッド距離に基づく方式に替えて，ここでは内積演算すなわち単純類似度を適用した場合についての実験を行った．LVQ に見られるようにこの種の方式ではユークリッド距離が用いられる事が多く，本研究でもここまでの基本的な実験では認識部にユークリッド距離を用いて来た．これに対してここでの実験の目的は，ユークリッド距離を内積演算に置き換えることによってフィルタリングの内積演算を含めてシステム全体を内積演算で統一した場合の調査，認識精度の評価，生成されたフィルタマスクの評価を行なうことにある．

システム全体を内積演算で統一することはシステム設計や理論的考察を行う上でメリットがあり，その調査・研究は是非取り組んでおきたい課題である．内積型で性能面の問題が無いかどうかを本実験で調べている．

また FML を NN の観点で捉える意味では，認識部を内積演算に置き換えることによって，より一層 FML が普通の NN に近づくので，その点でもこれは興味深い実験項目である．

図 6.12 は認識部にユークリッド距離を用いた場合の学習後のマスク配列である．図 6.13 は内積型での学習後のマスク配列である．ユークリッド距離型と比べると若干異るが，内積型でもほぼ同じマスクが形成されている

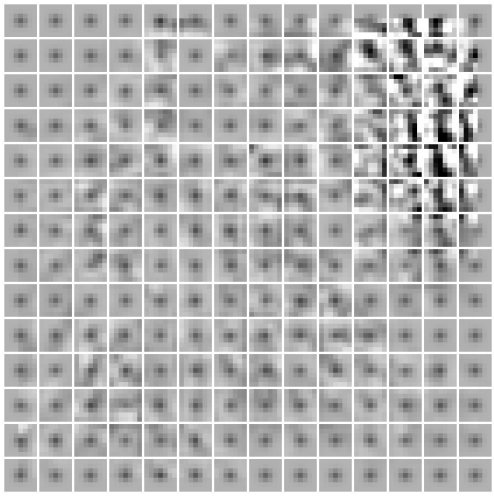


図 6.12: ユークリッド距離のケース

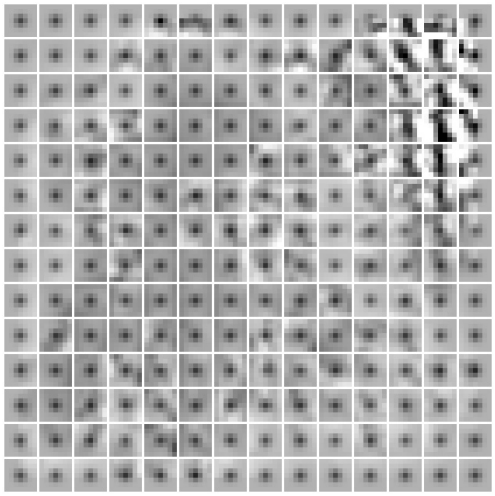
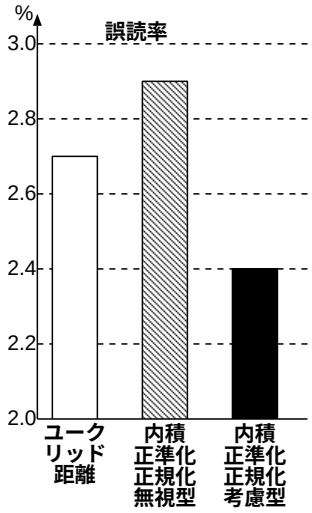


図 6.13: 内積演算のケース



方式	ユークリッド距離	内積	
		正準化・正規化無視型	正準化・正規化考慮型
誤読率	2.7%	2.9%	2.4%
学習回数	3050	2400	4080

図 6.14: ユークリッド距離型と内積型の比較実験（誤読率）

ことが確認できる．

6.3.2 内積型フィルタマスク学習での認識精度

認識精度を図 6.14 に示す．なおこの図の誤読率のグラフは他のグラフに比べて縦軸をかなり拡大しているので，差があるように見えるが実際にはその差は少ない．

ユークリッド型と内積型の正準化・正規化無視型では内積型の方が若干劣るが，ほぼ同程度の性能であった．

次に正準化と正規化の両方を考慮した場合の実験を行った．この実験での更新式は式（4.45）なので更新後の辞書はほぼ正規直交している．しかしながら更新による正規直交からのずれが累積する問題があるので更新後に無視型と同様，正規直交化を行うようにしている．

認識精度を図 6.14 に示す．これは無視型と比べると良い性能である．実験数が少ないのでこれだけでは断定できないが，正準化や正規化を無視することは性能劣化につながる可能性がある．

また，これによってできたマスクパターンも目視でチェックし，無視型と同様であることを確認した．

類似パターン識別に対するフィルタマスク学習実験において，ユークリッド距離を内積に置き換えても認識精度と学習されたマスクのそれぞれについてほぼ同等と言って良い結果が得られた．

この結果から FML を内積演算で統一することができるようになり，これによって FML の将来の研究課題を，ニューラルネットと同様，あるいはニューラルネットの１形式として調査・研究することも可能となる．

また内積演算における正準化，正規化の実験を行った結果によると，それらを考慮した学習方式の認識精度において若干の効果があることも確認された．

6.4 多面化 FML の実験

ここでは FML の拡張の１つである多面化（MFML）に関する実験の報告を行なう．

6.4.1 ガウス関数型初期マスクによる多面化 FML の実験

まず最初にフィルタマスク配列の面数を４とし，その初期マスク配列として前の実験で用いたのと同じガウス関数型初期マスク配列を用いた場合の実験を行なった．用いた文字パターンは「門構え」である．

実験では総ての面で同じマスク配列を用いる．当然，総ての面で同じ初期マスク配列を用いると総ての面は完全に対称であるから，それぞれの学習におけるマスク値の更新量も同じになるので多面化する意味が無くなってしまう．そこでここでの実験は前の章で説明したディレイド・スタートを用いた．

具体的にはスタートを，学習回数にして 1,000 回づつ遅らせるようにしてディレイド・スタートを実施した．詳しく述べると，今回の方式では最初に面番号 0 の面（マスク配列）のみ学習させ，その他の面は初期マスクのまま維持し，1000 回学習後から面番号 1 の面を学習に追加し，2000 回学習後から面番号 2 の面を学習に追加し，3000 回学習後から面番号 3 の面を学習に追加し，以降すべての面の学習を行なうようにしている．

図 6.15 にこの方法によって生成されたマスクを示す.（０）から順に 0 回，1000 回，2000 回，3000 回スタートの学習後マスク配列である．学習回数は 8850 回である．

図 6.15 を注意深く見ると同じ位置のマスクが（０）～（３）で異っていることが分かる．この違いを見やすくするために同じ位置の４つのマスクを縦に並べたものを図 6.16 に示す．全部で７箇所の位置について表示した．上から０オリジンで１１行目の位置で，左から０オリジンで横方向に２番目の位置から８番目の位置のマスクを表示した．図 6.16 の上側の番号はこの横方向の位置を示す番号である．この番号の下に縦に並んだマスク４個がその位置の局所領域に対応する１ユニットに含まれるマスクである．左側の縦に示した（０）～（３）はユニット内の面番号 0 のマスクから面番号 3 のマスクを意味しており，この記号は図 6.15 の（０）～（３）に対応している．

この図を見ると１ユニットの中に２方向ないし３方向の線分抽出フィルタの機能を持ったマスクが生成されていることが４，５，７，８の位置のユニットを見ると分かる．このことは多面化の方式はディレイド・スタートのような非対称性を導入することで，複数の線分方向を抽出する能力のあるフィルタマスク群を生成できることを示している．しかしながらこれらのマスクは一様でなく，またノイジーである．本来マスクが生成されるべき局所領域でも，ある局所領域では方向性が出るが別の所では出ないといった具合である．このことはガボールフィルタなどの人工的な線分方向抽出フィルタと比べた時に，マスクの作りが良好とは言えない状況であると考えられ認識精度の面で問題があると予想される．しかしながらこの実験結果で重要なことは FML にこのような複数の線分方向抽出フィルタマスクを生成する能力があるということである．

図 6.15 によると，生成された線分抽出マスクの強度は最初にスタートした面から順に強くなっている．これらの４個のマスク配列の強度は均一では無い．この現象は初期フィルタマスクの影響が後々まで残ることを示している．実験してみるまでもなく良く知られていることではあるが，この事実は「初期値の学習に与える影響が大きい」ということであり，FML で言えば初期マスク配列の設計が重要だということである．

さて，この各面の不均一性が認識性能面に与える影響はあるのであろうか？そこで，それぞれの面の学習回数が同じになるように適宜，回数の多い面の学習を休ませるようにして各面の強度が均一になるように調整した学習の実験も行なった．具体的には学習回数 4,000 以降，面 0 では 4,000-7,000 回，面 1 では 5,000-7,000 回，面 2 では 6,000-7,000 回において学習を行なわない設定とした．これによって生成されたマスクを目視すると若干の強度差は残るものの（最初にスタートした方の強さが若干残る）ほぼ均一の強度であった．認識精度を比

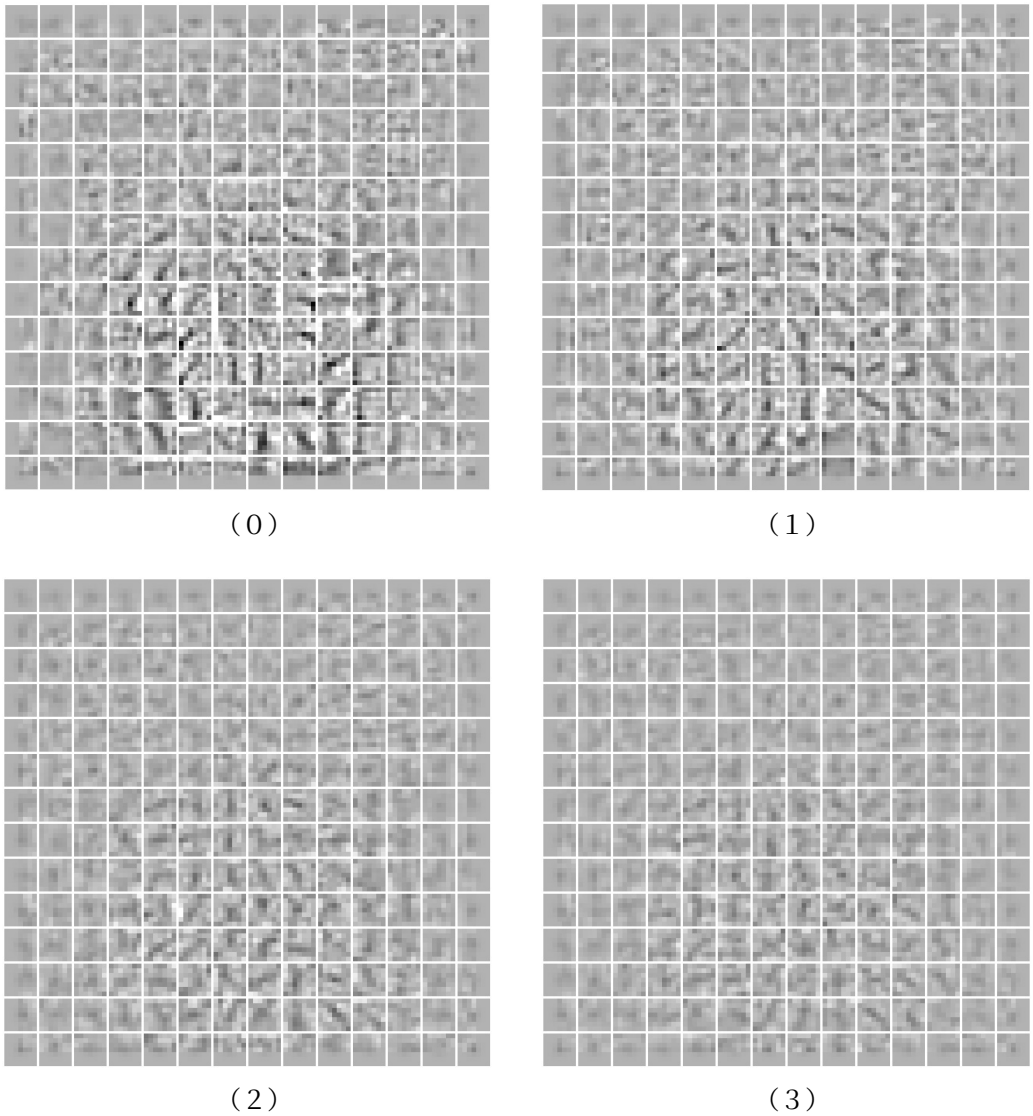


図 6.15: （0）→（3）ディレイド・スタートで得られた４面の各マスク配列

較すると均一化前で 13.3%（8850 回学習），均一化後で 13.2%（20470 回学習）となり両者でほとんど変わらなかった．このような不均一性が認識精度に悪影響を与えるということは無いようである．なお，比較実験のために均一化前の学習回数を 8850 回から 20470 回まで伸ばしてみたが，認識精度の向上は 13.3%が 13.2%と，ほとんど変化無く学習はおおむね 8,500 回のところでサチっていた．そのため，次の節で述べる認識精度の表では，図 6.15 のマスクの学習回数にあわせ 8850 回の方の認識精度を掲載した．

多面化における直交化の手法についても実験を行なったが予備実験段階で目だった性能向上が見られなかったので本格的な実験までは行なわなかった．この予備実験では各マスクをベクトルと見て，それら同士のなす角のコサインの自乗を平均としたものを直交化の尺度として用いることとした．この値は直交化を行なわなくても学習が進むにつれて下ることが確認できたが，直交化を行なうことによって行なわない時よりも一層急速に値を下げて０に収束することを確認した．このように直交化自体が行なわれることは確認したが，認識精度についてははっきりした差は認められなかった．しかしながらこれは予備実験レベルでの調査なので将来 FML の調査研究がさらに進んだ時点で，もう一度調査する必要があると考えている．

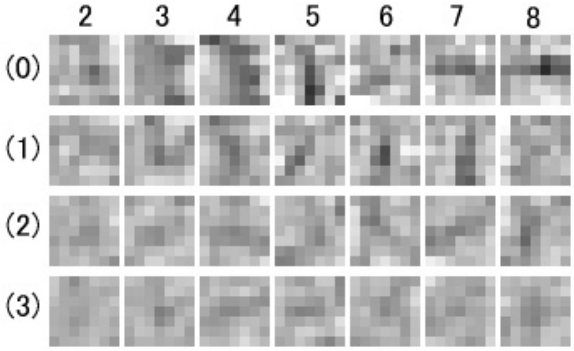
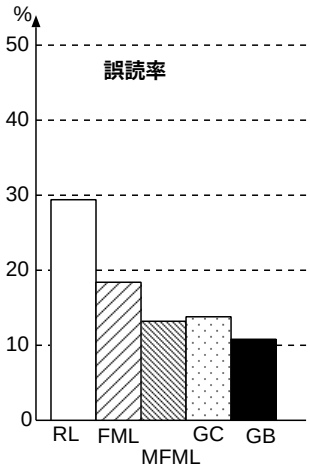


図 6.16: 同じ位置の４つのマスク（０）〜（３）.



	RL	FML	MFML	GC	GB
面数	1	1	4	4	4
誤読率	29.5%	18.4%	13.3%	13.9%	10.9%

図 6.17: ガウス関数型初期マスク用いた場合の多面化 FML の認識精度と他方式との比較

6.4.2 ディレイド・スタートによる多面化 FML での認識精度

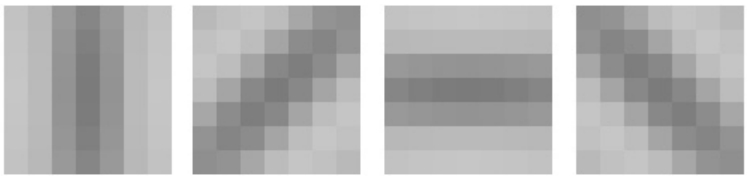
ここでは，ガウス関数型初期マスクとディレイド・スタートを用いた多面化 FML での認識精度の比較を図 6.17 に示す．実験対象は前の実験と同じく「門構え」である．

この表で「RL」は FML における辞書のみ学習，すなわち LVQ と同じもの,「FML」は FML での結果,「MFML」は多面化した FML の結果である.「GC」や「GB」は比較のために行なった人工的なフィルタによる認識結果でその認識方式は LVQ である．

「GB」は４方向のマスク（４方向× ２種類）を持ったガボールフィルタ,「GC」は４方向のマスクを用いた普通のフィルタリングの特徴抽出方式で，そのフィルタマスクにガボールフィルタで用いられているサイン，コサインの２種類マスクのうちコサインのマスクを用いたものである．

面数は「GB」以外では配列数に等しく,「GB」ではその 1/2 である.「GB」すなわちガボールフィルタでは２つのマスクの内積計算結果を平方根関数で１つの値にする形式なので，実際に１ユニット内のマスク数は表に記載された面数の２倍となる．比較は出力特徴の次元数が同じもの同士を比較しているので，そのように記載すべきだが，ここでも分かりやすさのために面数という言い方を導入し，ガボールの場合には２つのマスクを１セットとしてこれを面と呼ぶことにした．なお，面数 1 は多面化していないフィルタを意味している．

「RL」と「FML」の差は FML の効果を示すもので,「FML」と「MFML」の差は多面化の効果を示すものである．ここまでは順調にその性能を伸ばしている．一方「MFML」の結果を同じ４面の「GC」と比べると若干性能が上回っており MFML の能力の高さを確認できる．この理由は評価に用いた文字セットが「門構え」だからである.「GC」では特徴として捉えるべきでない「門」の部分まで一様にフィルタリングして特徴として取り出すので，その分この文字セットについては「GC」の方にハンデがあると考えるべきである．一方この実験



（A）　（B）　（C）　（D）

図 6.18: 多面化実験での方向性を持った初期マスク

結果での「MFML」と「GC」との差はそれ程大きく無いので，一般的な文字セットすなわち局所的に同じ部分が無いような文字セットの場合には,「GC」の方が認識精度で上回ると予測される．ここまでの比較で考えると「門構え」のような文字セットに対しては多面化の効果は大きいと判断できるが，これに対して「GB」と比較すると MFML はかなりの程度で認識精度が悪い．その意味ではさらなる認識精度の向上策が必要である．

6.4.3 方向性を持った初期マスクによる多面化 FML の実験

次に，MFML におけるもう１つの非対称性の導入法として初期フィルタマスクの各面に異なるマスクを用いる方式について調査した．

この実験での初期ベクトルは GC と同じものである．GC のフィルタマスクは必ずしも「門構え」文字セットにチューニングしたものでは無いが，一般的な文字認識のために最適化したものなので,「門構え」文字セットに対してある程度高性能なマスクとなっている．そのことは前節の実験結果からもうかがわれる．それ故にこの初期マスクから出発して，さらに高精度な学習後マスクが作れるかどうかは FML の実用化に向けた調査研究としては重要な一里塚となる．ここでの実験でそのことを確かめる．

まずこの実験で用いた初期フィルタについて説明する．以下は本実験で用いるフィルタマスクを生成する式で f_c と f_s の２種類のマスクを記述している．これはガボールフィルタで用いられるもので，本実験ではこれらのうち f_c のマスクを用いる．

$$
\begin{aligned}
f_c &= g\cos(2\pi\frac{r_x}{u}),\\
f_s &= g\sin(2\pi\frac{r_x}{u}),\\
g &= Aexp(-\frac{1}{2}\{\frac{r_x^2}{\sigma_x^2}+\frac{r_y^2}{\sigma_y^2}\}),\\
r_x &= \quad x\cos\theta_t + y\sin\theta_t,\\
r_y &= -x\sin\theta_t + y\cos\theta_t.
\end{aligned}
\tag{6.2}
$$

これらの式で θ_t はマスクの線抽出方向を決めるパラメータ，x, y は画素位置， σ_x, σ_y は分散， u はコサインの周期を決める係数，A はフィルタの強度を調整する係数である．方向を示す θ_t の実際の値は方向数が 4 のケースでは $\pi/2, \pi/4, 0, -\pi/4$ である．その他のパラメータは $\sigma_x^2 = 2.5$， $\sigma_y^2 = 10$， $u = 7.0$， $A = 0.3$ である．

この式ではマスクサイズがコサインの周期に比べて小さいので，ピークが中央に１個ある程度のマスクになっており一般的なガボールの波打つ形と異なるが文字認識ではこの程度が一般的である．このマスクパターンを図 6.18 に示す．

図 6.19 は学習後のマスクを表示したもので，それぞれ初期マスクの（A）～（D）に対応している．学習回数は 15,240 である．これを見ると中央下部が大きく変化していることが分かる．一方，上部，左右の変化は少ない．これはどの配列でも同じ傾向である．

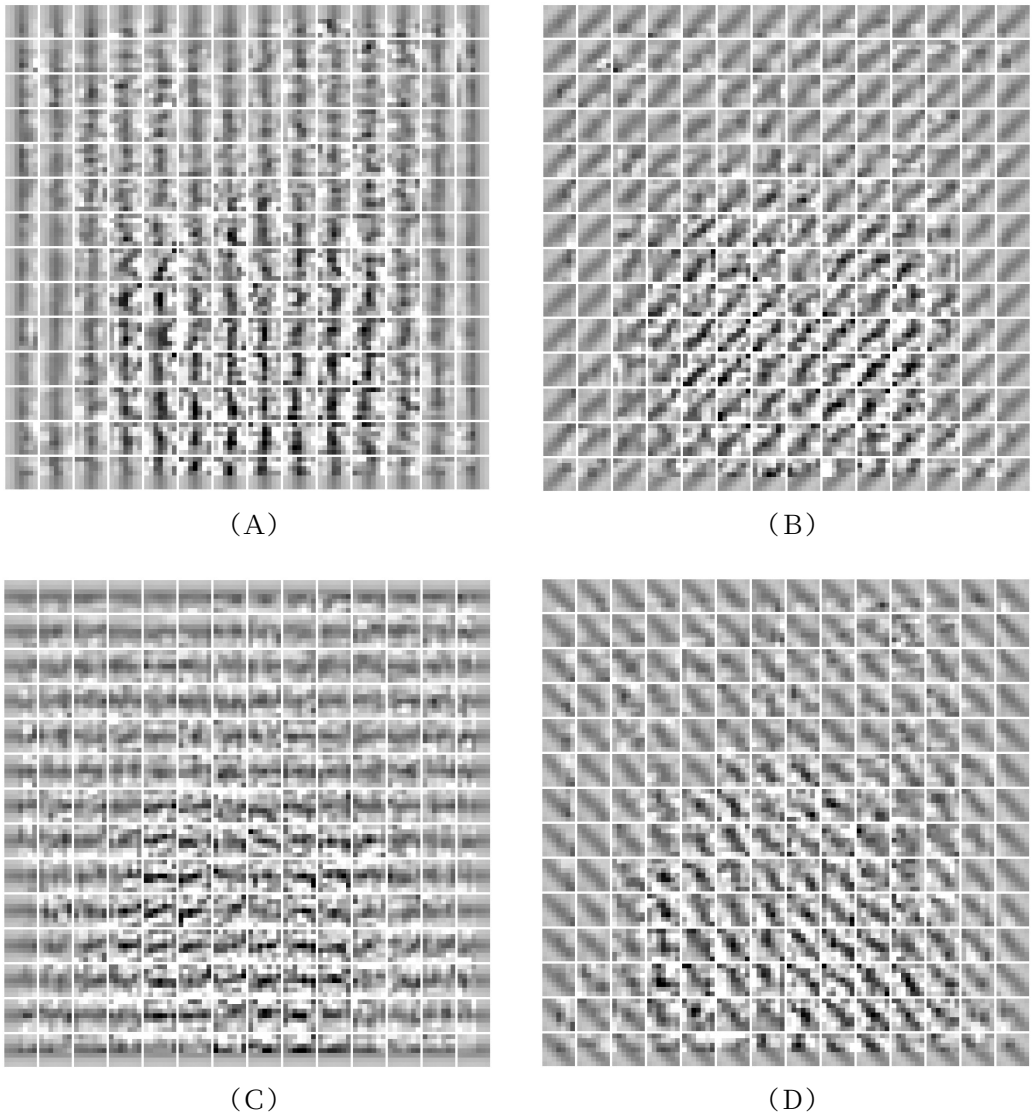


図 6.19: 方向性を持った初期マスクによる多面化実験での学習後のマスク配列

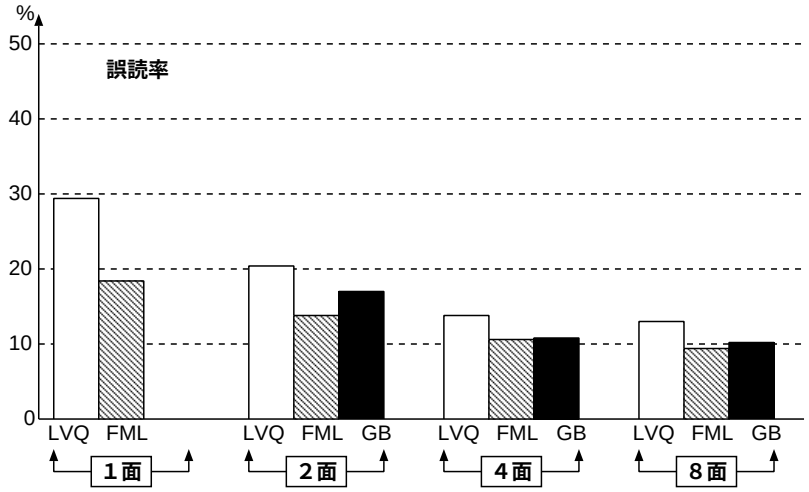
6.4.4 方向性を持った初期マスクによる多面化 FML での認識精度

ここではマスクの方向数を 2,4,8 として実験を行った．その結果を図 6.20 に示す．マスク数 1 は Gauss マスクのもので図 6.10 のものを再掲した．この表中で「面数」は前の実験と同様に「GB」以外では配列の数,「GB」ではその 1/2,「LVQ」 は GC マスクと LVQ の結果,「FML」 は FML の結果,「GB+LVQ」はサイズ 14 × 14 のガボールと LVQ の結果である．数値はいずれも誤読率である．

前の実験でも述べたように LVQ と FML の比較が学習の有効性を示すので，ここでの学習は有効に行われていると判断できる．また面数も多ければ多い程良いという結果も出ている．もっとも 8 面の場合には面数ほどの性能は出ていないようでもある.「FML」の結果は「GB+LVQ」の結果よりも良い．これは生成されたフィルタが LVQ を認識に使うという前提ではガボールと等しいか，それ以上の性能を有しているといえる．その範囲内での改良は成功と言える．

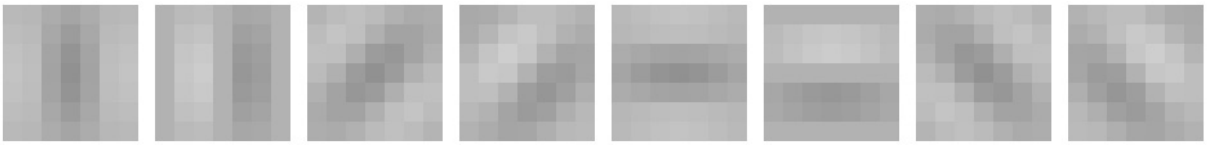
学習時間については多面化 4 面,「門構え」の学習 1 回に要する時間が 3.4G PenIV PC で約 19.0 秒であった．

次に多重化による改良について述べる．



面数	1	2	4	8
LVQ	29.5%	20.3%	13.9%	13.0%
FML	18.4%	13.9%	10.6%	9.7%
GB+LVQ	-	17.0%	10.9%	10.1%

図 6.20: 方向性を持った初期マスクによる多面化 FML の認識精度と他方式との比較



↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑
（コサイン型 サイン型） （コサイン型 サイン型） （コサイン型 サイン型） （コサイン型 サイン型）
（ A B ） （ C D ） （ E F ） （ G H ）
第０面 第１面 第２面 第３面

図 6.21: 多重化実験における初期マスク配列

6.5 多重化 FML の実験

6.5.1 多重化 FML で生成されるフィルタマスク

多面化によって FML の性能向上が確認できたもののガボールと比較した場合にはその性能を大きく上回るレベルになく，実用性を考えた場合にはさらなる性能向上が望ましい．そこでここでは多面化に加えて多重化（CFML）の実験を行なった結果について報告する．

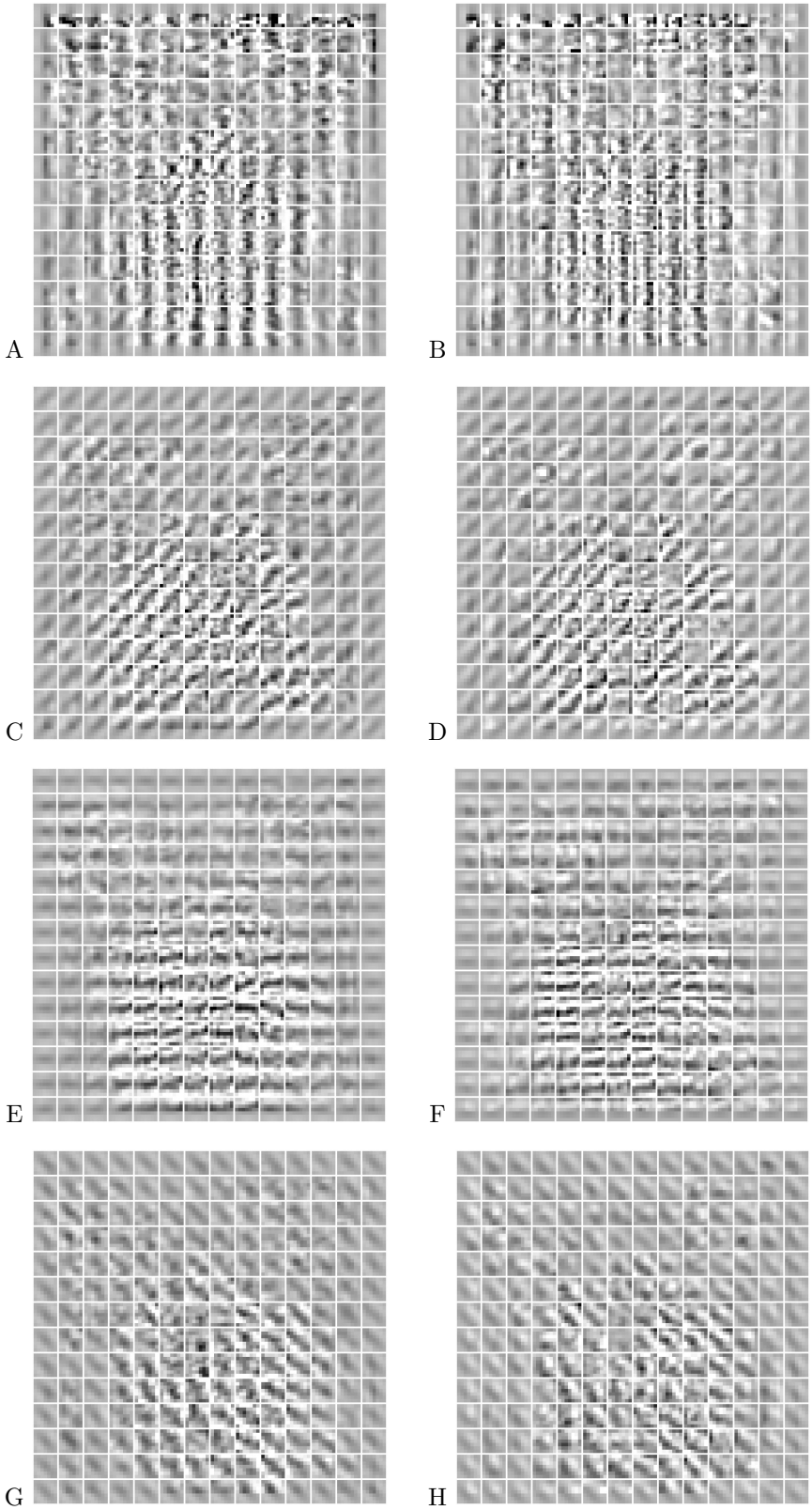
この実験の初期マスクもガボール型のもので式（6.2）である．そして，２つのマスク f_s, f_c を局所領域 i のベクトル $f_s^{(i)}, f_c^{(i)}$ として記述し，以下の式により特徴値 $\hat{x}_i$ を計算する．

$$\hat{x}_i = \sqrt{(f_c^{(i)T} z^{(i)})^2 + (f_s^{(i)T} z^{(i)})^2}. \tag{6.3}$$

実際のパラメータは $\sigma_x^2 = 3.5$，$\sigma_y^2 = 2.5$，$u = 6.0$，$A = 0.05$ であり方向数は 4 である．マスクパターンを図 6.21 に示す．左から順に２個づつの組で，左側がコサイン型，右側がサイン型で，４方向分全部で８個のマスクである.（A，B）が第０面のコサイン型とサイン型のマスク,（C，D）が第１面,（E，F）が第２面,（G，H）が第３面である．

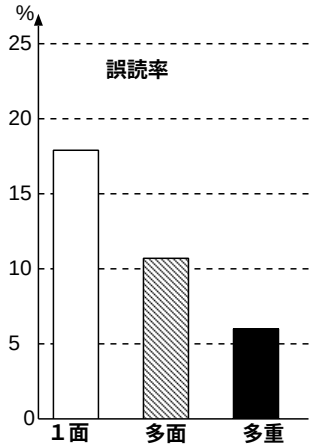
図 6.22 に学習後のマスク配列を示す．左側がコサイン型，右側がサイン型である．縦方向に４方向分のマスクの組を表示している．学習回数は 13,480.

「多面化」の時と同じように中央下部で大きな変化，周辺部分が無変化という学習傾向がどのマスクにも見られる．



（A，B）が第０面のコサイン型とサイン型．以下同様.

図 6.22: A〜H：多重化における学習後のマスク配列



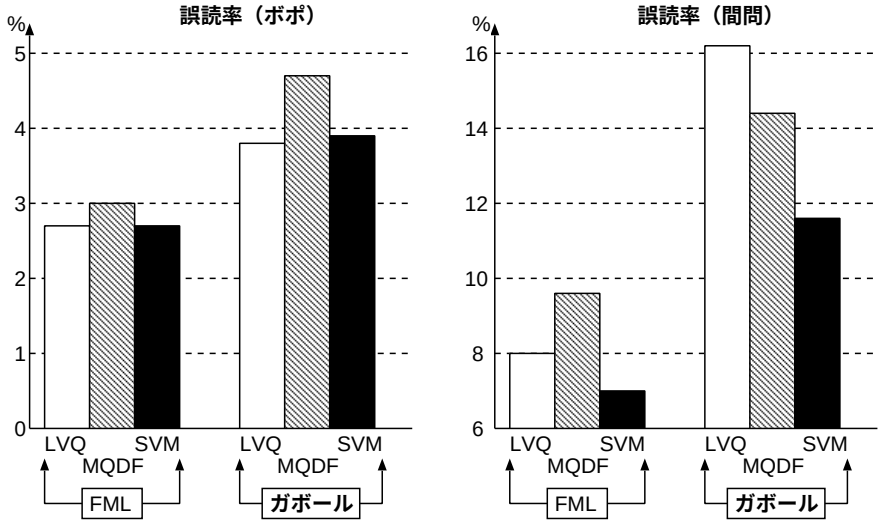
GB14144+LVQ	１面	多面	多重
10.9%	18.4%	10.6%	6.0%

図 6.23: 多重化実験の結果（誤読率）

6.5.2 多重化 FML で達成される認識精度

認識結果を図 6.23 に示す．ここで「GB14144+LVQ」は多重化の辞書のみ学習，すなわちこれまでの実験で使ってきた記号で言えば「RL」に相当している．それ故,「多重」と「GB14144+LVQ」を比べることによって多重化 FML の学習効果を確認することができる．この表によればここでも学習効果があったことが認められる．次に「多面」と「多重」を比較することで「多重化」の効果が確認できる．

学習時間については多重化４面,「門構え」の学習１回に要する時間が，3.4G PenIV PC でおよそ 37.6 秒であった．



文字セット	ボポ		間問	
特徴	FML	GB	FML	GB
LVQ	2.7%	3.8%	8.0%	16.1%
MQDF	3.0%	4.7%	9.5%	14.3%
SVM	2.7%	3.9%	7.0%	11.5%

図 6.24: １面 FML で作成されたマスクを LVQ 以外の認識方式と組み合わせた場合

6.6 FML で生成したマスクを LVQ 以外の認識方式に適用する実験

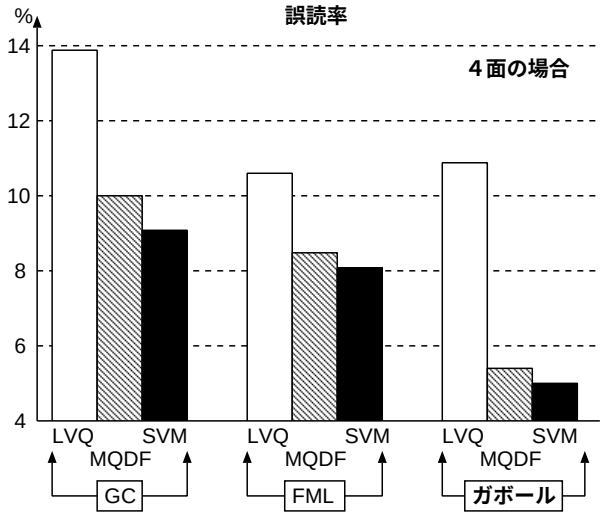
ここでは FML によって生成されたマスクを LVQ 以外，すなわち学習で用いた認識系以外の認識系に適用する実験を行った．理論からすれば学習で用いたものと大きく異る認識系に学習済のマスクを用いることは望ましくは無い．しかしながら実際の実用の場面ではそのような要請も十分有り得ることであり，こうした観点からこの「他流試合」とも言うべき実験を実施した．

ここではテストする認識方式として LVQ, SVM（Support Vector Machine）, MQDF を用いた．SVM としては「svmlight」 [17] を用い，MQDF としては文献 [29] のものを用いた．このうち LVQ は FML の認識部であるからこれは「他流」ではなく，いわば「同門」である．これは FML を認識システムとして用いる場合とほぼ同じ性能になるが，比較・チェックのために実験項目に加えた．

6.6.1 「ボポ」「間問」の文字セットで１面 FML のケース

まず「ボポ」「間問」の場合の結果を図 6.24 に示す．

この結果から見ると FML は「GB」すなわちガボールフィルタより優れていることが分かる．ガボールフィルタは「ボポ」「間問」のように異なる部分が小さい類似文字ペアの識別のように単純な識別問題に対してはオーバースペックで，かえって識別性能が劣る結果となってしまうものと思われる.「ボポ」「間問」のような識別問題には FML が適しているのではないかということが良く推測できる結果である．



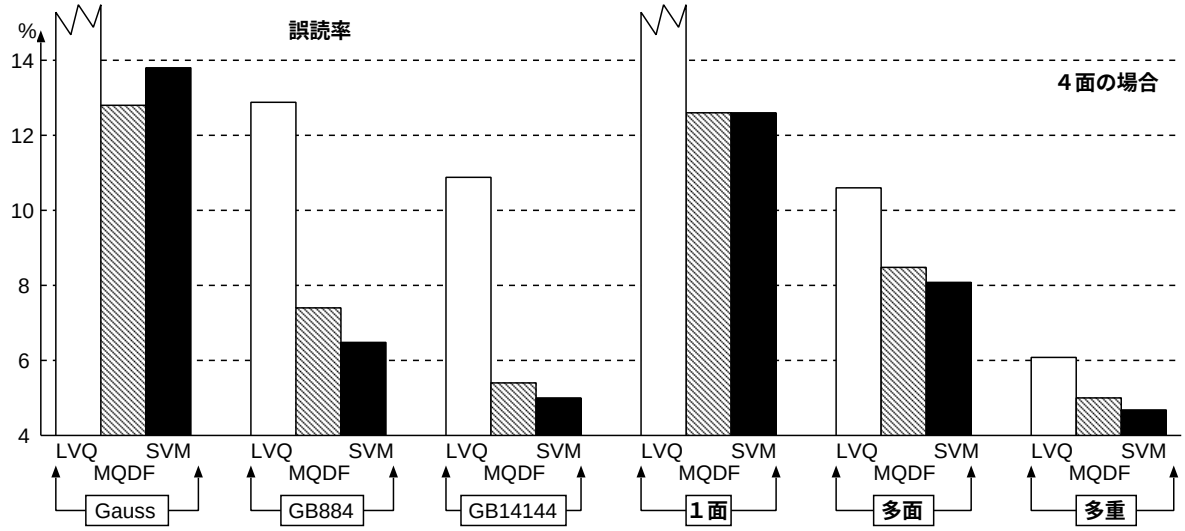
面数	1		2			4			8		
特徴	GC	FML	GC	FML	GB	GC	FML	GB	GC	FML	GB
LVQ	29.5%	18.4%	20.3%	13.9%	17.0%	13.9%	10.6%	10.9%	13.0%	9.7%	10.1%
MQDF	12.8%	12.6%	11.6%	10.0%	9.0%	10.0%	8.5%	5.4%	9.2%	7.5%	5.1%
SVM	13.8%	12.6%	11.6%	10.3%	7.8%	9.1%	8.1%	5.0%	8.8%	7.5%	4.9%

図 6.25: 1 面および多面化 FML で作成されたマスクを LVQ 以外の認識方式と組み合わせた場合

6.6.2 「門構え」の文字セットで多面化 FML のケース

次に「門構え」の文字セットに多面化 FML を適用した場合の実験結果を図 6.25 に示す．ここでの初期フィルタマスクも GC 型のものである．この中で特に 4 面の場合をグラフとして示す.「FML」は FML で生成されたマスクによる特徴抽出と各認識方式との組み合わせ,「GC」はガボールのコサイン型のマスクを用いた特徴抽出部との組み合わせ,「GB」はガボールフィルタとの組み合わせである．SVM に関しては特徴ベクトルの次元が 4 面以上では 784 以上という大きな数になるので，400 次元まで次元圧縮により次元を削減してシミュレーションを行った．次元圧縮には主成分分析によるものを用いた．また面数が 1 のものは初期フィルタマスクがガウス関数型のものである．

これらの結果から FML によって生成されたフィルタマスクは，たとえ認識部が SVM に変わったとしても GC より高性能であることが分かる．一方で GB は FML よりも良い．類似性が緩和する「門構え」になるとガボールが威力発揮してくることがこの実験からも見て取れる．



Method	Gauss	GB884	GB14144	1面	多面	多重
LVQ	29.5%	12.9%	10.9%	18.4%	10.6%	6.1%
MQDF	12.8%	7.4%	5.4%	12.6%	8.5%	5.0%
SVM	13.8%	6.5%	5.0%	12.6%	8.1%	4.7%

図 6.26: 1 面，多面，多重 FML で作成されたマスクを LVQ 以外の認識方式と組み合わせる場合の評価実験

6.6.3 「門構え」の文字セットで多重化 FML のケース

多重化の実験は４面のみで行なった．１面の結果や多面化の結果を含めた一覧を図 6.26 に示す．ここでの初期フィルタマスクも GB すなわちガボール型ものである．

この表で「Gauss」はガウス関数型マスクによる特徴抽出と各認識方式との組み合わせ,「Gabor884」「Gabor14144」はガボールフィルタとの組み合わせ,「１面」は１面 FML で生成されたマスクによる特徴抽出との組み合わせ,「多面」は多面 FML で生成されたマスクによる特徴抽出との組み合わせ,「多重」は多重 FML で生成されたマスクによる特徴抽出との組み合わせである．

この比較実験によると 1 面の FML で作られたマスクによる結果（「１面」）はほとんどガウスフィルタ（「Gauss」）に近く，性能向上はほとんど無かったと言って良い．これに対して多面化することで，対ガウスと言う意味では性能向上が認められたが，対ガボール（「Gabor884」,「Gabor14144」）という意味では性能向上は無かった．さらに多重化することで，ようやくガボールに対して遜色の無い結果を得ることができたということが分かる．しかしながら現状ではまだ同程度の実力なので,「他流試合」で勝てるほどの効果があったとは言えないレベルであり，今後もさらなる性能向上を検討していく必要がある．

また学習効果という視点で言うと,「１面」の初期フィルタと「Gauss」のそれは類似しているので,「１面」と「Gauss」の結果を比較することで FML の学習効果を確認でき，同様に「多重」の初期フィルタは「Gabor14144」のそれは類似しているので,「多重」と「Gabor14144」の結果を比較することで FML の学習効果を確認できる．この比較によると学習効果は「他流試合」で評価した時は大きいとは言えない結果であった．これは学習で用いる文字認識部が異ることからやむを得ない現象であるけれども，実用面からは何らかの改善が望まれる．

6.6.4 文字認識実験結果に対する考察

ここで本実験で得られた結論について簡単にまとめておく．

（1）「ボポ」や「間問」では FML の効果を確認できた．このような特に類似した文字ペアを識別対象類似文字として選定することにより，FML による認識系を複数作成して組み合わせることで，全体として高性能な類似文字識別システムを実現できる見通しが立った．

（2）「門構え」についてはさらなる改良が必要である．他流試合の実験では従来方式を若干上回る認識精度が得られたものの，今後のさらなる改良が必要である．また，その余地があるとの知見も得られた．

（3）マスクの可視化によって FML の有効性を視覚的に確認することができた．

（4）非一様なフィルタマスクを作り出すことに成功し，また，その有効性を確認した．

（5）多面化，多重化の拡張を行ない，その拡張方式が有効であることを確認した．

6.7 学習プロセスにおける課題と解決策

6.7.1 パラメータ設定の問題

パラメータ ε_f と ε_r の関係

パラメータ設定は安定で効果的な学習を進める上で重要な課題である．特に式（4.20），式（4.21）の ε_f と ε_r は重要で，降下法の考え方からするとこの両方の値は同じで良いはずだが実際にはそれではうまく行かない．これらのパラメータの値をそれぞれ振って実験を行ない，学習が順調に進むケースを拾って行くと，これらのパラメータの間に関係があることが分かる．その結果からすると適切な学習におけるこれらの値は必ずしも等しくは無い．

この問題を以下，考察して行く．まず１つの例として図 6.27 に示すような簡単な最適化問題を考えてみよう．この図は $x-y$ 平面上に，ある評価関数を設定し，その等高線を表したもので，それぞれの等高線は細長い楕円形をしているものとする．また評価関数は値の低いものほど良いとし，点 B を最小値，すなわち B を底に峡谷のような形をした関数であるとする．

この図で点 A を開始点とし，降下法を適用して最適化しようとすると，最初の進行方向は矢印 C となる．降下法の原理からこの方向は等高線に垂直な方向だからである．この時にこの図の楕円の長軸を短軸の 1,000 倍であると仮定する．つまりこの峡谷が極端なグランドキャニオンのような形だったとするのである．こうすると学習を破綻なく進めるための C の妥当な長さに対し，A と B の長さはあまりにも長くなってしまう．結果的にこの最適化プロセスで A から B に近付くためには膨大な学習時間を要することになる．このケースで短時間の学習状況を見ただけでは学習がまったく進んでいないようにも見えるであろう．実はこの C の方向は望ましい方向とはまったく違うのである．

ここで採用するこの問題の回避手段はこのような極端に薄い楕円形の評価関数を適切なサイズの楕円に変換する手法である．そのような変換作業は単に ε_f と ε_r に異る値を持たせることによって実現することができ，実際に本研究ではその手段を用いた．

さて，ここで問題となるのが実際にそれらをどのような値にするかであるが，それについて以下に考察して行く．

認識系がユークリッド距離の場合の FML でのマスク値の更新量と，辞書ベクトルの要素値の更新量は次式で表される．

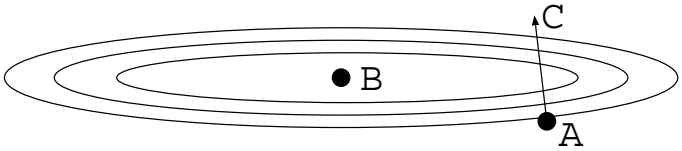


図 6.27: 降下法におけるグランドキャニオン問題

$$\Delta F_{ij} = \mp 2\varepsilon_f l'(d)\rho'(\hat{x}_i)(x_i - \varphi_i)z_j, \quad \Delta\varphi_i = \pm 2\varepsilon_r l'(d)(x_i - \varphi_i). \tag{6.4}$$

ここで新しい記号を導入する．まず $\overline{|a|}$ をある値 a の絶対値の平均の推定値を表すものとする．そして $\langle\langle F_{ij}\rangle\rangle$, $\langle\langle \Delta F_{ij}\rangle\rangle$, $\langle\langle \varphi_i\rangle\rangle$, $\langle\langle \Delta\varphi_i\rangle\rangle$ を次のように定義する．

$$\begin{aligned}
&\langle\langle F_{ij}\rangle\rangle = \overline{|F_{ij}|}, \quad \langle\langle \varphi_i\rangle\rangle = \overline{|\varphi_i|}, \\
&\langle\langle \Delta F_{ij}\rangle\rangle = 2\varepsilon_f \overline{|l'(d)|}\,\overline{|\rho'(\hat{x}_i)|}\,\overline{|(x_i - \varphi_i)|}\,\overline{|z_j|}, \\
&\langle\langle \Delta\varphi_i\rangle\rangle = 2\varepsilon_r \overline{|l'(d)|}\,\overline{|(x_i - \varphi_i)|}.
\end{aligned} \tag{6.5}$$

これらの絶対値の平均の推定値を自動的に実験データから得ることも可能であるが，予想できないような値の変動によるトラブルを避ける意味で，ここではこれを実験データから目視により人為的に決めることによって推定した．以下の値はそのようにして求めた数値である．

$$\langle\langle F_{ij}\rangle\rangle = 0.2, \quad \langle\langle \Delta F_{ij}\rangle\rangle = 1200\varepsilon_f, \quad \langle\langle \varphi_i\rangle\rangle = 32, \quad \langle\langle \Delta\varphi_i\rangle\rangle = 8\varepsilon_r. \tag{6.6}$$

ここで学習対象となる辞書ベクトルの要素値やフィルタマスク値に対し，その値を更新する量の絶対値の平均値を考えた時，その比率が辞書学習とフィルタマスク学習で同じであるように学習パラメータを設定することにする．すなわちこれは細長い楕円を丸くする意図である．式で表すと仮定（6.7）となる．

$$\langle\langle \Delta F_{ij}\rangle\rangle / \langle\langle F_{ij}\rangle\rangle = \langle\langle \Delta\varphi_i\rangle\rangle / \langle\langle \varphi_i\rangle\rangle. \tag{6.7}$$

この仮定（6.7）と（6.6）によって $\varepsilon_f/\varepsilon_r$ は以下のように求まる．

$$\varepsilon_f/\varepsilon_r = 40 \times 10^{-6}. \tag{6.8}$$

この推定値と実験的に求めた $\varepsilon_r/\varepsilon_f$ の関係を比較したものが図 6.28 である．この実験ではまず予備実験や過去の実績から ε_f をある値に設定し，その上で ε_r を振って実験をする．これによって認識結果がベストとなる学習プロセスの ε_r を決める．これを ε_f を振って実験することにより ε_f に対応して決まる ε_r を求める．すなわち ε_f と ε_r の対応表を作成するのである．なお，この時の最適な ε_r の決め方は単に認識精度の最高のものを採用するのでは無く，最高のものと比較してそれほど差の無い結果となった ε_r について，それらの平均値を求めることによって求めた．実際の学習から求まる認識精度にはかなりのバラツキがあるのでこのような平滑化手段が必要である．

この ε_r と ε_f の対応は横軸に ε_r，縦軸に ε_f を取った図 6.28 に “result” として示されている．一方，“estimated” と示されている直線は式（6.8）を図にプロットしたものである．

この推定値は ε_r が 0.03 になるところまでは良く実験値と一致している．結果として，ここで述べたアプローチは ε を推定する上で有効であると判断できる．しかしながらまだ実験例が少なく信頼できるレベルには無いので本研究の他の実験ではこの方法を用いることは避け，実際には ε_r と ε_f の値を振って実験を行ない，最適と思われるものを採用するようにしている．

なお上記の実験は１面ユークリッド距離の FML で認識対象パターンは「ボポ」である．

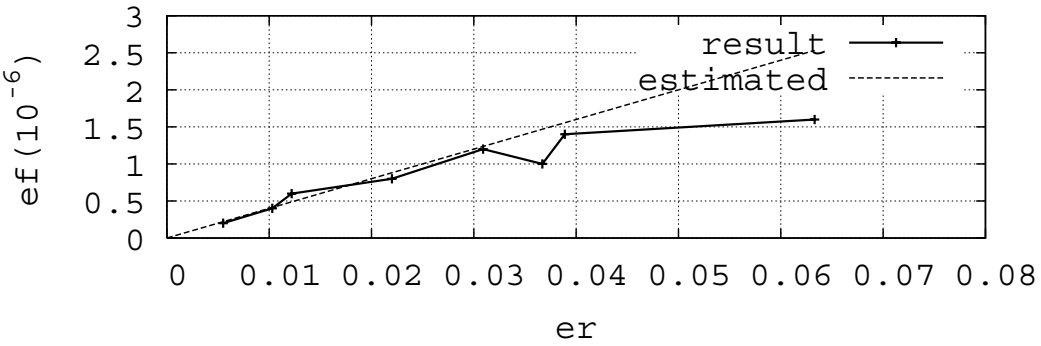


図 6.28: 推定された $\varepsilon_f/\varepsilon_r$ と実験結果との比較

シグモイドの微分型の窓関数における δ の決め方

学習後の認識精度は式（5.17）における δ の影響をかなり受ける．この値は次のようにして決めた．まず d の値の分布をヒストグラムで表示し，目視によって観察することで δ を人為的に決める．おおむね全パターンの 1/5 から 1/10 が学習に参加するようなレベルの値を使っている．この値は経験的なもので LVQ にもあてはまるものである．もちろん単純に人為的に決めた値だけで最終的な値とするのでは無く，その値の回りで δ を振った実験も行なって最終的な値を決めている．

このようなプロセスを自動化することは重要な課題であるが今回の実験の中では取り組んでおらず，これについては将来の研究課題とした．

なお，ここでで取り上げた $\varepsilon_r, \varepsilon_f, \delta$ だけでもパラメータは３個あり，これらを決めるためにはパラメータを振る実験が必要で，パラメータ数が増えるとその実験量は指数関数的に増えてしまう．実際，今回の FML 実験を行なった経験では，実験の量から言って振れるパラメータの数は２～４個が限界だと感じている．したがってこのパラメータを自動的に決める技術は，さらなる性能向上や実用化を狙う意味では重要な課題である．

しかしながら自動化技術を開発するためには，その自動化技術が妥当かどうかを判定するための正解パラメータが必要で，その正解パラメータはパラメータを振った実験でしか求められない．要するに何であれ，最初はパラメータは振らざるを得ないわけで，その意味で今回の実験はそのためのデータ収集作業と考えて地道な調査を実行した．

6.7.2 学習の崩壊とその対策

学習シミュレーションの状況を確認するための手段として，学習が進んで行く様子を学習回数に対する認識精度の変動として捉え，これをグラフに表して観察することが一般に行なわれる．本研究でもその手段を使って学習が正常に行なわれているかどうかの判断，学習を終了させるかどうかの判断，パラメータ設定をどうするかの判断など，さまざまな判断を行なっている．

この観測を行なっていると急に認識精度が悪くなる現象にしばしば出会うが，本節ではこの現象について報告する．ここではこの現象を「崩壊」と呼ぶ．

学習履歴

図 6.29 は「ボポ」の学習経過を示す図である．横軸が学習回数，縦軸が認識精度である．実線が評価用データ，破線が学習用データでの結果である．学習用では認識精度は 5000 回前後まで順調に上昇している．この図の外側になるが，その後，10000 回程度まで若干上昇しているものの認識精度の向上はほとんど停止状態になっている．一方，評価用データでは 1500 回でほぼ認識精度の上昇は終り，3000 回前後でピークとなった後，なだらかに下降している．1000 回直前で学習用と評価用がクロスしているが，これは学習前の認識精度が，学習用

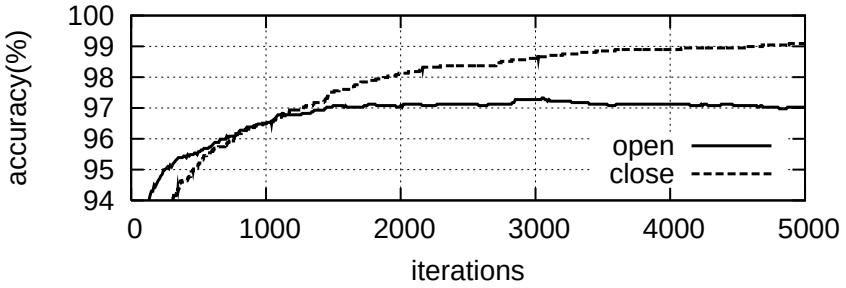


図 6.29: 学習履歴の例「ボポ」

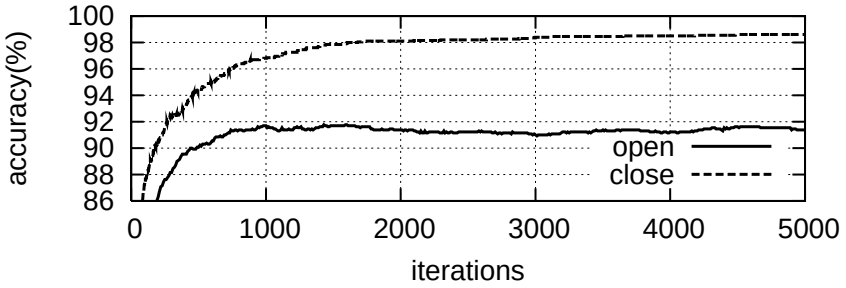


図 6.30: 学習履歴の例「間問」

79.6%，評価用 84.1%と評価用の方が高く，一方で学習後は学習用の認識精度の方が高くなるためであり特に意味はない．

図 6.30 は「間問」の学習の経過を示す図であり「ボポ」と同様の結果となっている．

学習崩壊

高速に学習させるためには学習強度を強く，すなわち FML の更新式におけるの ε を大きくする必要があるが，逆に学習強度を強くすると崩壊が起こる．図 6.31 の実線は「ボポ」の学習で崩壊した例である．これは図 6.29 の学習強度の 25 倍で学習させた時のものであり，同じ図にプロットしている normal の破線は図 6.29 の学習履歴カーブを学習回数ベースで横方向に 1/5 に縮め，重ねて表示したものである．

学習強度の強いケースでは認識精度がいきなり大きく落ち込み，また急回復するとしばらくそのまま学習が続き，また急落するという現象を繰り返している．この例では正常に学習しているケースがちょうど繰り返し崩壊の包絡線のようになっているところが興味深い．しかも 900 回あたりで崩壊が終り，その後は何事も無かったかのように順調に学習を進めている．

一方，図 6.32 は「間問」の学習で崩壊した例である．これは図 6.30 の学習強度の 25 倍で学習させたもので，図の normal の破線は図 6.30 のカーブを 1/10 に縮めたものである．こちらの崩壊は図 6.31 とは異り，学習スタート後すぐに学習ペースが落ちて，突然の崩壊の後，細かい振動が続いて 250 回あたりから大きな周期の振動が加わっている．図 6.31 とは違って normal の認識精度のラインまでには至らない．

図 6.33 にまた違った崩壊パターンの例を示す．これは手書数字認識に本手法を適用した場合の崩壊の例である．この例では崩壊後に暫くフラットな低迷期が続き，その後 1600 回あたりで急回復して，その後かなりインターバルの長い繰り返し崩壊が現れている．

この崩壊現象はフィルタマスク学習だけでなく LVQ でも見られる現象である．

このように学習強度をあげすぎた場合の崩壊現象が幾つか観察された．この崩壊現象を避けるためには学習強度を下げれば良いのであるが，学習強度を下げすぎると学習速度が遅くなりすぎて，あたかも学習が進んでいないかのように見えてしまうこともあるので注意が必要である．

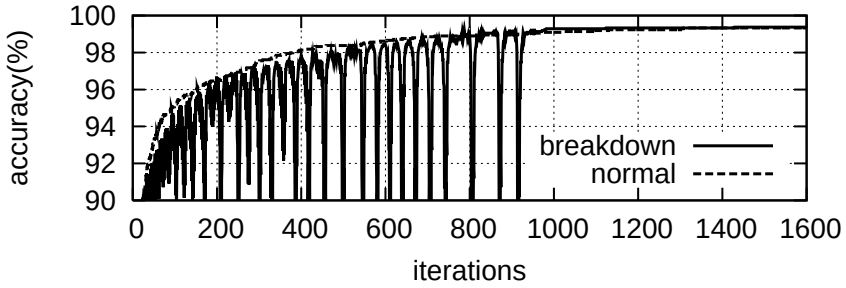


図 6.31: 学習崩壊の例,「ボポ」

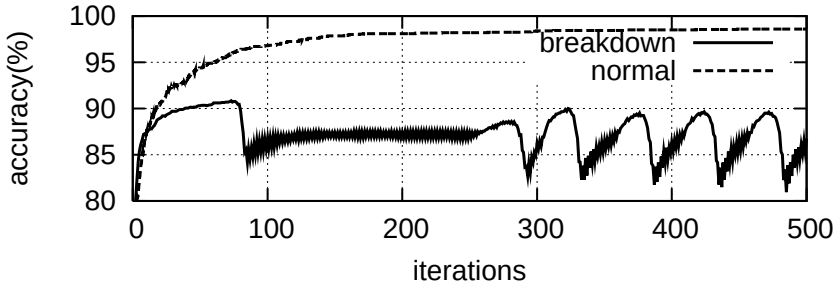


図 6.32: 学習崩壊の例,「間問」

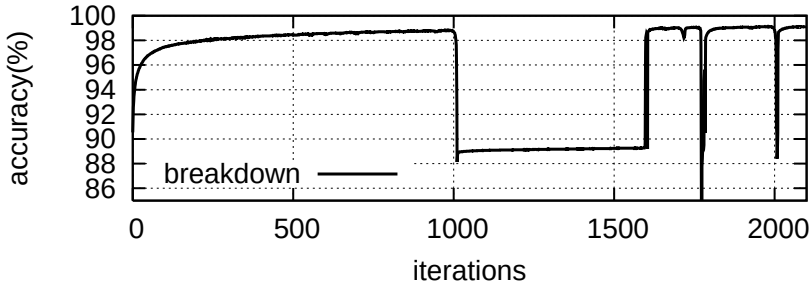


図 6.33: 学習崩壊の例，数字

学習崩壊の原因とその解決策

ここで述べた学習崩壊の現象は LVQ でも見られる現象で，そもそもこの種の学習方式，特に NN では良く見られる現象のようである．しかしながらこのような現象について詳しく書かれた文献あまり多くないようで，この現象は単に学習時のちょっとしたノウハウとしてあまり重要視されていない可能性がある．

しかしながら仮に学術的に意味が無いとしても，この問題は実用上は重視すべき課題である．以下に説明するように，この学習崩壊は学習強度を決めるパラメータ，本報告で言えば ε を下げることによって回避することができるが，そのことによって今度は逆に学習に時間がかかるようになり別の問題が出てくる．実際，場合によっては学習に極めて多大な時間がかかるケースがあり，そのようなケースで学習強度を下げると学習が全く進まなくなり，その結果，学習が本質的に駄目なのか，あるいは単に学習が遅いだけなのかを見分けられなくなる．このことは実用での辞書開発や学習方式改良を実施する上で深刻な障害となる．

この学習崩壊について LVQ で起きた学習崩壊を詳しく分析したところ，その学習崩壊が起きている付近で学習が激しく振動していたことが分かった．この振動を模式図で表したものが図 6.34 である．この図の縦に斜めの直線群が評価関数の等高線で，中央の値が低く外側の値が高いとしたものである．また最適化プロセスとしては，評価関数の値の低い方へ降下法によって下がっていくシチュエーションを想定している．仮に A 地点から

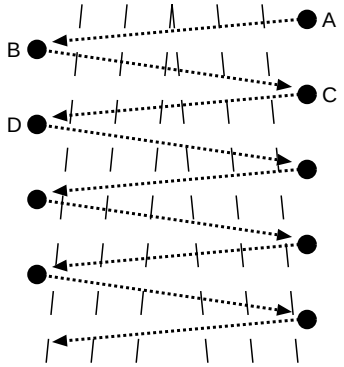


図 6.34: 学習振動の例

降下するとしてどうなるかを図示したのが横方向の点線の矢印である．この評価関数は中央を底にした谷になっており，その谷が図の下方向に向かって開いている．両サイドの崖の傾斜は等しいとする．こうすると降下法における移動方向は等高線に垂直，移動量は崖の傾きに比例するので一定となる．この例で最初の移動が点 A から点 B だったとすると以下 C，D，と図のような振動が起こるようになることが分かる．なお，ここではこの A〜D を探索点と呼ぶことにする．

このような事実は文献 [22] に記述があり，本実験で用いた LVQ の実験でも起きている．従って，このことは FML の実験でも同様であると推定している．なおこの問題を本研究の一環として LVQ で調査したが，それについては付録 B で述べた．

この図のように振動していても結果的に探索点が最適値の位置に近付いてくれれば問題無いのであるが，崩壊が起こると探索点が何処か遠くの，望ましくない場所に飛んで行ってしまうことがあるので問題なのである．従って道を外れないように学習強度を下げ，それによって振動しつつもゆっくり最適値に向かって歩んでくれるようにする．なお，ここで図 6.34 の例では学習強度を下げても振動自体は収まらず，振幅が減るだけなことは容易に分かると思う．実際の例でも学習強度を下げることによって崩壊を避けることができるが，振動自体は収まっていないと思われる．

さて，ここで問題となるのは学習強度を下げると学習に時間がかかるようになることである．この対策として学習が比較的起きやすい最初の段階では学習強度を低く，学習が進んだ段階で学習強度を上げる方法がある．実際に本研究での実験のいくつかではそのような手段を取った．しかしながら本研究は基礎段階の実験なので，できるだけ複雑な方法は避けるというポリシーから，学習強度や窓関数のパラメータについてはできるだけ固定値を用いるようにした．

文献 [22] には学習振動の解決策として慣性項と言われる，過去の更新量の重み付き和を更新量から減算する提案がなされている．この手段は特に NN の学習では定番の手法として良く使われているようである．本研究では慣性項の導入が学習に悪影響があるリスクも考慮してその手段は取らなかった．ベースとなる実験データを集め，その結果をリファレンスにして慣性項を入れた実験を行ない，その悪影響を調査し，その上で導入をはかるべきだと考えたからである．

いずれにしても学習の高速化は急務であり，FML の性能を向上させていくための重要課題である．

第7章　結果と結論

本研究において構造解析的パターン認識，統計的パターン認識，類似文字識別方式のそれぞれにおいて新提案を行ない，それぞれの有効性を確認した．また実験を通して，これらの要素技術によって文字認識の高精度化が実現できる見通しも得ることができた．

以下では特に FML に関しての結論を述べる．

類似文字識別における FML の有効性は実験により確認することができた．さらにこの FML に「多面化」「多重化」の拡張を施し，その効果を実験的に調査した．その結果それぞれが有効な改良方法であることを確認した．

また FML によって生成されたフィルタを目視によって確認することにより，識別するために重要な部分すなわちパターンの違いが顕著な部分において学習が進むことが「多面化」「多重化」の拡張後でも確認することができた．

「ボポ」や「間問」の類似文字ペアに関しては従来方式よりも良好な実験結果を得ることができた．この結果に基づけば，FML をこのように特に類似した文字ペアに適用していくという方針を採ることで，FML 実用化のメドを立てることができた．

一方で,「門構え」の他流試合の実験では従来方式の性能を若干上回る結果が得られたものの，性能向上のノリシロとしてはまだ不十分と言える結果になった．その意味では今後のさらなる改良の余地が残されている．

また FML 研究によって特徴抽出パラメータの学習手法の研究開発の今後に見通しをつけることができた．今回の実験は濃度特徴をフィルタによって求める単純な方式をターゲットとしたが，今後はこの経験を生かすことによって，このような手法を実際のＯＣＲで用いられているような複雑な特徴抽出方式へ展開して行くことが期待できる．もちろん特徴抽出部以外の部分に適用していける可能性もある．

要するに評価関数の設定方法，降下法のコントロール方法，パラメータ設定のやり方，実際の学習システムの構築方法などに数多くの知見と経験を得たので，これをベースにパターン認識における様々な学習問題に取り組むことができるということである．

これらをまとめると次のようになる．

（1）「ボポ」や「間問」の実験では実用性が見えた
FML をこのような，特に類似した文字ペアに適用していくことで実用化のメドが立った．

（2）「門構え」についてはさらなる改良が必要
他流試合の実験で従来方式を若干上回る認識精度が得られた．また今後のさらなる改良に向けての知見も得られた．

（3）マスクの可視化によって FML の有効性を確認
視覚的に FML の学習の進む様子を確認することができた．

（4）マスクの非一様性の効果を確認
初期マスクに一様マスクを用い，これを学習することで非一様マスクを作り出した．その結果かなりの性能向上が実現できたので，その意味で非一様マスクの効果を確認することができた．

（5）多面化，多重化の効果を確認

多面化，多重化の改良で順次認識精度が向上していくことを確認した．さらに一層の改良策を加えていくことで，さらなる改良が期待できる．

（6）特徴抽出部などに対する学習方式を今後開発していくための足掛かりとなる
本研究で得られた知見は今後の特徴抽出部だけでなく，文字認識システムの中の様々な要素に対する学習方式を開発していく上でのノウハウとして役立つ．

最後に本研究に続く今後の課題を以下にあげて本論文の終りとする．

（1）構造解析的パターン認識＋逐次学習
構造解析的パターン認識については，学会・メーカーを通してその研究はスローダウンしている．あまり進展が見られない．これに対して構造解析的パターン認識に学習を導入することが突破口になるのではないかと期待される．その学習に関しては今回の研究開発経験が生かせると考えられる．

（2）統計的パターン認識＋逐次学習
本研究で取り組んだ統計的パターン認識はパラメトリック型と言われるもので，逐次学習を行なわないタイプのものである．このタイプの方式についてはその性能向上は頭打ち傾向にあると思われる．これに逐次学習を導入するとノンパラメトリック型になってしまうが，そうした試みを検討する必要あるのではないかと思われる．過去にその方向の研究が幾つかあるが，さらに深いレベルで逐次学習を導入する必要がある．

（3）文字認識システム全体としての性能向上
本研究で開発した方式を含めて文字認識システム全体を構築し，性能評価・実用化をはかることは言うまでもない．

次以降は FML に関する課題である．

（4）学習パラメータの自動設定
学習で用いたパラメータ，たとえば ε_r，ε_f，δ などの値の決定を予備実験を通して決めねばならず，決定するのに時間がかかる．これらを自動的に決める手法は是非，開発したいアイテムである．

（5）FML のさらなる性能向上
類似文字識別以外のケースは本研究の範囲外であるが，予備的に行なった数字の実験では認識部として単純なユークリッド距離を用いたこともあって，認識精度としては実用レベルの精度は出ていない．このような類似文字識別でないケースにおいて，OCR で実用化されているような高度な認識方式と FML を組合わせて十分な性能が出せるのかどうかは今後の課題である．

（6）学習崩壊に対する対策
学習の途中で繰り返し崩壊現象が見られ，これは実用化時には問題になる可能性がある．学習プロセスの状況に関しては可視化ツールを用いたさらなる調査が必要である．

（7）学習の高速化
今回の実験ではかなりの学習時間を要したとは言え，何とか実験を終らせることができた．今後さらにデータ量が増えたりシステム規模が増えたりすることを考えると，一層の高速化の検討と改良が必要になる．

（8）文字認識以外への展開
文字以外の応用も是非，取り組みたい課題である．この場合に課題が類似性の識別であれば相応の成果が期待できる．

付 録 A　単純類似度型 FML における正規化と正準化

ここでは単純類似度を認識部に用いた FML について，その更新式を正規化と正準化を考慮した場合について求めた．ここでは，これを求めるための式変形について詳細に記述している．

A.1　正規化考慮の場合

まず正規化を考慮するケースについて計算する．この時の類似度は次の形である．

$$S = \frac{(x,\varphi)^2}{\|x\|^2\|\varphi\|^2} = (x,\varphi)^2\frac{1}{\|x\|^2}\frac{1}{\|\varphi\|^2} = (\sum_i x_i\varphi_i)^2\frac{1}{\sum_i x_i^2}\frac{1}{\sum_i \varphi_i^2}. \tag{A.1}$$

【 $\Delta\varphi$ の計算】

まず $\frac{\partial S}{\partial \varphi_i}$ を求める．

$$\begin{aligned}
\frac{\partial S}{\partial \varphi_i} &= 2(x,\varphi)x_i\frac{1}{\|x\|^2}\frac{1}{\|\varphi\|^2} + (x,\varphi)^2\frac{1}{\|x\|^2}(-\frac{1}{\|\varphi\|^4})(2\varphi_i) \\
&= 2(x,\varphi)\frac{1}{\|x\|^2}\frac{1}{\|\varphi\|^2}\{x_i + (x,\varphi)(-\frac{1}{\|\varphi\|^2})(\varphi_i)\} \\
&= 2(x,\varphi)\frac{1}{\|x\|^2}\frac{1}{\|\varphi\|^2}\{x_i - (x,\frac{\varphi}{\|\varphi\|})\frac{\varphi_i}{\|\varphi\|}\} \\
&= 2(\frac{x}{\|x\|},\frac{\varphi}{\|\varphi\|})\frac{1}{\|\varphi\|}\{\frac{x_i}{\|x\|} - (\frac{x}{\|x\|},\frac{\varphi}{\|\varphi\|})\frac{\varphi_i}{\|\varphi\|}\}.
\end{aligned} \tag{A.2}$$

更新式は式（4.9）であるが，ここでは S は類似度なのでそのの符号を逆にしたもの，すなわち，

$$\Delta\Lambda_p = \pm\varepsilon l'(d)\frac{\partial S_c}{\partial \Lambda_p}, \tag{A.3}$$

が更新式となる．よって，

$$\Delta\varphi_i = \pm 2\varepsilon_r l'(d)(\frac{x}{\|x\|},\frac{\varphi}{\|\varphi\|})\frac{1}{\|\varphi\|}\{\frac{x_i}{\|x\|} - (\frac{x}{\|x\|},\frac{\varphi}{\|\varphi\|})\frac{\varphi_i}{\|\varphi\|}\}. \tag{A.4}$$

これをベクトル表現して，

$$\Delta\varphi = \pm 2\varepsilon_r l'(d)(\frac{x}{\|x\|},\frac{\varphi}{\|\varphi\|})\frac{1}{\|\varphi\|}\{\frac{x}{\|x\|} - (\frac{x}{\|x\|},\frac{\varphi}{\|\varphi\|})\frac{\varphi}{\|\varphi\|}\}, \tag{A.5}$$

となる．さらに，

$$y = \frac{x}{\|x\|},\ \ \psi = \frac{\varphi}{\|\varphi\|}, \tag{A.6}$$

と置いて以下のようになる．

$$\Delta\varphi = \pm 2\varepsilon_r l'(d)\frac{1}{\|\varphi\|}(y,\psi)\{y-(y,\psi)\psi\}. \tag{A.7}$$

【 ΔF の計算】

次に $\frac{\partial S}{\partial F_{ij}}$ を求める．$x_i = \rho(\hat{x}_i)$，$\hat{x}_i = \sum_j F_{ij} z_j$ なので，

$$\begin{aligned}
\frac{\partial S}{\partial F_{ij}} &= 2(x,\varphi)\varphi_i \frac{\partial x_i}{F_{ij}}\frac{1}{\|x\|^2}\frac{1}{\|\varphi\|^2} + (x,\varphi)^2(-\frac{1}{\|x\|^4})(2x_i)\frac{\partial x_i}{F_{ij}}\frac{1}{\|\varphi\|^2} \\
&= \{2(x,\varphi)\varphi_i \frac{1}{\|x\|^2}\frac{1}{\|\varphi\|^2} + (x,\varphi)^2(-\frac{1}{\|x\|^4})(2x_i)\frac{1}{\|\varphi\|^2}\}\frac{\partial x_i}{F_{ij}} \\
&= 2(x,\varphi)\frac{1}{\|x\|^2}\frac{1}{\|\varphi\|^2}\{\varphi_i - (x,\varphi)(\frac{1}{\|x\|^2})x_i\}\frac{\partial x_i}{F_{ij}} \\
&= 2(x,\varphi)\frac{1}{\|x\|^2}\frac{1}{\|\varphi\|^2}\{\varphi_i - (x,\varphi)(\frac{1}{\|x\|^2})x_i\}\frac{\partial x_i}{\partial \hat{x}_i}\frac{\partial \hat{x}_i}{F_{ij}} \\
&= 2(x,\varphi)\frac{1}{\|x\|^2}\frac{1}{\|\varphi\|^2}\{\varphi_i - (x,\varphi)(\frac{1}{\|x\|^2})x_i\}\rho'(\hat{x}_i)z_j.
\end{aligned} \tag{A.8}$$

更新式は式（A.3）から，

$$\Delta F_{ij} = \pm 2\varepsilon_f l'(d)(x,\varphi)\frac{1}{\|x\|^2}\frac{1}{\|\varphi\|^2}\rho'(\hat{x}_i)\{\varphi_i - (x,\varphi)(\frac{1}{\|x\|^2})x_i\}z_j, \tag{A.9}$$

となる．$\rho'(\hat{x})$ を $\rho'(\hat{x}_i)$ を対角に並べた対角行列を表すとして，

$$\begin{aligned}
\Delta F &= \pm 2\varepsilon_f l'(d)(x,\varphi)\frac{1}{\|x\|^2}\frac{1}{\|\varphi\|^2}\rho'(\hat{x})\{\varphi - (x,\varphi)(\frac{1}{\|x\|^2})x\}z^T \\
&= \pm 2\varepsilon_f l'(d)\rho'(\hat{x})\frac{1}{\|x\|^2}\frac{1}{\|\varphi\|^2}(x,\varphi)\{\varphi - (x,\varphi)(\frac{1}{\|x\|^2})x\}z^T,
\end{aligned} \tag{A.10}$$

である．

ここで $\{\varphi_i - (x,\varphi)(\frac{1}{\|x\|^2})x_i\}z_j$ が $\{\varphi - (x,\varphi)(\frac{1}{\|x\|^2}x\}z^T$ になることは式（4.21）の所で説明した通りである．

さらに上の式を変形し，

$$\begin{aligned}
\Delta F &= \pm 2\varepsilon_f l'(d)\rho'(\hat{x})\frac{1}{\|x\|}(\frac{x}{\|x\|},\frac{\varphi}{\|\varphi\|})\{\frac{\varphi}{\|\varphi\|} - (\frac{x}{\|x\|},\frac{\varphi}{\|\varphi\|})\frac{x}{\|x\|}\}z^T \\
&= \pm 2\varepsilon_f l'(d)\rho'(\hat{x})\frac{1}{\|x\|}(y,\psi)\{\psi - (y,\psi)y\}z^T,
\end{aligned} \tag{A.11}$$

が最終的に得られる．

A.2 正準化考慮の場合

次に正準化を考慮した場合について計算する．ここでは入力パターンを z，辞書ベクトルを ζ とし，

$$\begin{aligned}
&y = \frac{x}{\|x\|}, \quad x = v - \frac{1}{n}(v,\vec{1})\vec{1}, \quad v = \rho(\hat{v}), \quad \hat{v} = Fz, \\
&\psi = \frac{\varphi}{\|\varphi\|}, \quad \varphi = \zeta - \frac{1}{n}(\zeta,\vec{1})\vec{1},
\end{aligned} \tag{A.12}$$

とする．

【 $\Delta\varphi$ の計算】

正規化ありの時と同様に式（A.1）により更新量を求める．ここで，

$$\varphi_i = \zeta_i - \frac{1}{n}\sum_h \zeta_h, \tag{A.13}$$

なので,

$$\frac{\partial \varphi_i}{\partial \zeta_k} = \delta(i,k) - \frac{1}{n}, \tag{A.14}$$

である．これを用いて,

$$\frac{\partial S}{\partial \zeta_k} = \sum_i \frac{\partial S}{\partial \varphi_i}\frac{\partial \varphi_i}{\partial \zeta_k} = \sum_i \frac{\partial S}{\partial \varphi_i}(\delta(i,k) - \frac{1}{n}) = \frac{\partial S}{\partial \varphi_k} - \frac{1}{n}\sum_i \frac{\partial S}{\partial \varphi_i} = \frac{\partial S}{\partial \varphi_k} - \frac{1}{n}(\frac{\Delta S}{\Delta \varphi}, \vec{1}), \tag{A.15}$$

となる．なおこの式で $\frac{\Delta S}{\Delta \varphi}$ は $\frac{\partial S}{\partial \varphi_i}$ を並べたベクトルである．

よって更新式は式（A.3）から,

$$\Delta \zeta_k = \pm \varepsilon_r l'(d)\{\frac{\partial S}{\partial \varphi_k} - \frac{1}{n}(\frac{\Delta S}{\Delta \varphi}, \vec{1})\}, \tag{A.16}$$

となる．これをベクトル表現すると,

$$\Delta \zeta = \pm \varepsilon_r l'(d)\{\frac{\Delta S}{\Delta \varphi} - \frac{1}{n}(\frac{\Delta S}{\Delta \varphi}, \vec{1})\vec{1}\}. \tag{A.17}$$

次に $\frac{\partial S}{\partial \varphi_i}$ を計算する．これは式（A.2）と同じなので,

$$\frac{\partial S}{\partial \varphi_i} = 2(\frac{x}{\|x\|}, \frac{\varphi}{\|\varphi\|})\frac{1}{\|\varphi\|}\{\frac{x_i}{\|x\|} - (\frac{x}{\|x\|}, \frac{\varphi}{\|\varphi\|})\frac{\varphi_i}{\|\varphi\|}\} \tag{A.18}$$

$$= 2(y,\psi)\frac{1}{\|\varphi\|}\{y_i - (y,\psi)\psi_i\}, \tag{A.19}$$

となる．ゆえに,

$$\frac{\Delta S}{\Delta \varphi} = 2\frac{1}{\|\varphi\|}(y,\psi)\{y - (y,\psi)\psi\}, \tag{A.20}$$

である．ここで，$(y,\vec{1}) = 0$ ， $(\psi,\vec{1}) = 0$ を使うと,

$$(\frac{\Delta S}{\Delta \varphi}, \vec{1}) = 2\frac{1}{\|\varphi\|}(y,\psi)\{(y,\vec{1}) - (y,\psi)(\psi,\vec{1})\} = 0, \tag{A.21}$$

となるので,

$$\Delta \zeta = \pm \varepsilon_r l'(d)\frac{\Delta S}{\Delta \varphi} = \pm 2\varepsilon_r l'(d)\frac{1}{\|\varphi\|}(y,\psi)\{y - (y,\psi)\psi\}, \tag{A.22}$$

が最終的に得られる．この式は正規化考慮型の式（A.7）と同じである．

【 ΔF の計算】

次に $\frac{\partial S}{\partial F_{ij}}$ を求める．

$$x_i = v_i - \frac{1}{n}\sum_h v_h, \tag{A.23}$$

なので,

$$\frac{\partial x_i}{\partial v_k} = \delta(i,k) - \frac{1}{n}, \tag{A.24}$$

である事を用い,

$$\begin{aligned}\frac{\partial S}{\partial F_{kj}} &= \sum_i \frac{\partial S}{\partial x_i}\frac{\partial x_i}{\partial v_k}\frac{\partial v_k}{\partial F_{kj}} = \sum_i \frac{\partial S}{\partial x_i}(\delta(i,k)-\frac{1}{n})\frac{\partial v_k}{\partial F_{kj}} = \{\frac{\partial S}{\partial x_k}-\frac{1}{n}\sum_i\frac{\partial S}{\partial x_i}\}\frac{\partial v_k}{\partial F_{kj}} \\ &= \{\frac{\partial S}{\partial x_k}-\frac{1}{n}(\frac{\Delta S}{\Delta x},\vec{1})\}\frac{\partial v_k}{\partial F_{kj}}, \end{aligned} \tag{A.25}$$

となる．この式で $\frac{\Delta S}{\Delta x}$ は $\frac{\partial S}{\partial x_i}$ を並べたベクトルである．

よって更新式は式（A.3）から，

$$\begin{aligned}\Delta F_{kj} &= \pm\varepsilon_f l'(d)\{\frac{\partial S}{\partial x_k}-\frac{1}{n}(\frac{\Delta S}{\Delta x},\vec{1})\}\frac{\partial v_k}{\partial F_{kj}} \\ &= \pm\varepsilon_f l'(d)\{\frac{\partial S}{\partial x_k}-\frac{1}{n}(\frac{\Delta S}{\Delta x},\vec{1})\}\frac{\partial v_k}{\partial \hat{v}_k}\frac{\partial \hat{v}_k}{\partial F_{kj}} \\ &= \pm\varepsilon_f l'(d)\{\frac{\partial S}{\partial x_k}-\frac{1}{n}(\frac{\Delta S}{\Delta x},\vec{1})\}\rho'(\hat{v}_k)zj \\ &= \pm\varepsilon_f l'(d)\rho'(\hat{v}_k)\{\frac{\partial S}{\partial x_k}-\frac{1}{n}(\frac{\Delta S}{\Delta x},\vec{1})\}zj, \end{aligned} \tag{A.26}$$

となる．なお，ここで $\hat{v}_k = \sum_j F_{kj}z_j$ を用いている．

これをベクトル表現すると，

$$\Delta F = \pm\varepsilon_f l'(d)\rho'(\hat{v})\{\frac{\Delta S}{\Delta x}-\frac{1}{n}(\frac{\Delta S}{\Delta x},\vec{1})\vec{1}\}z^T. \tag{A.27}$$

次に $\frac{\partial S}{\partial x_i}$ を計算する．

$$\begin{aligned}\frac{\partial S}{\partial x_i} &= 2(x,\varphi)\varphi_i\frac{1}{\|x\|^2}\frac{1}{\|\varphi\|^2}+(x,\varphi)^2(-\frac{1}{\|x\|^4})(2x_i)\frac{1}{\|\varphi\|^2} \\ &= 2(x,\varphi)\frac{1}{\|x\|^2}\frac{1}{\|\varphi\|^2}\{\varphi_i-(x,\varphi)(\frac{1}{\|x\|^2})x_i\} \\ &= 2(y,\psi)\frac{1}{\|x\|}\{\psi_i-(y,\psi)y_i\}, \end{aligned} \tag{A.28}$$

であるから，

$$\frac{\Delta S}{\Delta x} = 2\frac{1}{\|x\|}(y,\psi)\{\psi-(y,\psi)y\}, \tag{A.29}$$

となる．ここで，$(y,\vec{1})=0$ ，$(\psi,\vec{1})=0$ を使うと，

$$(\frac{\Delta S}{\Delta x},\vec{1}) = 2\frac{1}{\|x\|}(y,\psi)\{(\psi,\vec{1})-(y,\psi)(y,\vec{1})\} = 0, \tag{A.30}$$

となるので，

$$\begin{aligned}\Delta F &= \pm\varepsilon_f l'(d)\rho'(\hat{v})\frac{\Delta S}{\Delta x}z^T, \\ &= \pm\varepsilon_f l'(d)\rho'(\hat{v})2\frac{1}{\|x\|}(y,\psi)\{\psi-(y,\psi)y\}z^T, \\ &= \pm 2\varepsilon_f l'(d)\rho'(\hat{v})\frac{1}{\|x\|}(y,\psi)\{\psi-(y,\psi)y\}z^T, \end{aligned} \tag{A.31}$$

が最終的に得られる．この式は正規化考慮型の式（A.11）と同じ式である．$\rho(\hat{x})$ と $\rho(\hat{v})$ の記法上の違いはあるが，この場合 $\hat{x}$ と $\hat{v}$ はそれぞれ Fz であり同じものである．

このように正準化を考慮する場合も正規化を考慮する場合と形式的には同じ更新式になる．もちろん特徴抽出に正準化が入っている分，実際の計算は異る．

付 録B　LVQの発散問題

B.1　LVQの発散問題の概要

LVQの発散問題とは文献 [9] で指摘されている問題点であるが，この文献に関連した研究発表がなされる以前に筆者の同僚の１人である河村氏によっても発見されていたものである．

その問題とはシグモイドを損失関数として用いたLVQで２クラス識別を考えた時，標準パターンが学習と共に境界から離れて行ってしまうというという問題である．このために相違度の値が全体的に大きくなって学習強度や識別面の構成に影響を与えるというものである．例えば相違度が増加すると相違度差も大きくなって，シグモイドを微分した窓関数の裾野部分に行くパターンが多くなり，結果的に学習強度が学習の進行につれて下ると考えられる．一方で標準パターンと学習パターンとの距離 $(x-\varphi)$ は大きくなるので，この面での学習強度は大きくなると考えられる．

文献 [9] ではこの問題を解決するために新しい窓関数（以降，sato窓関数と呼ぶ）を提案している．この関数は標準パターンの位置を学習パターンの平均的なエリアから，はずれないように制御することで上記問題を回避するものである．

なお本研究ではこのsato窓関数は用いていない．この窓関数は学習パターンの平均的なエリアの影響を強く受ける手法なので，基礎実験としてはシンプルなシグモイド関数を使って認識系を単純にしておいた方が良いと考え，ここではシグモイドをそのまま使用した．窓関数を決めるパラメータ δ については，そのパラメータを振った実験に基づいて，崩壊の悪影響が出ないように適切な値を選んだ．

B.2　窓関数がstep関数の場合

この問題が何故起こるのかを図 B.1 を用いて説明する．この図はパターン空間を２次元として図示したもので，認識対象パターンをＡとＢの２カテゴリとしたものである．右側の楕円がパターンＡの存在エリア，左側の楕円の一部がパターンＢの存在エリアである．図中の a はＡの代表パターン，b はＢの代表パターンである．この a,b の垂直２等分線がパターンを識別する境界線となる．

ここでは窓関数がステップ関数の場合，すなわち誤読した時のみこの更新を行う場合を考える．図中の縦と横の縞模様の部分が誤読パターンの存在するエリアであり，左の縦縞がＡをＢに間違えたケースで，右の横縞がＢをＡと間違えたケースである．パターンＡに属する入力パターンが誤読した時，すなわち入力パターンが縦縞エ

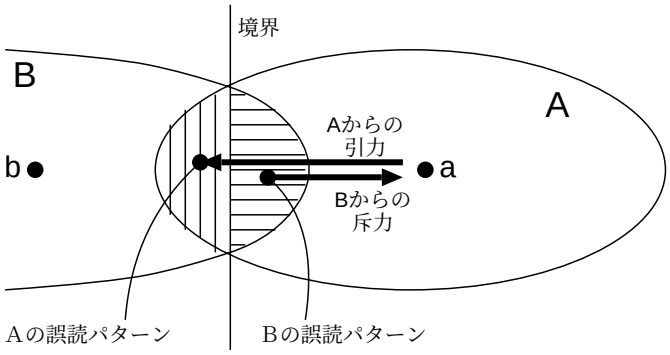


図 B.1: LVQの代表パターンが引かれ合う様子

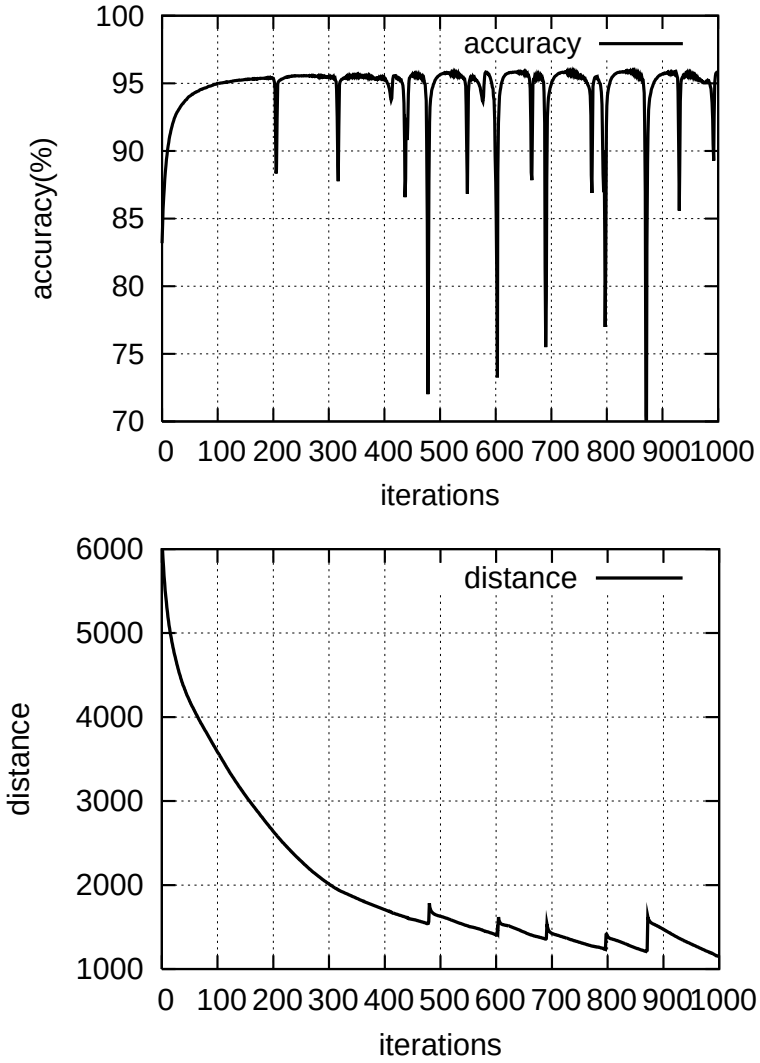


図 B.2: ステップ型窓関数での学習履歴（上），代表パターン間の距離が縮まる様子（下）

リアにある場合の学習則は入力パターンを x として，

$$a = a + \alpha(x - a), \quad b = b - \alpha(x - b),$$

であり，パターンBが誤読したの時，すなわち入力パターンが横縞エリアにある場合の学習則は，

$$a = a - \alpha(x - a), \quad b = b + \alpha(x - b),$$

となる．

この誤読パターンによって代表パターン a が変化する量は縦縞エリアのパターン x については更新量 $+\alpha(x-a)$ である．一方，横縞エリアのパターンについては $-\alpha(x-a)$ となる．前者の更新は a を縦縞エリアに引き寄せる引力になる．後者の更新は逆の斥力になる．この時の力に相当するものは $|x-a|$ であり，全体でならして見れば各縞エリア内の学習パターンの平均と代表パターンとの差である．各縞の学習パターン数が同じであれば，この差は縦縞からの引力の方が大きくなる．従って代表パターンは際限なく境界線上に向かって引き寄せられて行くことになる．

その様子を実験的に確かめた．図 B.2（上）が学習履歴を示す図である．通常は評価用データの結果を図示すべきであるが，この LVQ の発散問題に関する実験ではより基本的な情報を得る目的で，学習用データでの結果

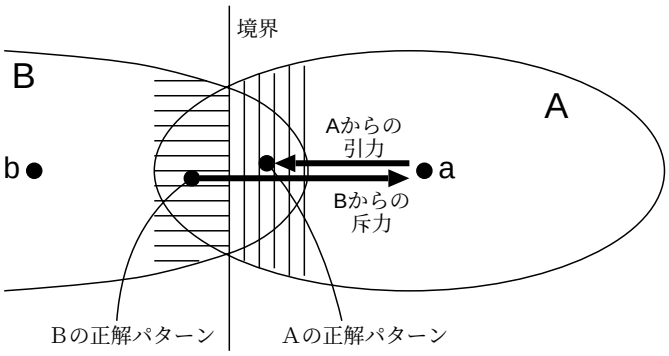


図 B.3: LVQ の代表パターンが反発する様子

を示している．縦軸が認識精度，横軸が学習回数，また学習パターンは手書き数字で「0」と「6」のみ学習したものである．図 B.2（下）は代表パターン同士の距離の履歴をプロットしたものである．代表パターン同士の距離が縦軸，横軸が学習回数である．これを見て分かるように距離は学習の回数が増すに従って減少している．なお，この間にも崩壊現象が起きており，図 B.2（下）で滑らかでない変化の見られる場所が図 B.2（上）で崩壊の起きている場所と一致している．

B.3 窓関数が sigmoid 関数の場合

一方，窓関数がシグモイドの場合には正読の場合にも更新が行われるので事情が異なっている．シグモイドでは誤読したパターンが存在するエリアからの力も働くが，誤読より正読のほうが遥かに多いので図 B.3 の縦縞や横縞で示したA及びBの正読エリアからの力のみを検討すれば良い．

更新に使われるパターンは境界線に近いところ，すなわち図 B.3 の縦縞と横縞のエリアにあり，図 B.3 の縦縞エリアについてはそのエリア内の入力パターンを x として， a に対して $+\alpha(x-a)$ の更新が行なわれる．また，逆に横縞のエリアにある入力パターンを x として， a に対して $-\alpha(x-a)$ の更新が行なわれる．この更新はステップの時に説明したのとは逆方向の力となることが図 B.3 から分かる．従って，代表パターンの動きはステップ型窓関数の時とは逆に，代表パターン同士の距離が離れる方向に変化して行く．

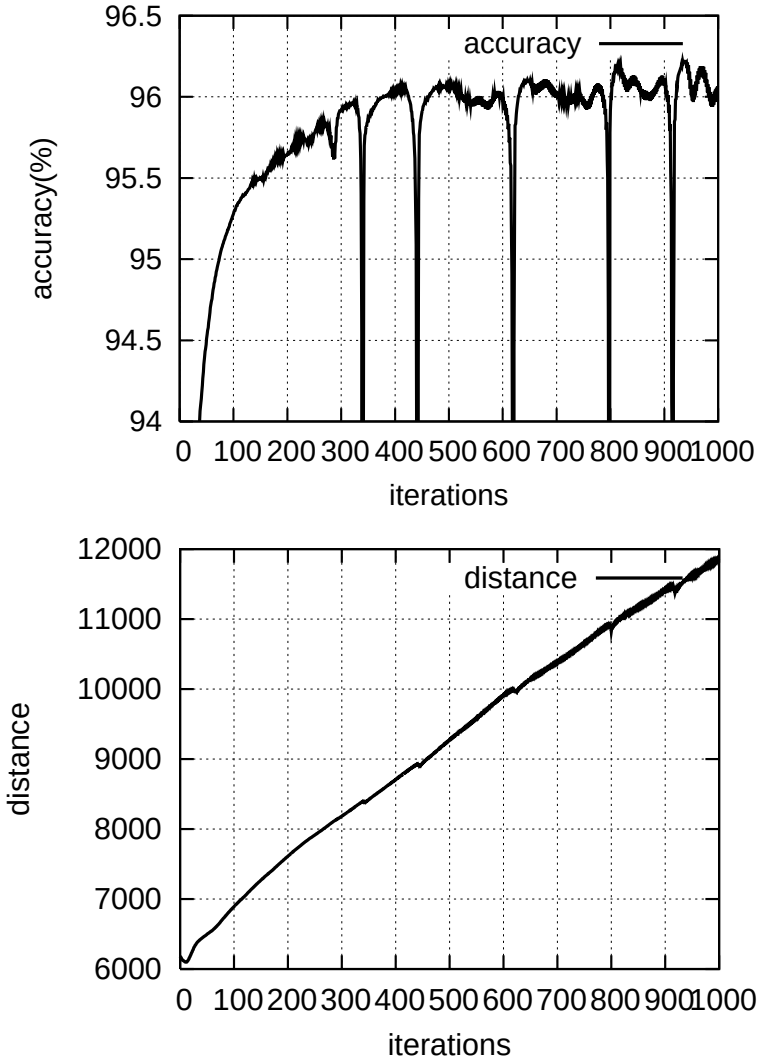


図 B.4: シグモイドの窓関数での学習履歴（上），代表パターン同士の距離が広がる様子（下）

実際にその様子を実験的に確かめた．図 B.4（上）が学習履歴，図 B.4（下）が距離の履歴である．実験条件は前の実験と同じく数字の「0」と「6」である．確かに学習回数に従って距離が増大して行くことが確認できる．

念のため更に学習回数を増やして 30,000 回にしたものが図 B.5 である．距離がこのあと頭打ちになるのかどうか分からないが，少なくともこの段階でも距離の増加は継続しつづけている．

なお，この図の認識精度でも崩壊が見えるが，この図の縦軸のスケールは前のものとは変えていて，この実験での実際の崩壊の程度は前の実験より小さい．距離の図における崩壊の影響が少ないのはそのためである．

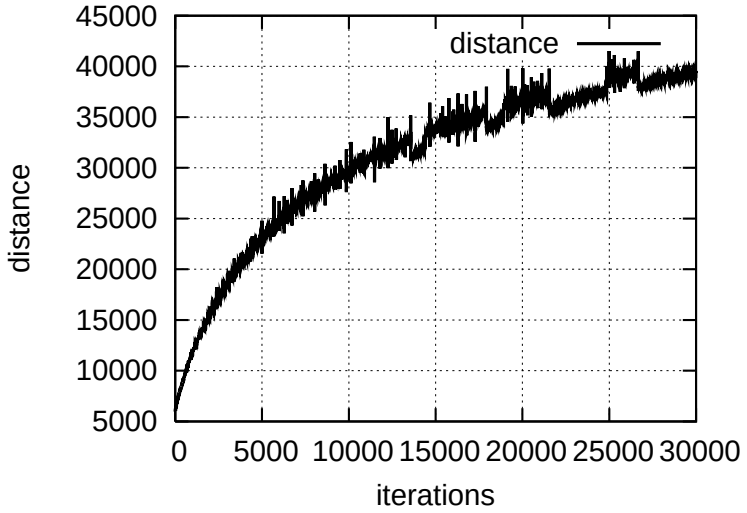


図 B.5: シグモイドの窓関数で代表パターン同士の距離が広がる様子（30,000 回学習）

B.4 窓関数が sato 窓関数の場合

次に sato 窓関数について調査する．sato 窓関数は次のようなものである．

入力パターンと正解カテゴリの相違度を s_{ok}, 不正解カテゴリの中で最小の相違度のものを s_{err} とした時，損失関数は通常 $l(s_{ok} - s_{err})$ であるが sato 窓関数では，

$$l\left(\frac{s_{ok} - s_{err}}{s_{ok} + s_{err}}\right), \tag{B.1}$$

で定義する．これを s_{ok} に含まれるパラメータ φ_i で偏微分すると，

$$2l'\left(\frac{s_{ok} - s_{err}}{s_{ok} + s_{err}}\right)\frac{s_{err}}{(s_{ok} + s_{err})^2}\frac{\partial s_{ok}}{\partial \varphi_i}. \tag{B.2}$$

また， s_{err} に含まれるパラメータ φ_i で偏微分すると，

$$2l'\left(\frac{s_{ok} - s_{err}}{s_{ok} + s_{err}}\right)\left(\frac{-s_{ok}}{(s_{ok} + s_{err})^2}\right)\frac{\partial s_{err}}{\partial \varphi_i}, \tag{B.3}$$

となる．相違度をユークリッド距離の 2 乗のケースで考えて，正解代表パターンのパラメータを φ_{ok}，不正解の方を φ_{err} とする．また，φ_{ok} や φ_{err} の更新量を $\Delta\varphi_{ok}, \Delta\varphi_{err}$ として，

$$\begin{cases} \Delta\varphi_{ok} = 4\varepsilon l'\left(\frac{s_{ok}-s_{err}}{s_{ok}+s_{err}}\right)\left[\frac{s_{err}}{(s_{ok}+s_{err})^2}\right](x - \varphi_{ok}) \\ \Delta\varphi_{err} = -4\varepsilon l'\left(\frac{s_{ok}-s_{err}}{s_{ok}+s_{err}}\right)\left[\frac{s_{ok}}{(s_{ok}+s_{err})^2}\right](x - \varphi_{err}), \end{cases} \tag{B.4}$$

となる．

次に，この関数で学習の動きがどうなるかを考察する．ここで，
$g_{ok} = x - \varphi_{ok}$, $g_{err} = x - \varphi_{err}, \hat{s} = (s_{ok} - s_{err})/(s_{ok} + s_{err})$ と置くと，
$s_{ok} = g_{ok}^2, s_{err} = g_{err}^2$ なので上記の式は次のようになる．

$$\begin{cases} \Delta\varphi_{ok} = 4\varepsilon l'(\hat{s}))\left[\frac{g_{err}^2}{(g_{ok}^2+g_{err}^2)^2}\right]g_{ok} \\ \Delta\varphi_{err} = -4\varepsilon l'(\hat{s}))\left[\frac{g_{ok}^2}{(g_{ok}^2+g_{err}^2)^2}\right]g_{err}. \end{cases} \tag{B.5}$$

この式を見て分かる通り，たとえば $\Delta\varphi_{ok}$ の更新式について言えば，窓関数がシグモイドの時の更新式に比べて $l'()$ の中を除いて，g_{ok} が

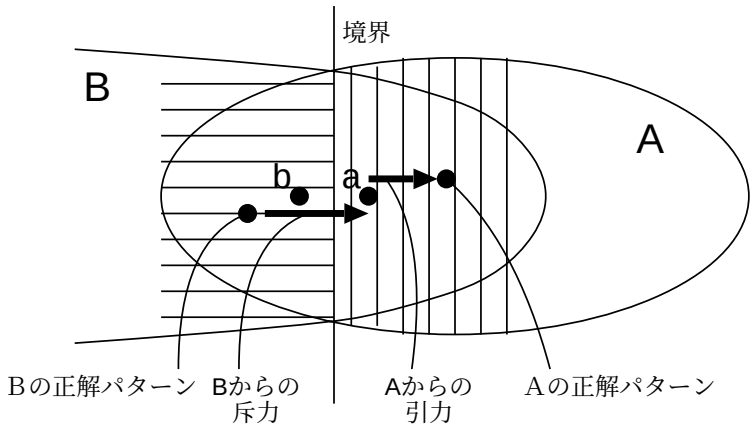


図 B.6: sato 窓関数で，代表パターンが境界付近に近いケース

$$\left(\frac{g_{err}^2}{(g_{ok}^2+g_{err}^2)^2}\right)g_{ok}, \tag{B.6}$$

に置き換わっている．ここで図 B.3 のケース，すなわち代表パターンが境界付近の学習に影響の大きいパターン・エリアから離れている場合を考える．この時の式（B.6）の分母は g の４乗の形であるのに対して， 分子は g の３乗の形であることから，この値は $|g|$ が大きくなると逆に小さくなることがわかる．ちなみに sigmoid の時の更新式は g そのものなので $|g|$ が大きくなると更新量は大きくなる．したがって，図 B.3 の「Aからの引力」の力は「B からの斥力」よりも，sigmoid の時とは逆に大きくなり，このケースでは反発ではなく，引き合うことがわかる．

一方，図 B.6 は代表パターンが境界付近に近いケースを図示したもので，この場合，代表パターン a への更新は「Aの正解パターンからの引力」も「Bの正解パターンからの斥力」も同じ方向であり，結果として代表パターンは境界付近から離れていくことになる．

結局，代表パターンは境界に近寄りすぎれば境界付近の学習に影響の大きいパターン・エリアの中央部分に押し戻されるし，またそのエリアから離れようとすると逆に引き戻されることが分かる．つまり，この窓関数は代表パターンを境界付近の学習に影響の大きいエリアに留まるように制御していることがわかる．

しかしながら，このような制御が必ずしも最適な境界線を引くための学習を保証しているわけでは無い．学習は安定化することが予測されるが学習効果は分からない．窓関数の１つの選択枝として他の窓関数との併用，比較などを行って慎重に利用した方が良いと思われる．また内積型の類似度を用いた認識システムでは，そのままの式では適用できないのではないかと思われ，その点も注意が必要である．

実際に sato 窓関数での LVQ の学習状況を実験的に確かめた．図 B.7（上）が認識精度の履歴，図 B.7（下）が距離の履歴である．実験条件は step 関数と同じである．

ここではパラメータを変えて４回実験を行なった．学習回数は 30,000 回である．図の 0.02/150 などの表記はパラメータを表すもので，順に α, δ を示している．パラメータ α は学習強度で値が大きいほど学習強度が上る．一方 δ は窓関数のパラメータで値が大きいほど学習強度が下がる．

この図 B.7（上）の認識精度の履歴では，４回のうち α=0.05, δ=150 のケースは表示からはずしている．これは，このケースでは繰り返し崩壊が極端で図が真っ黒になってしまうためである．学習が崩壊している α=0.05, δ=150 以外の学習パラメータは α=0.02 で，上から順に（認識率の良い順）δ=150， 100，70 である．この中では α=0.02, δ=70 で繰り返し崩壊が見られる．この実験で，α を増やすだけで大きく崩壊したことが分かったので，sato 窓関数でも繰り返し崩壊が起こることが確認された．ただしシグモイドに比べて崩壊は起こりにくい．またパラメータによって得られる認識率も異なるのでパラメータ依存があることも分かる．

次に辞書間の距離を見て見よう．図 B.7（下）は距離の変遷の図で下側で崩壊しているのが α=0.05, δ=150 で，上３個がそれぞれ α 1=0.02， δ=150，100，70（右から順）である．δ が小さい程より早く距離が増大している．いずれの場合もほぼ同じ距離のところで停止していることが良く分かる．一方 α=0.05, δ=150 のケー

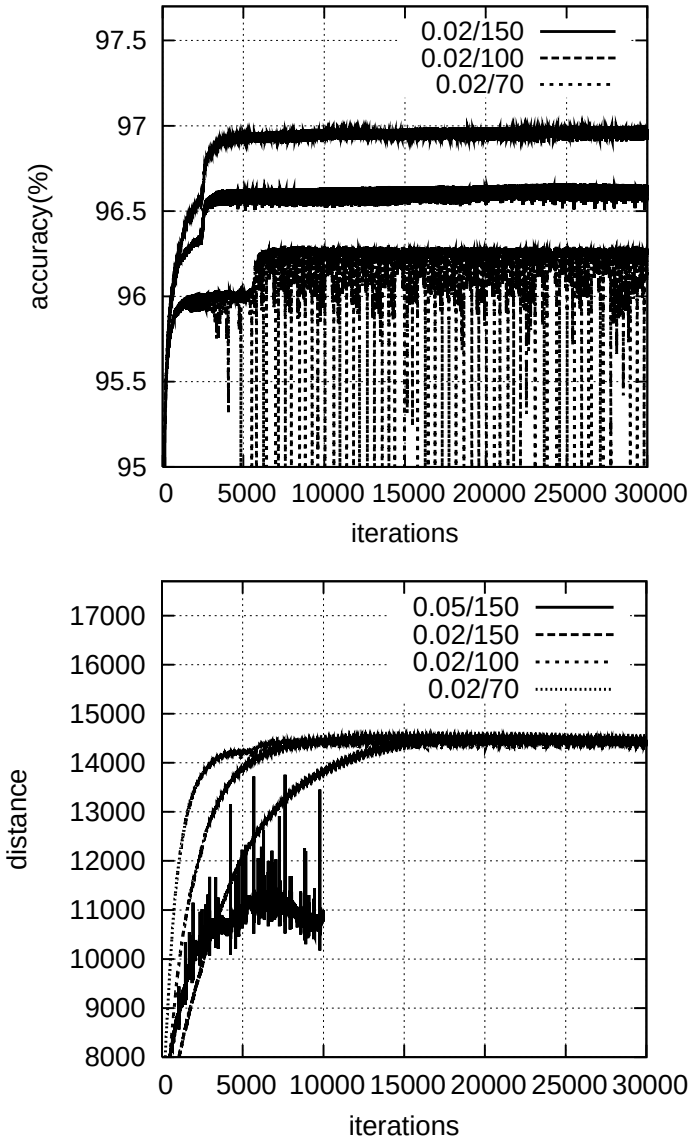


図 B.7: シグモイドの窓関数での履歴（上），代表パターン同士の距離が広がる様子（30,000 回学習）

スを見ると，学習強度が強すぎてその位置まで移動できないようだ.

付 録C　学習崩壊の観察

ここではFMLでの学習崩壊の原因を分析するための手段について述べる．一般にこのような崩壊や学習が進まない原因の１つに「強すぎる学習強度」があることは良く知られている．しかしながら実際の事例において本当にそうなのかどうかは可能であればチェックしておきたい．以下は本研究において開発した，そのための学習可視化技術の説明である．

C.1　降下法型学習プロセスの可視化

ここでは降下法による学習プロセスの可視化技術について述べる．これは一般的な降下法による最適化プロセスの可視化にも適用できるものであり，まず本節ではその一般論について説明していく．

降下法の対象となる評価関数をベクトル v を変数として $f(v)$ とする．このベクトルを降下法によって逐次求めたもののうち k 回目で求まったものを v_k と書き $v_r, v_{r+1}, \cdots, v_{r+s-1}$ における学習状況を観測したいとする．ベクトル v は一般には３次元以上の多次元空間の元なので，その空間を可視化することは難しい．そこで，この多次元空間を平面でスライスして学習状況を観測することを考える．このスライス平面を求めるために学習によって得られたベクトル v_k の足取りを追跡し，それらを良く近似する平面を求めてそれをスライス平面とする．そのためにベクトル $v_r, v_{r+1}, \cdots, v_{r+s-1}$ を用いて主成分分析を行ない，その第０固有ベクトルと第１固有ベクトルからスライス平面を構成する．

この固有ベクトルを φ_0, φ_1 とする．こうすると，

$$P_{xk} = (v_k, \varphi_0), \tag{C.1}$$

$$P_{yk} = (v_k, \varphi_1), \tag{C.2}$$

によって (P_{xk}, P_{yk}) なる点列がスライス平面上に得られる．すなわち v_k をこの２次元平面上にプロットできるのである．この点を結んだものが学習の足取りで，これは v が属する多次元空間を平面でスライスし，そこに v の足取りを投影したものである．仮に足取りが局所的に２次元平面上にほぼ乗っていたとすれば，一応，近似的に学習の足取りが見えていることになる．

次にこの平面を格子で区切り，各格子点を (x_i, x_j) とする．するとその位置の評価値は

$$e_{ij} = f(x_i\varphi_0 + y_i\varphi_1), \tag{C.3}$$

によって求めることができる．この値をスライス平面上の対応する位置の濃度として表示すれば多次元空間内の評価関数の値を平面でスライスした画像として見ることができる．評価関数の断層像である．

この画像の上にさきに求めた学習ベクトルの足取りをプロットすれば，どのような評価関数の上で学習がどのような動きをしているのかがグラフィカルに分かる．

次の節で実際にこの可視化をLVQに適用した例を示す．

C.2　LVQ可視化による学習崩壊の観察

ここで行なった実験対象は手書数字文字である．文字パターンを 14×14 の濃度特徴で特徴抽出し LVQ によって学習を行なった．図 C.1 はその学習履歴で繰り返し崩壊が起こっている．その最初の崩壊が学習回数 180 付近で，その拡大図を丸で囲んで図示している．

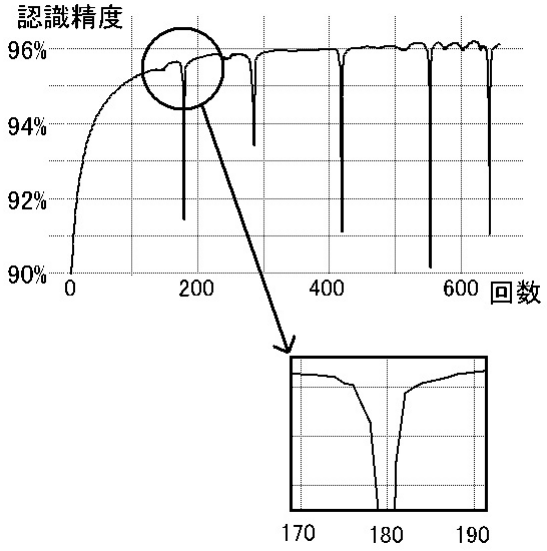


図 C.1: 手書数字の LVQ における学習履歴，繰り返し崩壊のケース

この状態で分析することも可能であるが，より詳細に分析するにはどのカテゴリで崩壊が起きているのかを調べる必要がある．そのための各学習回数における誤読の数をカテゴリごとに調査し，その状況を見た．その結果，このケースでは誤読数を縦軸，学習回数を横軸に取ると「7」と「9」のカテゴリでその誤読数が振動していることが分かった．学習回数の 175 から 184 までをチェックすると誤読数が学習１回ごとに交互に大きくなっていたのである．振動の原因はこのケースでは１０のカテゴリのうち２つのカテゴリのペアによるものであった．

この結果に基づいて，この「7」と「9」だけを学習に用いた LVQ の分析も行なったが，これは数字全体での分析と同じような結果であったので本報告では数字全体での分析結果について報告する．

この実験での v は辞書ベクトルを全部合わせたものであり，(パターン・ベクトルの次元）×（カテゴリ数）の高次元ベクトルである．スライス平面を構成するベクトル φ_0, φ_1 は学習の結果として求まる辞書ベクトルの系列から求めることができる．辞書ベクトルを全部合わせて新しいベクトル v を作成し，その系列から求めるのである．また学習の足取りもこの v を φ_0, φ_1 が作る２次元平面上に射影することでプロットすることができる．ここではこれを学習パスと呼ぶ．

一方，評価関数のスライス画像は実際の評価値を格子点上で求めれば描くことができる．評価値の計算式は与えられているので，辞書ベクトルを与えればその評価値は認識シミュレーションにより計算可能である．具体的には $x_i\varphi_0 + y_i\varphi_1$ が辞書ベクトルを合わせたベクトルなので，これをばらばらにして元の辞書に戻し，これに基づいた認識シミュレーションを行なうのである．その結果得られる相違度から評価値を求め，その位置の小領域をその求められた評価値に対応するの濃度で塗ることによってスライス画像を描画できる．

最終的に学習パスとスライス画像を重ねて表示する．

この結果得られた学習可視化画像を図 C.2 に示す．この図の左側のスライス画像は学習用パターンで求めた評価関数のスライス画像，中央が誤読率すなわち誤読数のスライス画像，右側が評価用パターンで求めた評価関数のスライス画像である．各画像上にプロットした学習パスはすべて同じであり，学習は左側から始まっている．右側まで行って若干，左方向に戻っている．これらの図の濃度は見やすいようにそれぞれ調整している．また，これら各ケースについて上から下にズームインして３つの異なるスケールのものを表示している．

これらを見ると，この程度に俯瞰したに場合には損失関数による評価関数や誤読率はあまり違いがない．また評価用パターンのスライス画像と学習用パターンのスライス画像の差も分からない．もちろん細かく見れば異なるのだが，この程度に俯瞰した形だとその違いが分からない．なお濃度については表示の時に調整しているので，ここではその違いは無視して良い．

これらの画像を見ると学習パスが振動している様子が良く分かる．図 6.34 と同じである．概念としては良く知られていることではあるが，このような実際の振動を可視化することはこれまであまり行なわれていない．この可視化技術は適切なパラメータの確認や学習の不具合のチェックなどに役立ち，また方式理解の助けにもなり実用化をめざした開発においては特に有用である．

図 C.3 の左側のスライス画像は図 C.2 と同じ条件のもので，この画像上に学習パスをプロットしはじめる学習回数 170 以降の学習強度のみを 0.0004 から 0.00035 に下げた可視化画像である．この画像のスケールは図 C.2 のもっともズームインした場合のものと同じで，当然スタート位置も左側の同じ位置である．見やすさのために線とドットだけ細く小さくしてある．この画像から学習強度を下げることによって振動が収まることが分かる．もっともこのケースでも右側では大きく振動し始めるので実際にはもっと学習強度を下げる必要がある．また学習強度を下げることで学習速度が落ちることも良く分かる．

図 C.3 の右側の画像も図 C.2 と同条件のもので，こちらは学習を安定化させるために学習回数 170 から慣性項を用いた学習に切替えたものである．慣性項のパラメータは文献 [22] の記号で $\alpha = 0.9$ である．図から学習が安定していることが分かる．

しかしながら慣性項なしと比べると学習パスがまったく異なり，その行き先が違っている可能性も見て取れて，慣性項を入れて良いのかどうかについては慎重さが必要だと思われる．

なお，これらのプロットにおいて各ベクトルのスライス平面からの距離は，おおむねこれらの振動の幅と同程度であった．その意味では，これは相当粗いプロットになっているが学習のおおまかな様子を見る意味ではこの程度で良いと思われる．

なお図 C.3 のスライス画像は図 C.2 のものと同じである．これは比較しやすさのためで，新たな学習パスから新たなスライス画像は作成しなかった．

C.3 評価関数の断面

前の節では学習パスを評価関数のスライス画像上でプロットしたが，ここではこのスライス画像の断面を見ることとする．図 C.4 は図 C.2 のズームインしていない一番上のスライス画像を x-y 平面上の関数と見て，この関数を $x = 0$ の位置で縦に切った時の断面をグラフ表示したものである．

図 C.4（A）の横軸はスライス画像の縦軸に対応し，縦軸は $x = 0$ の時の評価関数の値である．グラフの点線が窓関数のパラメータ δ を 100 とした時の関数値で，細い線は誤読個数である．この両者はほぼ一致しており，これは δ を大きくすると損失関数がステップ関数に近付き，評価値はすなわち誤読個数となることに対応している．一方，$\delta = 10$ とした時の評価関数はグラフの太い線で，誤読数の曲線を若干なまらせた形状であることがうかがえる．評価関数は δ が減ると値が大きくなるので，この太線は全体にバイアスされたかのように上に上がっているのである．

このグラフの $\delta = 100$ の細線と誤読個数の破線はほぼ重なっているように見えるが拡大してみると若干のずれがある．バイアス分を目視で除いて両曲線がなるべく重なるように表示したものが図 C.4（B）のグラフである．なお，このグラフでバイアス分，すなわち重ねるためにずらした量は 175 である．ここまで拡大するとバイアス分を除いても若干のずれがあることが分かる．

この時の評価用データでの評価関数はどうなっているのであろうか．それを図示したものが図 C.4（C）のグ

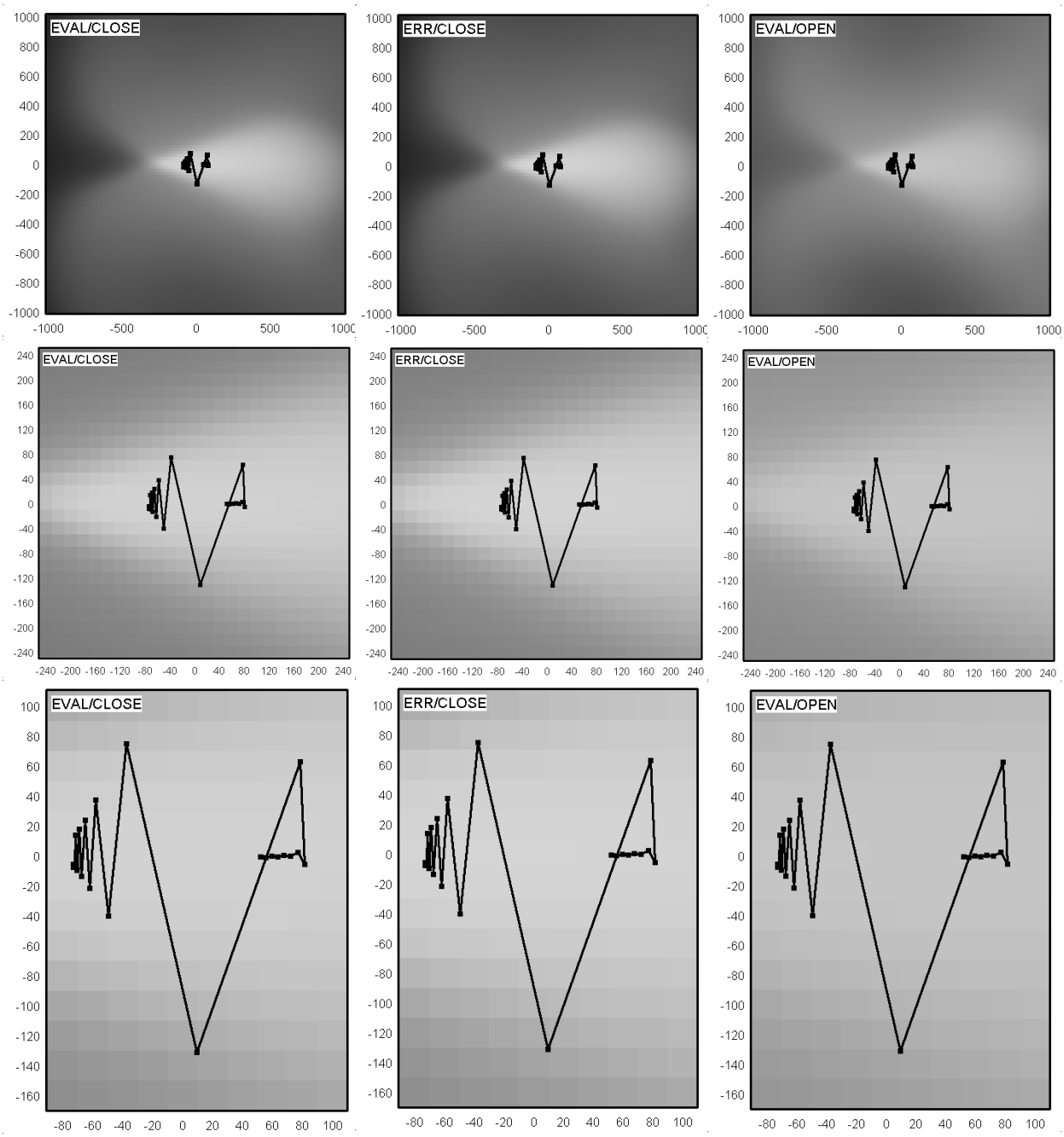
ラフで，学習用よりも評価関数値と誤読数の差が大きいことと，最小値を取る x の位置に若干のずれがあることも分かる．このずれが大きくて問題になるのが，いわゆる過学習の問題である．

これをさらに拡大したのが図 C.4（D）のグラフである．ここまで拡大すると誤読個数のグラフが階段関数になっていることがわかる．誤読個数は整数値なのでこれは当然である．損失関数に基づく評価関数は $\delta \to \infty$ の時に誤読数と一致するが，δ の値が小さくなるにつれて徐々になまってくる．そのことがこのグラフからも読み取れる．

パラメータ δ の値を大きくすると関数がガタガタになってトラップに捕まりやすくなり，要するにローカルミニマムに陥りやすくなり，一方で値が小さいと関数がなまりすぎて原型をとどめなくなり，最適値が何処だか分からなくなるということが，具体的にこうしたグラフを見ることで確認できる．

現実のこうした評価関数の形を見ると，その形が関数として本来の形に細かいノイズが乗ったものであることが分かる．このノイズ部分を取り除く手段が δ であり，その値によってどの程度までノイズを取り除くかを決めているのが δ の設定問題であると解釈することができる．

一般に δ の役割は「これが無いと損失関数がステップ関数になり，ステップ関数のフラットな部分での学習が行なわれなくなる」とか，あるいは「降下法における微分を可能ならしめるため」等と言われている．そのためにステップ関数の替わりにシグモイド関数を用い，δ によって学習強度を調節するというように説明される．しかしながら，このようなグラフを観察すると，δ の機能として上記のような評価関数のノイズ・リダクションとしての解釈も可能だと考えられる．



評価関数（学習用パターン）　誤読率（学習用パターン）　評価関数（評価用パターン）

図 C.2: 数字認識における LVQ での学習パス，上から下にズームイン（学習強度 $\alpha = 0.0004$）

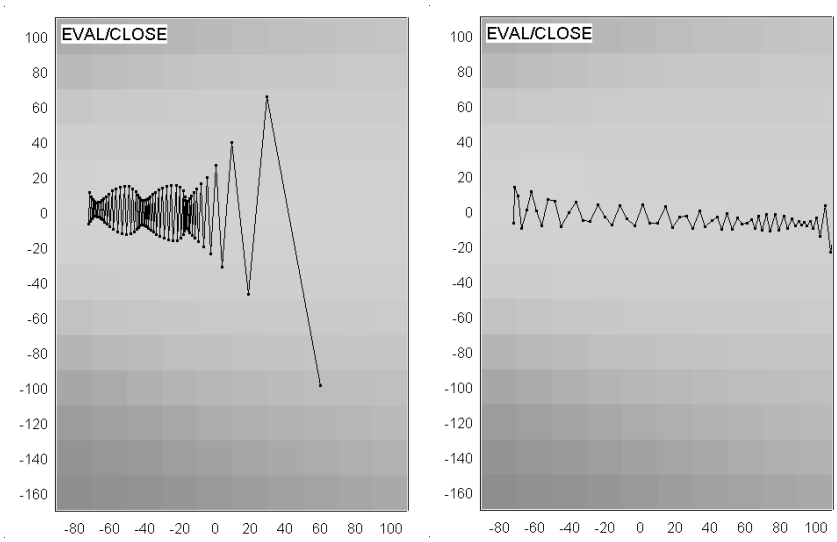


図 C.3: 数字認識における LVQ での学習パス，左は α =0.00035，右は慣性項有りの α =0.0004

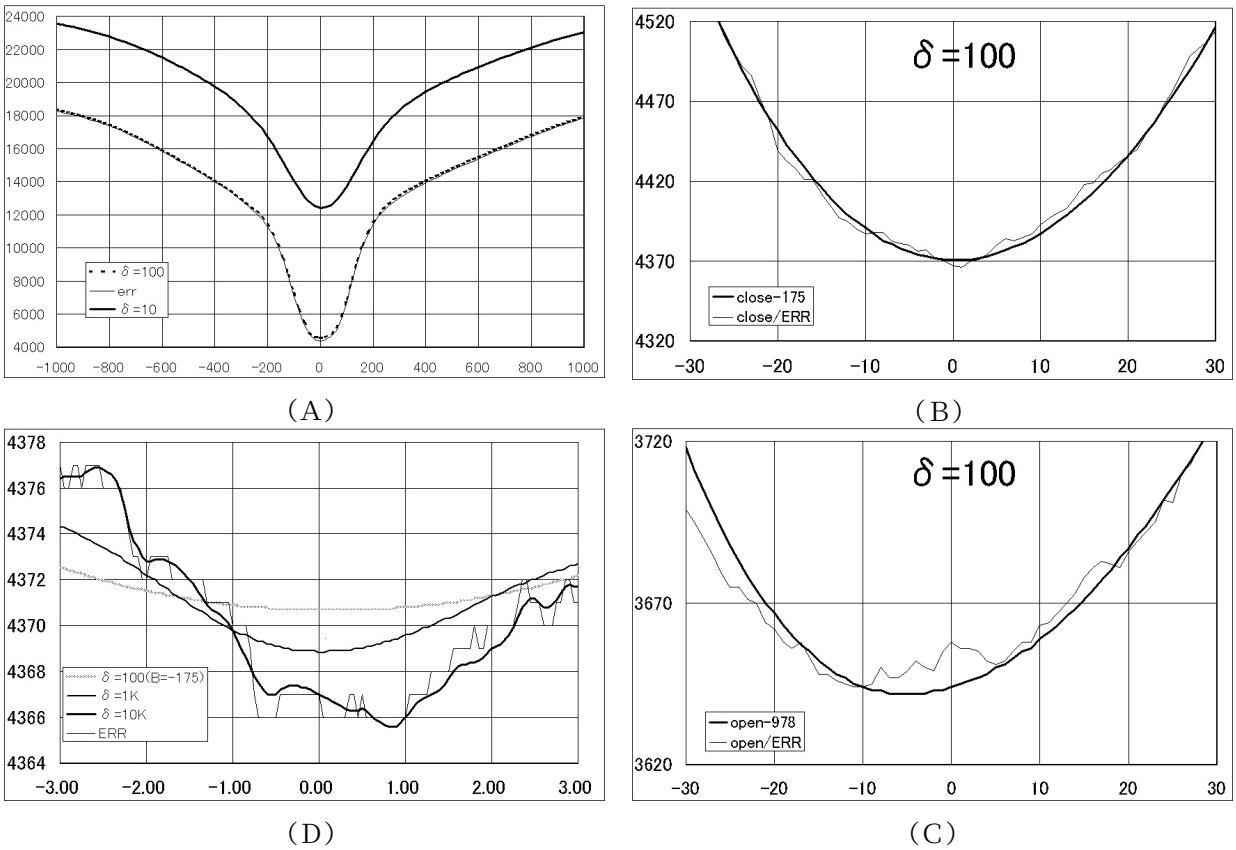


図 C.4: 評価関数の断面図の例

謝辞

研究および論文執筆にあたり御指導ご鞭撻頂いた東京工業大学の佐藤誠教授に感謝します．同教授には著者の学生時代から先輩として御指導ご助言を頂いて来ました．

また有益な議論と貴重なご意見を頂きました，東京工業大学の羽鳥好律教授，内川惠二教授，小林隆夫教授，熊澤逸夫教授に感謝します．

本研究の背景となるボケの理論やパターン認識理論については，著者が学部および修士課程時代の指導教官であった東京工業大学の飯島泰蔵名誉教授により御指導頂いたものであり，感謝いたします．

同じく元東京工業大学の小川英光教授からも特に学部および修士課程在学中に関してはお世話を頂き，本研究を進めるにあったっての数学的な基礎を教えて頂きました．

東京工業大学の学部時代の先輩であり，また東芝での先輩でもある前田賢一技監には一貫して御指導ご鞭撻，また議論の相手をして頂きお世話になりました．本研究を論文にまとめるにあたっても御協力をいただきました．感謝致します．

東芝における先輩，また上司として御指導を頂いて来た坂井邦夫氏，麻田治男氏に感謝します．お二人の御指導のおかげでこのような研究成果をあげることができました．また東芝における文字認識研究開発を進めて来た先輩，同僚，後輩の諸氏に感謝します．様々な面でお世話になりました．

東芝で一緒に研究開発進めて来たその他の分野の上司，先輩，同僚，部下，後輩の方々，そして東芝以外の多くの議論やアドバイスを頂いてきた方々にも感謝します．本研究を進める上ではいろいろな御協力を頂き，またお世話にもなりました．

筆者が現在所属する東芝ソリューションの幹部の方々には本研究を進めるにあたり様々な援助を頂いてきました．有難うございました．

最後に，本研究を進めていくにあたり様々な面で著者を励ましてくれた妻の文枝に感謝して謝辞の締めくくりとします．

参考文献

[1] S. Amari, “A theory of adaptive pattern classifiers”, *IEEE Trans. EC*, vol.16, no.3, pp.299-307 （1967）

[2] T. Kohonen, “Learning vector quantization”, *Helsinki University of Technology, Laboratory of Computer and Information Science, Report*, TKK-F-A601 （1986）

[3] S. Katagiri, C.-H. Lee, and B.-H. Juang, “A generalized probabilistic descent method”, *ASJ, Fall Conf.*, 2-p-6, pp.141-142, Nagoya, Japan （1990）

[4] A. Biem, and S. Katagiri, “Feature extraction based on minimum classification error / generalized probabilistic descent method”, *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, vol.2, pp.275-278 （1993）

[5] A. Biem, and S. Katagiri, “Filter bank design based on discriminative feature extraction”, *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Vol.1, pp.485-488 （1994）

[6] M. Bacchinai, and K. Aikawa, “Optimization of time-frequency masking filters using the minimum classification error criterion”, *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Vol.2, pp.197-200 （1994）

[7] K.K. Paliwal, M. Bacchiani, and Y. Sagisaka, “Minimum classification error training algorithm for feature extractor and pattern classifier in speech recognition”, *EUROSPEECH'95*, vol.1, pp.541-544 （1995）

[8] 河村聡典, 新田恒雄, “最小分類誤り学習による特徴選択型文字認識”, 信学論（D-II）, Vol.J81-D-II, No.12, pp.2749-2756 （1998）

[9] A. Sato, “Discriminative dimensionality reduction based on generalized LVQ”, *ICANN2001*, LNCS 2130, pp.65-72 （2001）

[10] C.-L. Liu, R. Mine, and M. Koga, “Building compact classifier for large character set recognition using discriminative feature extraction”, *Proc. ICDAR'05*, pp.846-850 （2005）

[11] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition”, *Proc. of the IEEE*, vol.86, no.11, pp.2278-2324 （1998）

[12] K. Fukushima, “Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position”, *Biol. Cybernet.*, vol.36, pp.193-202 （1980）

[13] J. Lampinen, and E. Oja, “Distortion tolerant pattern recognition based on self-organizing feature extraction”, *IEEE Trans., Neural Networks*, vol.6, no.3, pp.539-547 （1995）

[14] K. Ikuta, H. Tanaka, K. Tanaka, and K. Kyuma, “Learning algorithm by reinforcement signals for the automatic recognition system”, *Porc. SMC2004*, pp.4844-4848 （2004）

[15] G. Heidemann, and H. Ritter, “A neural 3-D object recognition architecture using optimized Gabor filters”, *Proc. ICPR'96*, D, pp.70-74 （1996）

[16] Y. Hamamoto, S. Uchimura, M. Watanabe, T. Yasuda, Y. Mitani, and S. Tomita, “A Gabor filter-based method for recognizing handwritten numerals”, *Pattern Recognition*, vol.31, no.4, 395-400 （1998）

[17] T. Joachims, “Making large-scale SVM learning practical”, *Advances in Kernel Methods - Support Vector Learning*, ed. B. Scholkopf, C.J.C. Burges, and A.J. Smola, pp. 185-208, MIT Press, Cambridge, MA （1999）

[18] T. Iijima, H. Genchi, and K. Mori, “A theory of character recognition by pattern matching method”, *Proc. 1st. IJCPR*, pp.50-56 （1973）

[19] 廣田薫, “画像パターン認識”, マグロウヒル （1984）

[20] 佐藤敦, 山田敬嗣, “正準化のための学習方法と低品質印刷文字認識”, 信学総全大, D-12-85, p.292 （1997）

[21] 若林哲史, 鶴岡信治, 木村文隆, 三宅康二, “手書き文字認識における特徴量の次元数と変数変換に関する考察”, 信学論（D-II）, Vol.J76-D-II, No.12, pp.2495-2503 （1993）

[22] D. E. Rumelhart, J. L. McClelland, and the PDP Research Group, “Parallel distributed processing”, pp.329-330, MIT Press, Cambridge, MA （1986）

[23] 山本和彦, “凹凸構造抽出と認識辞書の自動統合による手書きひらがな文字認識”, 信学論（D）, Vol.J65-D, No.6, pp.774-781 （1982）

[24] 飯島泰蔵, “混合類似度による識別理論”, 信学技報, PRL74-24, pp.45-54 （1974）

[25] S. Watanabe, “Karhunen-Loeve expansion and factor analysis”, *Trans. 4th Prague Conf. on Information Theory, Statistical Decision Functions, Random Processes*, Prague: Publishing House of the Czechoslovak Academy of Sciences, pp.635-660 （1967）

[26] 葛城純夫, “超楕円形識別関数による文字の識別”, 昭 45 信学全大 125 （1970）

[27] 葛城純夫, “文字パターン分布の要因分析と識別への応用”, 昭 45 電気四連大 2816 （1970）

[28] 栗田昌徳, 鶴岡信治, 横井茂樹, 三宅康二, “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, pp.105-112 （1982）

[29] F. Kimura, K. Takashina, S. Tsuruoka, and Y. Miyake, “Modified quadratic discriminant functions and the application to Chinese character recognition”, *IEEE Trans. PAMI*, Vol.9, No.1, pp.149-153 （1987）

[30] F. Kimura, M. Shridhar, and Y. Miyake, “Relationship among quadratic discriminant functions for pattern recognition”, *Proc. 4th IWFHR*, pp.418-422 （1994）

[31] 若林哲史, 鶴岡信治, 木村文隆, 三宅康二, “手書き数字認識における特徴選択に関する考察”, 信学論（D-II）, Vol.J78-D-II, No.11, pp.1627-1638 （1995）

[32] 池田正幸, 田中英彦, 元岡達, “手書き文字認識における投影距離法”, 情処論, Vol.24, No.1, pp.106-112（1983）

[33] K. V. Mardia, “Statistics of directional data”, Academic Press, New York, pp.233-236 （1972）

[34] 飯島泰蔵, “パターン認識”, コロナ社 （1973）

[35] 飯島泰蔵, “視覚パターンの特徴抽出に関する基礎理論”, 信学研究会, オートマトンと自動制御研究専門委員会資料, 1959 年 12 月 10 日 （1959）

発表論文リスト

論文誌

[36] 黒沢由明, 飯島泰蔵, “Harr 変換に基づく新しい濃淡画像の２値表現法”, 情処論, Vol.20, No.3, pp.218-224（1979）

[37] 黒沢由明, 飯島泰蔵, “濃淡画像の２値表示の方式”, 画像電子学会誌, Vol.10, No.2, pp.68-74 （1981）

[38] 黒沢由明, “平均濃度近似による濃淡画像の２値化表示方式”, 情処論, Vol.26, No.1, pp.153-160 （1985）

[39] 黒沢由明, “球面ガウス分布から導出される部分空間法”, 信学論（D-II）, J81-D-II, No.6, pp.1205-1212（1998）

国際会議

[40] Y. Kurosawa, H. Asada, “Attributed string matching with statistical constraints for character recognition”, *Proc. 8thICPR*, pp.1063-1067 （1986）

[41] Y. Kurosawa, “Similar pattern discrimination by filter mask learning with probabilistic descent”, *Proc. SCIA2007*, pp.958-967 （2007）

[42] Y. Kurosawa, “Incremental learning for feature extraction filter mask used in similar pattern classification”, *Proc. IJCNN2008*, pp.498-505 （2008）

[43] Y. Kurosawa, “Multiple filter mask learning of feature extraction in character recognition”, *Proc. ICFHR2008*, pp.106-111 （2008）

講演・チュートリアル

[44] 黒沢由明, “部分空間法入門”, 部分空間法研究会 Subspace2006（MIRU2006 サテライト研究会）, pp.136-143（2006）

[45] Y. Kurosawa, “The engineer’s guide to the subspace method - English version -”, *ACCV2007 Workshop Subspace2007*, pp.1-8 （2007）

研究会・全国大会

[46] 黒沢由明, ” 部分空間法とベイズ識別の関係から導かれる新しいパターン認識方式”, 信学技報, PRMU96-104, pp.1-8 （1996）

[47] 黒沢由明, ” 超球面上のガウス分布型パターン分布モデルに基づくパターン認識方式”, 平 9 信学総全大, D-12-100 （1997）

[48] 黒沢由明, ” 文字認識のための特徴抽出フィルタの学習方式”, MIRU2006, IS2-48, pp.940-945 （2006）

[49] 黒沢由明, ” 類似パターン識別における特徴抽出のフィルタ・マスク学習”, 信学技報, PRMU2006-267, pp.73-78 （2007）

[50] 黒沢由明, ” 類似文字識別のための単純類似度を用いた特徴抽出フィルタ･マスク学習”, 信学技報, PRMU2007-235, pp.121-126 （2008）