

修士学位論文

文字線画の伸縮変形モデルによる
文字認識に関する研究

東北大学大学院工学研究科 電気・通信工学専攻
加藤 毅

目次

第1章	序論	1
1.1	研究の背景	1
1.2	研究の目的	3
1.3	本論文の構成	3
第2章	複数ストローク対応型伸縮変形モデル	5
2.1	はじめに	5
2.2	文字骨格の変形	6
2.3	文字画像の生成	9
2.4	フィッティング	10
2.5	認識システム	15
2.5.1	辞書構成法	15
2.5.2	認識システム	15
2.6	評価実験	17
2.6.1	ETL8Bによる評価実験	17
2.6.2	ETL9Bによる評価実験	17
2.7	まとめ	20
第3章	多重解像度法に基づくフィッティング手法	21
3.1	グレースケール文字画像の生成	22
3.2	多重解像度法の適用	22
3.3	認識システム	25
3.4	評価実験	25
3.4.1	ETL8Bによる評価実験	25
3.4.2	ETL9Bによる評価実験	26
3.5	まとめ	28
第4章	形状修正アルゴリズム	35
4.1	形状修正アルゴリズム	35
4.2	認識システム	37
4.3	文字の識別	37
4.4	評価実験	38

4.5	まとめ	38
第5章	結論	46
5.1	本論文の成果	46
5.2	今後の課題	47
	参考文献	49

目次

1.1	ボトムアップ型システム	2
1.2	Revow らの文字モデル	2
2.1	モデルと未知パターンとの比較	5
2.2	モデル「す」	7
2.3	モデル「か」のインク発生器	9
2.4	モデル「と」のインク発生器	9
2.5	文字画像生成機構	11
2.6	モデル「あ」のインク発生確率	12
2.7	モデル「す」のインク発生確率	13
2.8	フィッティング手法の概要	14
2.9	認識システムの概要	16
2.10	ETL8B による認識率	18
2.11	ETL9B による認識率	18
2.12	正読例	19
2.13	誤読例 その1	19
2.14	誤読例 その2	19
2.15	誤読例 その3	20
3.1	グレースケール画像生成機構	23
3.2	多重解像度法によるフィッティングアルゴリズム	29
3.3	認識システムの概要	30
3.4	ETL8B による認識率	31
3.5	ETL9B による認識率	31
3.6	フィッティング例 その1	32
3.7	フィッティング例 その2	32
3.8	多重解像度法による改善例 その1	32
3.9	多重解像度法による改善例 その2	33
3.10	誤認識例 その1	33
3.11	誤認識例 その2	33
3.12	うまくフィットされていない例 その1	34
3.13	うまくフィットされていない例 その2	34

3.14	うまくフィットされていない例 その3	34
4.1	on/off-segment の特定	36
4.2	on-segment の組み合わせ集合	39
4.3	冗長なストロークの短縮	40
4.4	on-segment 間の再結合	40
4.5	最小二乗法による $X^{(oi)}$ の計算	41
4.6	形状修正アルゴリズムを伴ったフィッティング	42
4.7	認識システムの概要	43
4.8	ETL9B による認識率	44
4.9	形状修正アルゴリズムの効果 その1	44
4.10	形状修正アルゴリズムの効果 その2	45
4.11	形状修正アルゴリズムの効果 その3	45
4.12	誤認識された文字の例	45

第1章

序論

1.1 研究の背景

パターン認識は人間の能力を機械で実現するための要求から出発し、これまでに多くの研究が行われてきた。特に文字認識は最も歴史が古く、かつ実用化が進んだ分野である。文字は身近な存在である、記録性に優れ、再現性がよく、人間の直観を生かしやすく、一文字に一つの概念が対応するなど、パターン認識研究の素材としては格好の性質を備えている。また文字を読む技術は、コンピュータへのデータ入力をキーボードを経ずに直接行う省力装置の開発に直結することから、強い社会的ニーズに支えられてこれまで発展してきたともいえる。

手書き文字認識において、重ね合わせ法を基本とした多くの認識手法が提案されてきた。特に、方向性特徴を用いた手法は比較的丁寧に書かれた文字に対して高い認識率を示している [1] (図 1.1 参照)。これらの手法は、特徴抽出の前に字種非依存な正規化 [2], [3] を行って手書き文字の変形を緩和しようとしているが、このようなボトムアップ的な前処理では完全に変形を除去することは不可能で、大きな変形を吸収しきれないと考えられる。

より柔軟に変形に対応するため、前もってあらゆる変形を考慮してテンプレート側を変形に対応させる方法 [4] やこのとき、回転、平行移動などの変形パターンは無限に存在するため、計算機上で全ての変形を網羅するのは困難とし、離散的な変形パターンに限ってテンプレートを変形させている。Jain ら [5] は、文字のエッジ間のマッチングを行い、直交関数の重み付き線形和で表される変位関数でテンプレート画像の変形を表している。その係数の大きさを確率で表し、その尤度で変形を評価している。水上ら [6] は画像の濃度差で類似度を測り、画像の変形を変位関数の 1 次導関数の 2 乗和で評価している。内田ら [7] は、単調連続 2 次元ワープ法を文字画像に適用して変形させている。柳田ら [8] は文字画像に弾性膜をはり、質点の集合をポテンシャル場に落とすように力学的に運動させることで文字の整形を行なうモデルを提案している。しかし、これらの手法は切り出しミスなどに

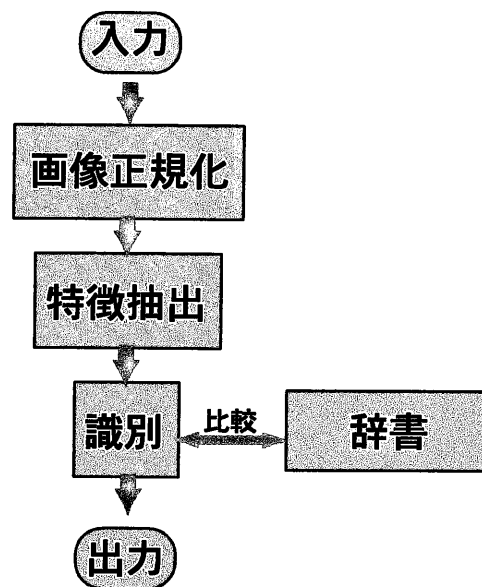


図 1.1: ボトムアップ型システム

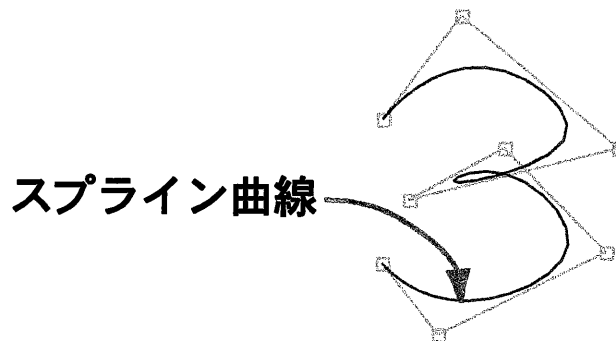


図 1.2: Revow らの文字モデル

より発生するノイズを取り込んでしまう可能性がある。人間のように入力画像からノイズと文字を構成するストロークを分離して認識するためには、基本的な構成要素であるストロークから文字をモデル化する必要があると考えられる。若原 [9] は文字を表すストロークを大量の点で表し、その画像に局所アフィン変換を繰り返して変形させている。Lamら [10] はストロークを直線近似し、山本 [11] は文字を多角形近似することで弛緩的な整合を行っている。

文字には曲線ストロークを持つものが多く存在するが、上記の手法は曲線に対する積極的な対処はされていない。一方、Revowら [12] の提案する生成モデルは文字を曲線の表現に適したスプライン曲線で表している(図 1.2 参照)。このモデルは入力画像に合わせて確率的に変形させる(フィッティング)ことができ、2値の文字画像をスプライン曲線上に配列された多数のインク発生器によって確率的に生成する。これをサイズの枠組み [13] に則って事前確率と尤度の正規化定数を自動的に算出する手法 [14] も提案されている。生成モデルでは、入力文字画像とのフィッティングは、モデルが文字画像を生成する確率を最

大にするパラメータの探索によって行われる。この探索には EM アルゴリズム [15] が用いられる。これは数字のような構造が簡単な文字では局所最適解に陥ることは少なく、多くの場合に大域的最適解に近い解が得られる。しかし、数字よりも多くのストロークを持つ平仮名のような文字はモデルのパラメータ空間がより複雑化するため、EM アルゴリズムをそのまま適用すると局所最適解に陥りやすい傾向がある。これを解決するために、様々な初期値を与えて、尤度最大のものを採用する方法がとられることもあるが、これはその分計算量を大幅に増やすことになるため、一つの初期値で良好な解を得るアルゴリズムが求められる。

1.2 研究の目的

本研究では、手書き文字を対象とし、多様な変形にも対応できる認識手法の開発を目指す。

生成モデルでは一本のスプライン曲線で表現されているので、認識対象となる文字種は数字のようなストロークを一本しか持たない文字に限られている。故に平仮名のような複数のストロークを認識する場合、何らかの改良が必要となる。また、生成モデルでは数字のような簡単な構造の文字を対象としており、平仮名のような複雑な構造を持つ文字にフィッティングさせるには工夫が必要である。以上のような考えに則って、複数のストロークを持つ複雑な文字にも対応できるモデルを考案し、認識性能の向上を目指す。

1.3 本論文の構成

本論文の構成は以下のとおりである。

第1章 序論

研究の背景や目的、論文の構成を述べる。

第2章 複数ストローク対応型伸縮変形モデル

変形の大きい文字に柔軟に対応できる生成モデルをもとに、複数のストロークを持つ文字を表現する新たな伸縮変形モデルを提案する。さらに、ストローク群のなす文字構造の変動を十分に表すため、ストロークどうしの関係などを利用した新たな評価関数を提案する。

第3章 多重解像度法に基づくフィッティング手法

EM アルゴリズムを適用して入力されたグレースケール文字画像とのフィッティングを行う際に多重解像度法を適用し、良好な解を安定して得る手法を提案する。

第4章 形状修正アルゴリズム

フィッティング性能を向上させる形状修正アルゴリズムを提案する。

第5章 結論

本研究の成果, 今後の課題について述べる.

第2章

複数ストローク対応型伸縮変形モデル

2.1 はじめに

本章では、数字のような一筆書きで表現できる文字の生成モデル [12] に基づき、平仮名のような複数のストロークを持つ文字画像を生成する確率モデルとして伸縮変形モデルを提案し、認識に用いるための未知パターンとのフィッティング手法、評価尺度である確率を説明する。本手法では 図 2.1 に示すようにモデルを変形させることによって未知パターンに近づけ、そのうえでマッチングを行っている。提案するモデルは変形を事前確率の対数で測り、モデルと画像との類似度も画像の生成する確率の対数尤度で求める。モデルからの文字画像の生成されやすさはこの2つの量の和で表され、その事後確率最大のモデルを見つけて認識結果とする。

2.2 では、複数ストロークの文字を表すモデル表現について述べ、このモデルが変形したときの変形度合いを定量化するため、モデルの変形する確率について述べる。2.3 では、

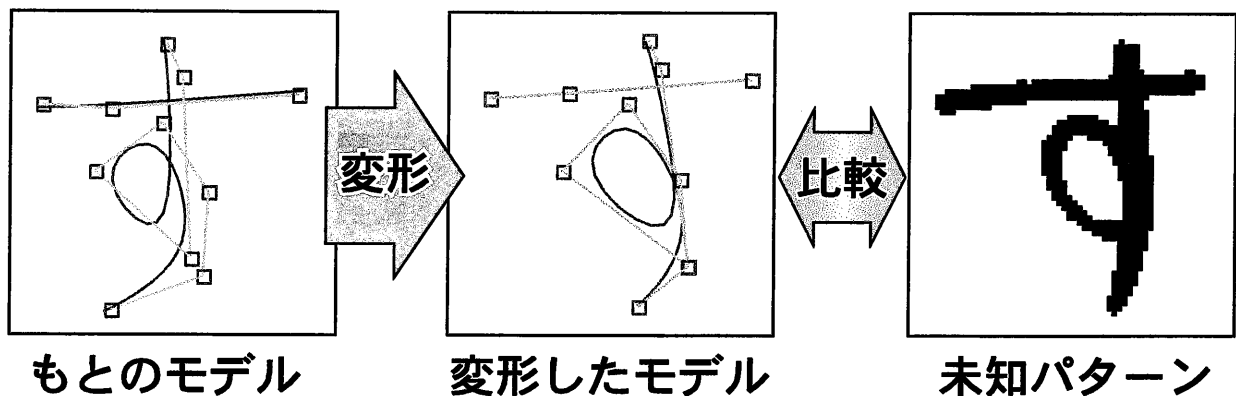


図 2.1: モデルと未知パターンとの比較

変形したモデルと文字画像との類似度を計算するため、変形したモデルが文字画像を生成する尤度について述べる。2.4では2.2及び2.3で述べた尤度を使って、従来より知られているEMアルゴリズムを用いてモデルを入力文字画像にフィッティングさせる方法について述べる。2.5では本手法における辞書の構成法、また、提案モデルを用いた認識システム、2.5.2では文字の識別手法について述べる。2.6では電子技術総合研究所提供の手書き文字データベースETL8B[16]とETL9B[17]の平仮名を使って評価実験を行うことにより、提案モデルの性能を調査し、文字の識別手法の有効性を確認する。

2.2 文字骨格の変形

文字のストロークを表すため、数個の制御点により定義されるスプライン曲線を用いる。標準的の文字パターンの各ストロークの制御点をホーム制御点と呼ぶ。文字は一般に複数のストロークを持ち、ストロークの本数分のスプライン曲線で文字をモデル化する。 N_{stroke} 本のスプライン曲線に対して i 本目の曲線を表すホーム制御点の集合を列ベクトル

$$\mathbf{H}^{(i)} = \begin{bmatrix} h_{x_1}^{(i)} \\ h_{y_1}^{(i)} \\ h_{x_2}^{(i)} \\ h_{y_2}^{(i)} \\ \vdots \\ h_{x_{n_i}}^{(i)} \\ h_{y_{n_i}}^{(i)} \end{bmatrix}$$

で表す。ここで、 $(h_{x_j}^{(i)}, h_{y_j}^{(i)})$ が第 j 制御点の座標となる。全てのストロークのホーム制御点集合を

$$\mathbf{H} = \{\mathbf{H}^{(1)}, \dots, \mathbf{H}^{(N_{\text{stroke}})}\}$$

と表す。このホーム制御点はその字種の平均的な形状を表すように、実際の文字画像から推定される。例えば、モデル「す」を表すホーム制御点は図2.2(a)のようになる。第 i ストロークの第 j 制御点を $i-j$ と表している。

制御点を動かすとスプライン曲線は変形し、変形した文字骨格が得られる。実際の文字の骨格はこの変形したスプライン曲線に対応し、同一字種の様々な文字パターンの存在は文字骨格が確率的に変形したものとみなすことができる。変形したスプライン曲線を表す制御点の集合を

$$\mathbf{X} = \{\mathbf{X}^{(1)}, \dots, \mathbf{X}^{(N_{\text{stroke}})}\}$$

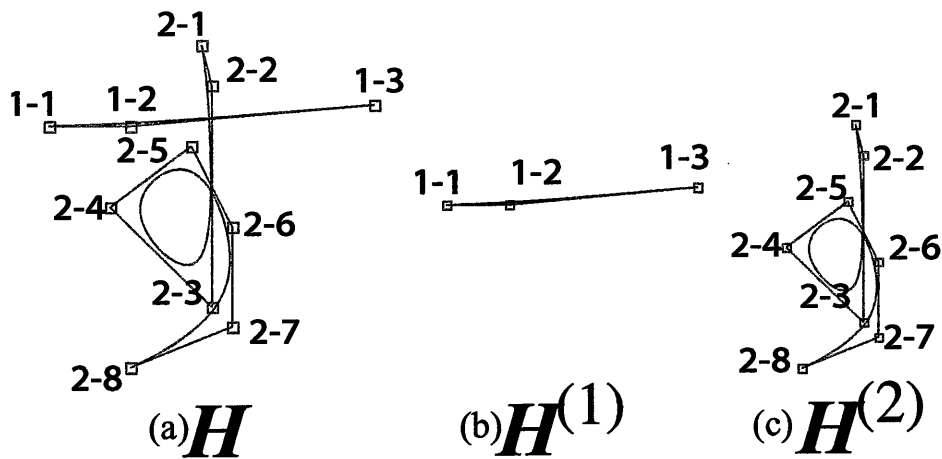


図 2.2: モデル「す」.

と表す. 第 i ストロークを表す制御点 $\mathbf{X}^{(i)}$ は

$$\mathbf{X}^{(i)} = \begin{bmatrix} x_{x_1}^{(i)} \\ x_{y_1}^{(i)} \\ x_{x_2}^{(i)} \\ x_{y_2}^{(i)} \\ \vdots \\ x_{x_{n^i}}^{(i)} \\ x_{n^i}^{(i)} \end{bmatrix}$$

と表す. ここで, $(x_{x_j}^{(i)}, x_{y_j}^{(i)})$ が第 j 制御点の座標となる.

この \mathbf{X} は \mathbf{H} に対して, 回転, 平行移動, 拡大縮小といった大局的な変換がなされた制御点から, さらに確率的な局所的移動により得られると考える. 大局的な変換は \mathbf{H} に対する自由度 6 のアフィン変換 A で表される. 変形された文字骨格は制御点 \mathbf{X} とアフィン変換 A で特定される. モデルのパラメータ $\{\mathbf{X}, A\}$ の確率は次式で表される.

$$P(\mathbf{X}, A | \mathbf{H}) = P(\mathbf{X} | A, \mathbf{H}) P(A | \mathbf{H}) \tag{2.1}$$

すなわち, 確率 $P(A | \mathbf{H})$ に従って \mathbf{H} を A で変換し, 変換された位置から確率 $P(\mathbf{X} | A, \mathbf{H})$ に従って制御点が移動して \mathbf{X} になると仮定している.

認識時に与えられた文字が, 標準文字骨格が確率的に変形したものであるとして, 与えられた文字 \mathbf{X} についてのモデルのパラメータ $\{\mathbf{X}, A\}$ の尤もらしさは, その確率の負の対数で評価する. これを変形エネルギー E_{def} と呼ぶ. E_{def} は文字骨格が変形した量をはかるものであるとみなせる.

$$E_{\text{def}} = -\log(P(\mathbf{X}, A | \mathbf{H})) \tag{2.2}$$

$$= -\log(P(A | \mathbf{H})) - \log(P(\mathbf{X} | A, \mathbf{H})) \tag{2.3}$$

次に $P(\mathbf{X}, A | \mathbf{H})$ の具体的な形を示す。まず、アフィン変換 A による変形の程度をはかるために、アフィン変換に関する確率 $P(A | \mathbf{H})$ を変換後のスプライン曲線を囲む矩形の縦横比 a に関する確率として定義する。すなわち、

$$P(A | \mathbf{H}) = P(a | \mathbf{H}) \quad (2.4)$$

とする。また、文字骨格の局所的な変形の確率を次式で定める。

$$P(\mathbf{X} | A, \mathbf{H}) = \prod_{i=1}^{N_{\text{stroke}}-1} P(r_i | A, \mathbf{H}) \cdot \prod_{i=1}^{N_{\text{stroke}}} P(\mathbf{X}^{(i)} | A, \mathbf{H}^{(i)}) \quad (2.5)$$

ただし、 r_i は第 i ストロークの最後の制御点と第 $(i+1)$ ストロークの最初の制御点との距離であり、ストローク相互の相対的位置を規定するものと考えた。

$P(a | \mathbf{H})$, $P(r_i | A, \mathbf{H})$ 及び $P(\mathbf{X}^{(i)} | A, \mathbf{H}^{(i)})$ は正規分布と仮定し、各分布のパラメータは学習データを用いて推定する。縦横比 a の平均と分散をそれぞれ μ_a , σ_a^2 とし、 r_i の平均と分散をそれぞれ μ_{r_i} , $\sigma_{r_i}^2$ とする。ここで

$$E_a = -\log(P(A | \mathbf{H})) \equiv \frac{(a - \mu_a)^2}{2\sigma_a^2} \quad (2.6)$$

とし、 $-\log(P(\mathbf{X} | A, \mathbf{H}))$ は $\{r_i\}$ に関する尤度 E_r と \mathbf{X} に関する尤度 E_{cp} に分ける。すなわち

$$-\log(P(\mathbf{X} | A, \mathbf{H})) = E_r + E_{\text{cp}} \quad (2.7)$$

ただし

$$E_r \equiv \sum_{i=1}^{N_{\text{stroke}}-1} \frac{(r_i - \mu_{r_i})^2}{2\sigma_{r_i}^2} \quad (2.8)$$

$$E_{\text{cp}} \equiv \sum_{i=1}^{N_{\text{stroke}}} \frac{1}{2} (A^{-1} \mathbf{X}^{(i)} - \mathbf{H}^{(i)})^t (\Sigma^{(i)})^{-1} \cdot (A^{-1} \mathbf{X}^{(i)} - \mathbf{H}^{(i)}) \quad (2.9)$$

ただし、 A^{-1} はアフィン変換 A の逆写像を表し、 $\Sigma^{(i)}$ は $A^{-1} \mathbf{X}^{(i)}$ の共分散行列を表す。本手法では、正規分布の対数をとったときの定数項を無視した。 E_a はその字種の平均的な縦横比と入力文字との縦横比が異なるほど、大きな値となる。 E_r はストローク端点間の距離が平均的な文字と異なるほど、大きな値となる。 E_{cp} はモデルの局所的な変形の程度を表す。

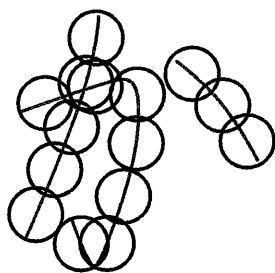


図 2.3: モデル「か」のインク発生器.

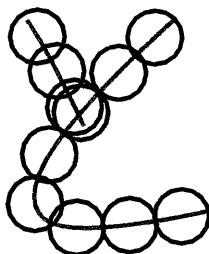


図 2.4: モデル「と」のインク発生器.

2.3 文字画像の生成

前節では、文字の骨格の変形について述べたが、未知パターンを認識するには、変形した文字骨格と入力文字画像との類似度を評価する必要がある。そこで、入力画像が文字骨格から生成されるものとして、その生成確率の対数尤度で評価することを考える。

画像生成は画面上にインクを生成する多数のインク発生器を通してなされるものとする。インク発生器はその位置の周辺に正規分布に従ってインクを発生させる。この分布の分散はパラメータ σ_b^2 として扱う。 σ_b をインク発生器のサイズともいう。インク発生器はそれぞれの文字骨格上にそのサイズのほぼ倍の長さの間隔を置いて均等に配置されるものとする。インク発生器は円形で、その半径がサイズに等しく、ストローク上に隙間無く並んでいるとみなせる。図 2.3, 図 2.4 にそれぞれ「か」「と」を表す文字骨格上にインク発生器を配置した例を示す。

画素 k が黒になる確率 $p(d_k = 1)$ は次のような混合分布で定める。¹

$$p(d_k = 1) = \frac{\pi_{\text{noise}}}{|\mathbf{Z}|} + \frac{\pi_{\text{ink}}}{|\mathbf{B}|} \sum_{b \in \mathbf{B}} N(z_k | s_b(\mathbf{X}, \sigma_b^2), \sigma_b^2) \quad (2.10)$$

ただし、 $N(\cdot)$ は正規分布を示す。 $\pi_{\text{noise}}, \pi_{\text{ink}}$ はそれぞれノイズとインク発生器との分岐確率であり、 $\pi_{\text{noise}} + \pi_{\text{ink}} = 1$ を満たす。 \mathbf{Z} は文字画像を表し、これをある閾値以上の濃度²をもつ画素の集合として定める。 $|\mathbf{Z}|$ は文字画像の画素数である。 \mathbf{B} はインク発生器の集合

¹[12] で用いられたものと等しい。

²閾値の設定は、計算量削減のためである。

第2章 複数ストローク対応型伸縮変形モデル

を表し, $|B|$ はその個数である. z_k は第 k 画素の座標, s_b は第 b インク発生器の座標を示す. 式 (2.10) は汚染モデル [18] の一種と見なすことができる. つまり, インク発生器以外から発生したノイズを一様分布とみなすことで, ノイズに対してロバストなパラメータ推定法を与えている.

この2値の文字画像を生成する機構を図示したのが図 2.5 である. 黒インクはまずノイズとして画像上に生成されるか, インク発生器から生成されるか, それぞれ確率 $\pi_{\text{noise}}, \pi_{\text{ink}}$ で分岐する. ノイズとして生成されることが選択された黒インクは画像上の任意の場所に落とされる. 残りの黒インクはインク発生器選択器によりどのインク発生器から落とされるか決められる. このとき, 全てのインク発生器は等確率で選ばれる. そしてインク発生器から分散 σ_b の正規分布に従って座標が決定され, その場所に黒インクが落とされる.

図 2.6 にモデル「あ」がそれぞれの画素にインクを発生させ黒くする確率を示す. 図 2.6(a) は分散を徐々に小さくしたときのインク発生器の配置形態を示し, 図 2.6(b) は画像中の各々の画素の濃度が 1(黒) になる確率を濃さで表している. これを見ると, 分散が小さくなるにつれ, モデルは精密化され, 生成される文字の形状に対応する位置を黒くする確率が高くなる様子が確認される. 図 2.7 はモデル「す」がそれぞれの画素にインクを発生させ黒くする確率を示す.

それぞれの画素のインクが独立に発生するものとする, 文字画像 Z が発生する確率は,

$$P(Z|X, A, H) = \prod_{k \in Z} p(d_k = 1) \quad (2.11)$$

で表される.

未知パターンに関する各文字であることの尤もらしさは, 変形した後のモデルからその未知パターンが生成される確率の対数である対数尤度で評価できる. その値をフィットエネルギー E_{fit} と呼ぶ.

$$E_{\text{fit}} = -\log P(Z|X, A, H) \quad (2.12)$$

これは変形後の文字骨格と未知パターンとの類似度を表しているといえる.

2.4 フィッティング

本節では, モデルを入力文字画像にあわせて変形するフィッティング手法について述べる. フィッティングは, 入力文字画像にモデルのストロークを対応させることに主眼をおき, 目的関数として $E_{\text{tot}} = E_{\text{cp}} + w_{\text{fit}} E_{\text{fit}}$ を考える. 重み係数 w_{fit} は文字の大きさ, 太さなどの影響を軽減するため $w_{\text{fit}} = w'_{\text{fit}} / \sum_k \lambda_k$ を用いた. ここで w'_{fit} は定数であり実験により決める. この目的関数を最小化するモデルのパラメータ $\{X, A\}$ を探索することが, 未知パターンへのフィッティングである.

まず, アフィン変換 A の初期値をモデルを囲む最小の矩形が入力文字画像 Z を囲む最小の矩形に合うように設定し, 制御点 X の初期値を AH とする. 目的関数を最小にするパラメータ $\{X, A\}$ の探索を ECM (Expectation Conditional Maximization) アルゴリズム

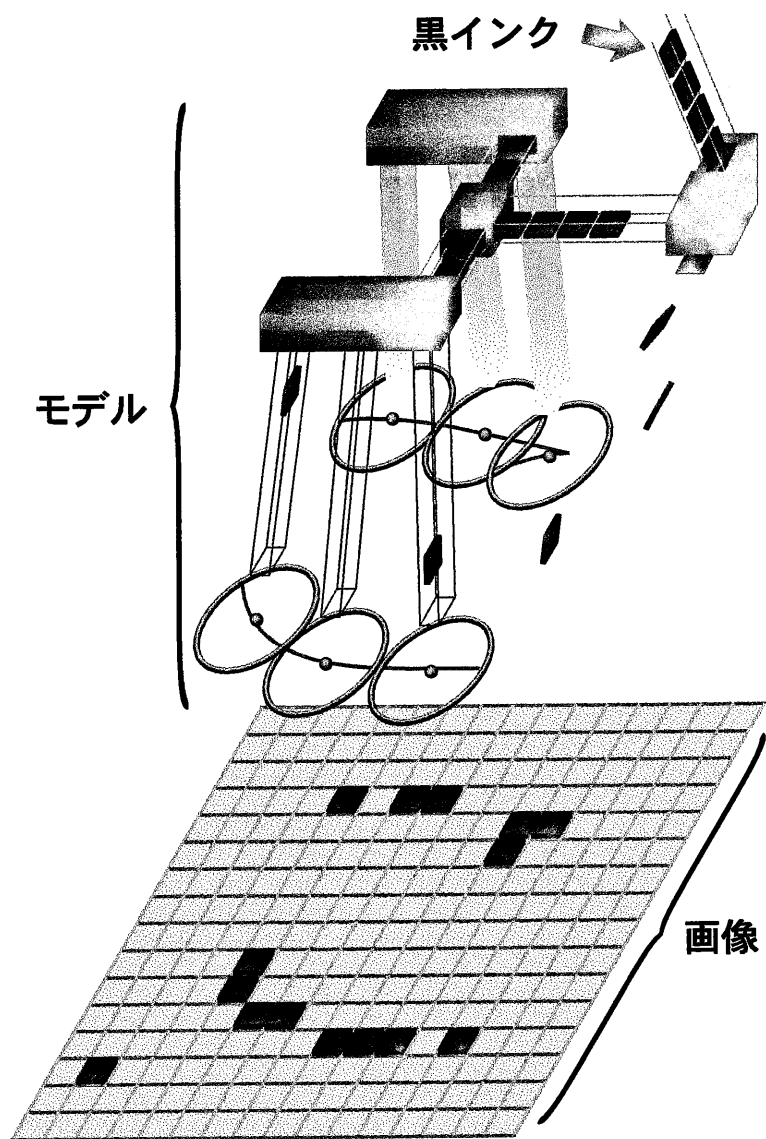
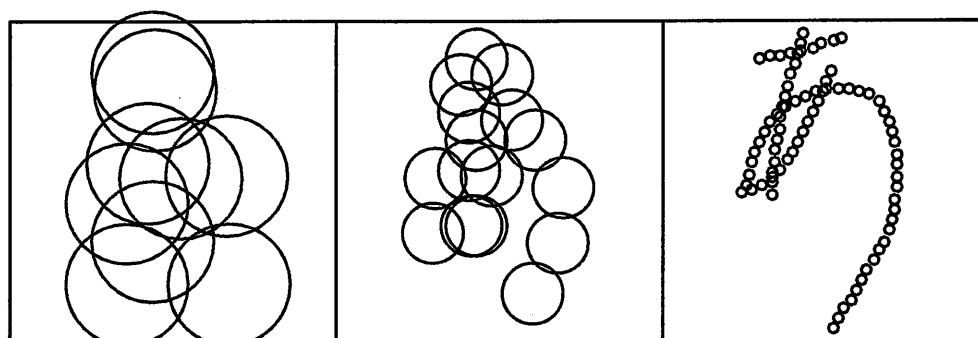
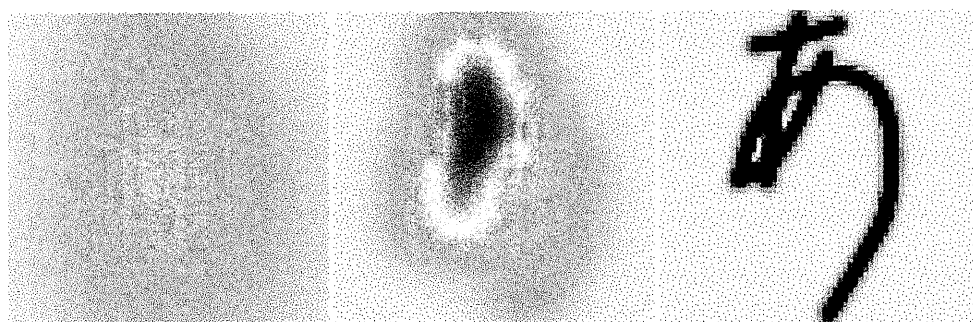


図 2.5: 文字画像生成機構

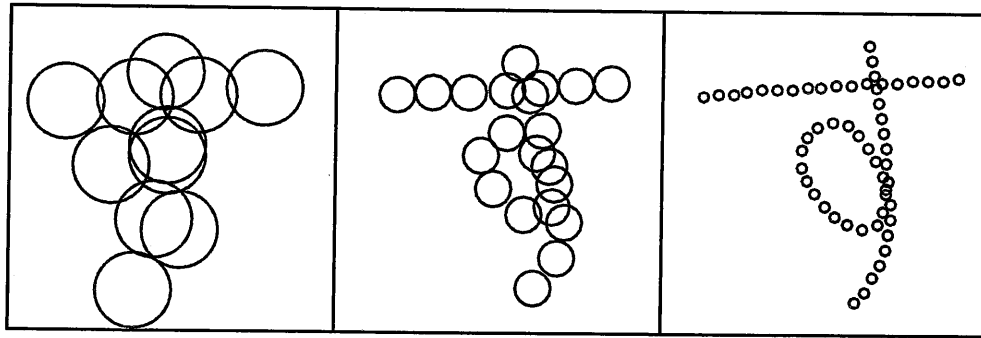


(a) Ink generators

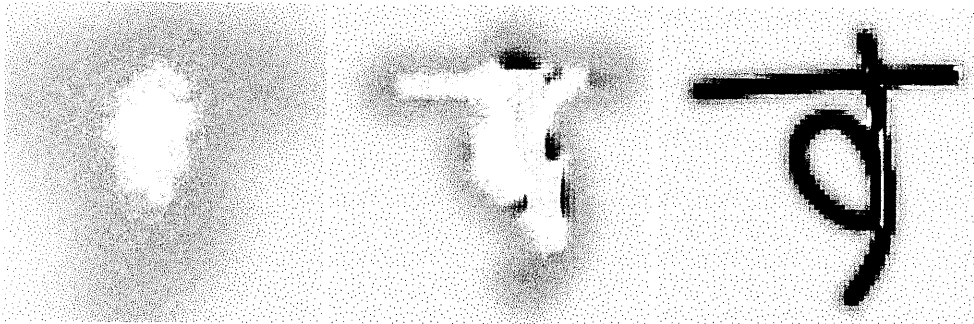


(b) Generated patterns

図 2.6: モデル「あ」のインク発生確率



(a) Ink generators



(b) Generated patterns

図 2.7: モデル「す」のインク発生確率

ム [12], [19] を使用して行なう。ECM アルゴリズムは EM アルゴリズム [15] の一種である。EM アルゴリズムは最尤推定の局所最適解を求める反復アルゴリズムであり、各イタレーションで得られるパラメータは目的関数を単調増加させることが知られている。フィッティング手法の概要を図 2.8 に示す。

具体的には、 A と X を初期化した後、入力文字画像に対して、以下の手続きによってフィッティングが行われる。

```

procedure FITTING(  $Z$  : image,
                    var { $X, A$ } : parameters )
begin
  repeat
    インク発生器をスプライン曲線上に配列;
     $\partial E_{\text{tot}} / \partial X = 0$  なる  $X$  に更新;
     $\partial E_{\text{tot}} / \partial A = 0$  なる  $A$  に更新;
    インク発生器のサイズ  $\sigma_b$  を最尤推定して更新;
  until  $E_{\text{tot}}$  収束.
end.

```

フィッティングのための EM アルゴリズムは、ここで画像 Z に正規化、特徴抽出といった前処理はされていないことを注意されたい。インク発生器の初期サイズとしてやや大きめ

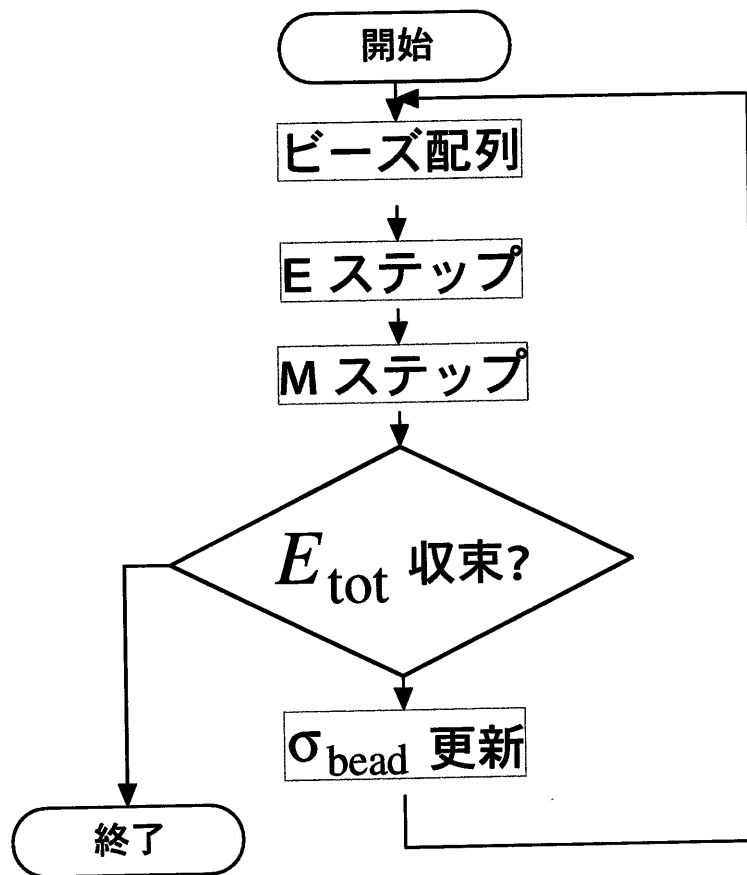


図 2.8: フィッティング手法の概要.

な値を与えておき, EM の各イタレーションごとにそのサイズを最尤推定する. 結果的に, インク発生器はイタレーションが進むにつれて, 徐々に小さくなる傾向がある. 目的関数 E_{tot} の収束の判定は目的関数の変化が閾値 θ_{tot} 以下になったかどうかで決める.

2.5 認識システム

本章では, 伸縮変形モデルを用いるための辞書構成法及び認識システムについて説明する.

2.5.1 辞書構成法

辞書は, 各字種毎にホーム制御点 H , その分散共分散行列 Σ , 縦横比の分布パラメータ $\{\mu_a, \sigma_a^2\}$, ストローク間距離の分布パラメータ $\{\mu_{r_1}, \sigma_{r_1}^2, \dots, \mu_{r_{N_{\text{stroke}}-1}}, \sigma_{r_{N_{\text{stroke}}-1}}^2\}$ からなる. まず辞書を構築するための制御点を各文字の典型的パターンから手動で作成した. この制御点をホーム制御点として, 学習用サンプル文字画像について, 2.4 で述べたフィッティング手法を適用する. 各学習サンプルに対して得られたモデルのパラメータ $\{X, A\}$ から, 制御点の座標の平均と分散共分散行列を求め, これをそれぞれ $H, \{\Sigma^{(i)}\}$ とする. また, 縦横比の平均 μ_a ・分散 σ_a^2 , ストローク間距離の平均 $\{\mu_{r_i}\}$ ・分散 $\{\sigma_{r_i}^2\}$ を求める. 学習用サンプルのフィッティングの際, 局所最適解に陥ったものを用いるのを防ぐため, 文字画像を上下左右に一定量平行移動したものと移動させない画像との5種類を用意し, それぞれに X の初期値を設定して, フィッティングを行い, E_{tot} が最小のものを採用した. これは, 学習用データが非線形な変形をしていることに対処するもので, 自動的に辞書を作成するために導入している.

認識時の計算時間と辞書記憶容量の点からは, 1 字種につき 1 個の制御点で認識を行なうシングルテンプレート方式をとり, モデルの柔軟性によって多様な変形を吸収するのが理想的な手法と言えるが, 平仮名の場合, 例えば「り」のような文字は 1 本のストロークで書く筆記者と 2 本で書く筆記者が存在するので, モデルの柔軟性だけでは対応しきれない. そこで, 「そ」, 「ゆ」, 「り」の 3 字種に限り, マルチテンプレートを採用した.

2.5.2 認識システム

図 2.9 に認識システムの流れを示す. この認識システムでは, 入力画像の画像正規化, 特徴抽出といったプロセスは不要である. 入力した画像は多重解像度法を用いて各字種のモデルとのフィッティングが行われる. 未知パターンの認識は, ここで得られた各字種のモデルのパラメータ $\{X, A\}$ から次の評価関数の値が最も小さい字種のモデルを選択することで行なわれる.

$$E_{\text{rec}} = w_{\text{def}} E_{\text{def}} + w_{\text{fit}} E_{\text{fit}} \quad (2.13)$$

この評価関数をトータルエネルギーと呼ぶ. $w_{\text{def}}, w_{\text{fit}}$ は重み係数で予備実験により決める.

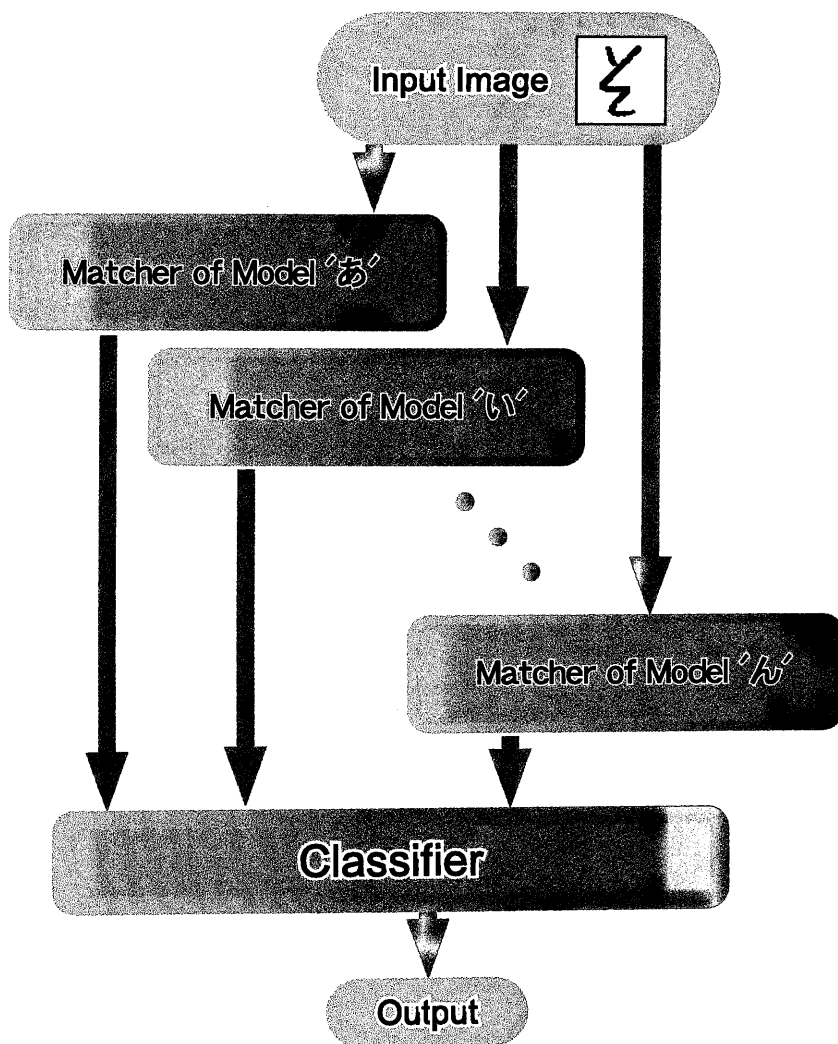


図 2.9: 認識システムの概要

2.6 評価実験

提案する認識システムの有効性を確認するため、電子技術総合研究所提供の手書き文字データベース ETL8B[16] と ETL9B[17] の平仮名を使って、認識実験を行なう。

2.6.1 ETL8B による評価実験

ETL8B は 71 字種の平仮名を含む手書き文字データベースで、1 字種につき 160 個のサンプルがある。このうち、濁点・半濁点を含む文字を除いた平仮名 46 字種に対して認識実験を行なった。学習用サンプルとテスト用サンプルを分けるため、偶数番号のサンプル 80 セットを学習用サンプルに、残りの奇数番号のサンプル 80 セットをテスト用サンプルとした実験と、反対に奇数番号を学習用に、偶数番号をテスト用に用いた実験との 2 通りを行ない、2 つの認識率の平均を本システムの認識率とした。

また、変形エネルギー (式(式(2.2))-式(式(2.9))) の各項の有効性を確認するため、手法 A,B それぞれについて以下の 4 通りのエネルギーで実験した。

- $E_{\text{def}} = E_{\text{def}}^{(\text{cp})} = E_{\text{cp}}$
- $E_{\text{def}} = E_{\text{def}}^{(\text{a+cp})} = E_{\text{a}} + E_{\text{cp}}$
- $E_{\text{def}} = E_{\text{def}}^{(\text{r+cp})} = E_{\text{r}} + E_{\text{cp}}$
- $E_{\text{def}} = E_{\text{def}}^{(\text{a+r+cp})} = E_{\text{a}} + E_{\text{r}} + E_{\text{cp}}$

図 2.10 に認識率の平均を手法別に示す。また、縦横比を考慮した $E_{\text{def}}^{(\text{a+cp})}$ の結果とストロークの端点間の間隔を考慮した $E_{\text{def}}^{(\text{r+cp})}$ の結果は $E_{\text{def}}^{(\text{cp})}$ を用いたものより、良好な結果が得られた。また、縦横比とストロークの端点間の間隔の両方を組み入れた $E_{\text{def}}^{(\text{a+r+cp})}$ はこれら全てを上回る認識性能を示した。

2.6.2 ETL9B による評価実験

手書き文字データベース ETL9B は、各字種毎に 200 セットのサンプルで構成されている。ETL8B による実験と同様、平仮名 46 字種を対象に実験を行った。サンプル 200 セットを 10 個の部分サンプル集合に分けた。これを Leave-one-out 法に従い、部分サンプル集合のうち 1 個だけをテスト用サンプルとし、残りの 9 個の部分サンプル集合を学習用サンプルとした。その全ての組合せ 10 通りについての認識率の平均を本システムの認識率として評価することにした。

4 種類の変形エネルギーを比較した。結果を図 2.11 に示す。また、縦横比もしくはストロークの端点間の距離の重み付きユークリッド距離を付加した変形エネルギー $E_{\text{def}}^{(\text{a+cp})}$, $E_{\text{def}}^{(\text{r+cp})}$ は、加えないもの $E_{\text{def}}^{(\text{cp})}$ より手法 A,B いずれも認識率の向上が見られた。両方加えた変形エネルギー $E_{\text{def}}^{(\text{a+r+cp})}$ はさらなる認識率向上が得られた。

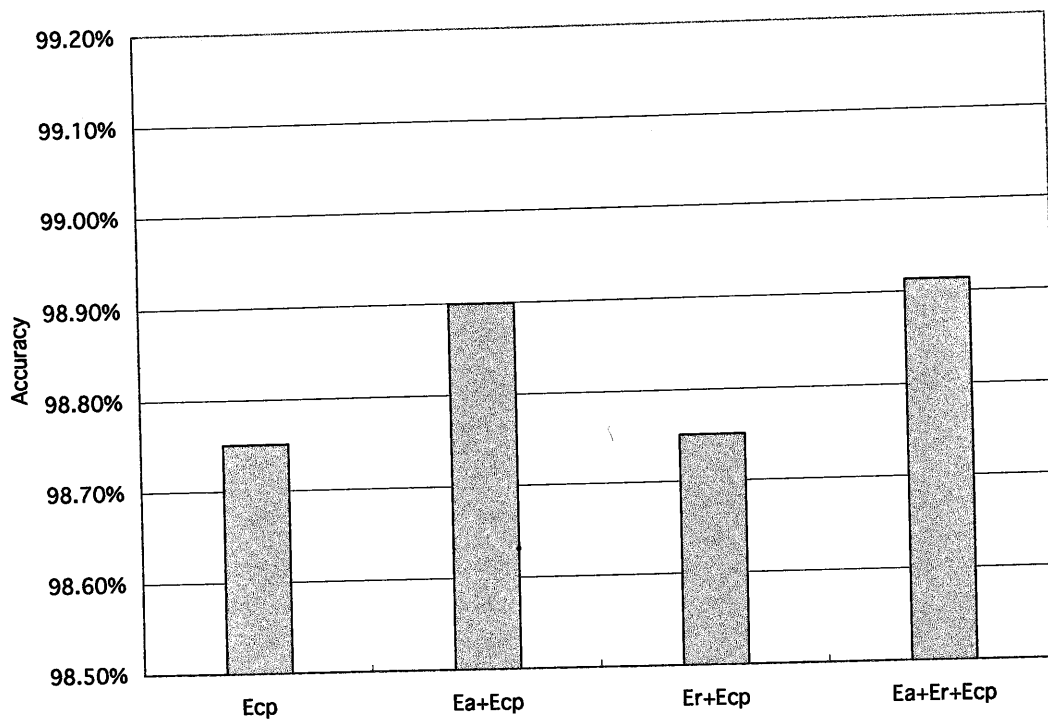


図 2.10: ETL8B による認識率

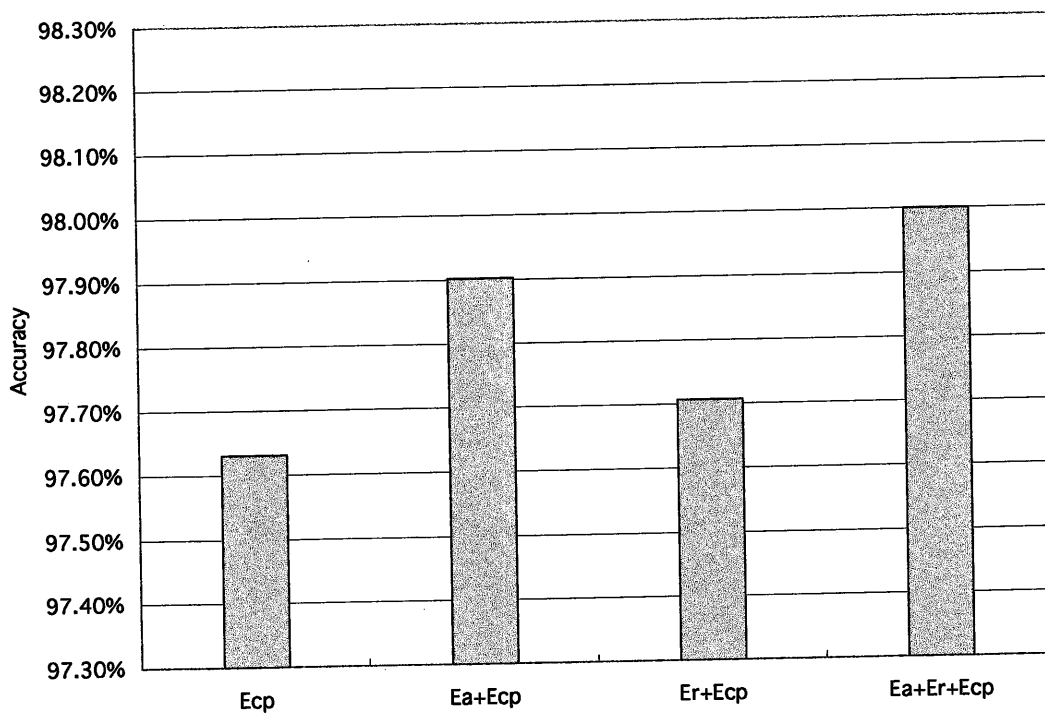


図 2.11: ETL9B による認識率

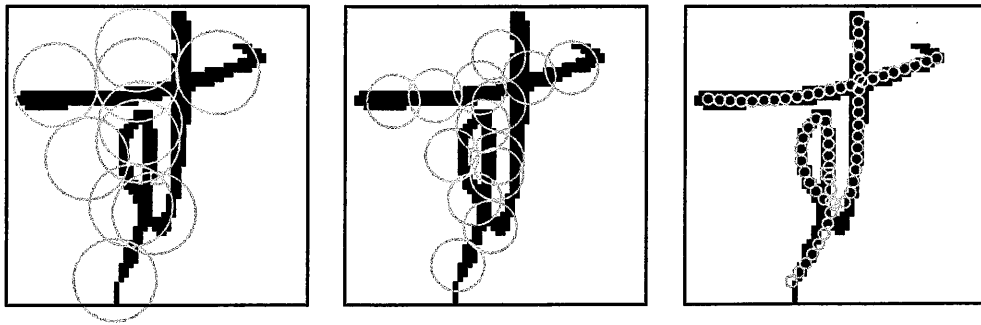


図 2.12: 正読例

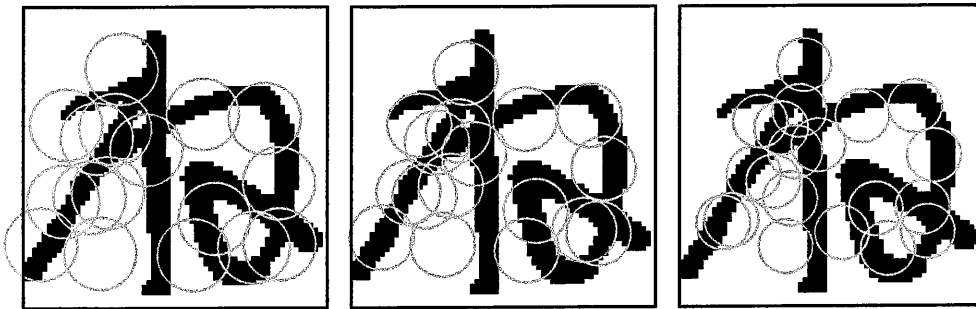


図 2.13: 誤読例 その1

図 2.12 に本手法を用いてフィッティングさせた例を示す。図は文字画像「す」にモデル「す」を使ってフィッティングさせた場合である。左から、初期状態、途中の状態、最終状態を示しているが、いずれもうまくフィットされている様子が確認される。

図 2.13 は文字画像「ね」にモデル「ね」をフィッティングさせた場合であるが、縦棒が縦棒にうまくフィットされず、インク発生器のサイズが大きいまま目的関数が収束している。この文字は誤認識された。図 2.14 は文字画像「そ」にモデル「そ」をフィッティングさせた場合であるが、モデルの第2画が正しくフィットされなかった。この文字は誤認識された。このように複数ストロークが似ている部分を含むような文字は数字のような簡単な構造の文字と比べ多くの局所最適解を持ってしまう傾向がある。EM アルゴリズムは初期値依存性が高いため、EM アルゴリズム単体では複雑な探索空間に対して、完全に局所最

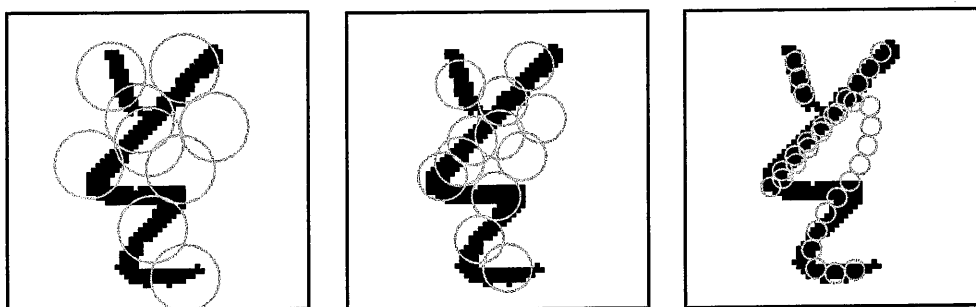


図 2.14: 誤読例 その2

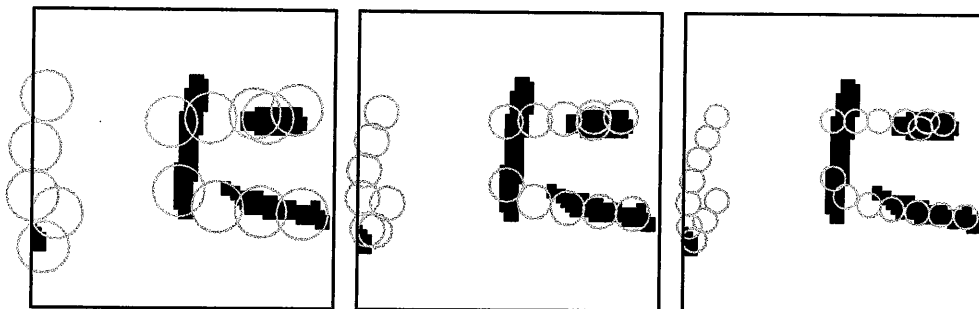


図 2.15: 誤読例 その3

適解を防ぐ方法はない。複雑な構造の文字は複雑な探索空間を持つため、しばしば局所最適解に陥ってしまい、誤認識した場合があった。用いたフィッティング手法では局所最適解をなるべく回避するため、インク発生器のサイズを徐々に小さくして粗密探索を実現していたが、それでもしばしば局所最適解に陥ってしまった。

図 2.15 は文字画像「に」にモデル「に」をフィッティングさせた場合である。この画像は切り出しミスによるノイズが左はしに存在している。本手法ではモデルが入力画素 Z の大きさに合うようにアフィン変換の初期値が定められるが、このようなノイズがある文字もこれを囲むように定められてしまう。「に」は左側の縦棒が分離しているため、このノイズの影響で第 1 ストロークがフィットできなかった。このように切り出しミスによるノイズの影響により誤認識した例もあった。切り出しミスなどによるノイズが存在すると、それも囲むように初期アフィン変換が設定され、ノイズが最後まで影響し、間違った文字にフィットされた。数字のような簡単な構造の画像ではノイズを無視してフィットされることもあるが、平仮名のような複雑な構造の文字ではノイズの影響により局所最適解に陥ることが多いことが分かった。

2.7 まとめ

本章では複数のストロークを持つ文字を少数のパラメータで表現できる変形に柔軟な複数ストローク対応型伸縮変形モデルを提案した。また、このモデルを使った有効な文字の識別手法を提案した。提案した複数ストローク対応型伸縮変形モデルの性能について調査し、また、識別用評価関数におけるモデルの変形の程度をはかる各項についての有効性を確認するため、電子技術総合研究所提供の手書き文字データベース ETL8B[16] と ETL9B[17] の平仮名を使って、認識実験を行った。その結果、識別用評価関数の各項の有効性が示された。また、モデルのフィッティング性能に関しては、複雑な文字に対してしばしばうまくフィットされないことがあることが分かった。これは(大域的最適解ではない)局所最適解に陥ったためと考えられる。平仮名は数字と比べ構造が複雑なものが多い。構造が複雑な文字は複雑な探索空間を持ち、そのため局所最適解に陥りやすいと考えられる。これが誤認識の大きな原因になっていたことが分かった。

第3章

多重解像度法に基づくフィッティング手法

前章では，平仮名のような複数のストロークを持つ文字画像を生成する確率モデルとして複数ストローク対応型伸縮変形モデルを提案した．このモデルは入力文字画像のストロークを探索する役割を担うインク発生器を用いている．フィッティングは目的関数の最小化問題に帰着されている．

前章で提案したフィッティング手法では画像処理などで良く用いられている粗密探索を実現するため，このインク発生器の大きさを最尤推定することによって徐々に小さくしている．しかしそれでも，実験の結果，複雑な構造を持つ文字はしばしば局所最適解に陥ってしまい，うまくフィットされないことが多いことが分かった．これはモデル側のみの粗密探索では不十分であることが原因と考えられる．モデル側の他に画像側にも粗密探索を実現すれば，局所最適解を回避することが期待できる．これを実現するため，多重解像度法の考えを導入する．しかし，前章で提案した複数ストローク対応型伸縮変形モデルは2値画像にしか対応していない．多重解像度法を導入するにはモデルをグレースケール画像に対応できるように拡張する必要がある．

3.1 では，前章で提案したモデルを拡張し，グレースケールの文字画像を生成するモデルを提案する．3.2 では，多重解像度法を導入したフィッティング手法を提案する．3.3 では，提案モデルを導入した認識システムについて述べる．3.4 では電子技術総合研究所提供の手書き文字データベース ETL8B[16] と ETL9B[17] の平仮名を使って評価実験によって，多重解像度法を用いた提案手法が有効であることを示す．

3.1 グレースケール文字画像の生成

前章で述べた複数ストローク対応型モデルは曲線上に配列されたインク発生器を通して多数の黒インクを発生させることにより2値画像を生成するものとしていた。本節ではグレースケール画像を生成するようにモデルを拡張する。この拡張により、変形した文字骨格と入力されたグレースケール文字画像との類似度が与えられる。

インク発生器はそれぞれの文字骨格上にそのサイズのほぼ倍の長さの間隔を置いて均等に配置される点は変わらない。インク発生器は濃度 $1/N$ ($N \gg 1$) の薄いインクを多数回発生させる (図3.1 参照)。すると、画素 k の濃度 d_k が $\lambda (= n/N)$ になるのは画素 k にモデルから n 回インクが発生されたとみなすことができ、その確率は次のように表される。

$$p(d_k = \lambda) = (p(d_k = 1/N))^n \quad (3.1)$$

$$= (p(d_k = 1))^\lambda \quad (3.2)$$

画素 k の濃度が1, すなわち黒になる確率 $p(d_k = 1)$ は式(2.10)を用いた。それぞれの画素のインクが独立に発生するものとする、文字画像 Z が発生する確率は、

$$P(Z|X, A, H) = \prod_{k \in Z} p(d_k = \lambda_k) \quad (3.3)$$

で表される。ただし、 λ_k は入力画像の画素 k の濃度である。 Z は文字画像を表し、これがある閾値以上の濃度¹をもつ画素の集合として定める。フィットエネルギーは

$$E_{\text{fit}} = -\log P(Z|X, A, H) \quad (3.4)$$

である。これは変形後の文字骨格と入力されたグレースケール文字画像との類似度を表しているといえる。

3.2 多重解像度法の適用

前章で提案した伸縮変形モデルは入力文字画像の確率モデルとして与えられ、そのフィッティングは目的関数の最小化問題に帰着されている。一般に複雑な確率モデルに関する推論は、困難な非線形最適化問題に帰着されることが多く、従来この問題を解くために最急降下法が用いられることが多かった [20], [21]。これは反復により勾配方向に向かって逐次解を改良する方法である。特に、ニューラルネットモデルの一種であるパーセプトロンでは誤差逆伝搬法と呼ばれる学習則に最急降下法を用いていることはよく知られている。前章で述べたモデルも最急降下法によってフィッティングさせることも可能だが、しかし、探索空間に多くの局所最適解が存在する場合、初期解の近くにトラップされるという問題があるため、他の最適化手法を選択する必要がある。前章で提案したモデルは不完全データからの学習アルゴリズムとして知られるEMアルゴリズム [15] を用いている。EMアル

¹閾値の設定は、計算量削減のためである。

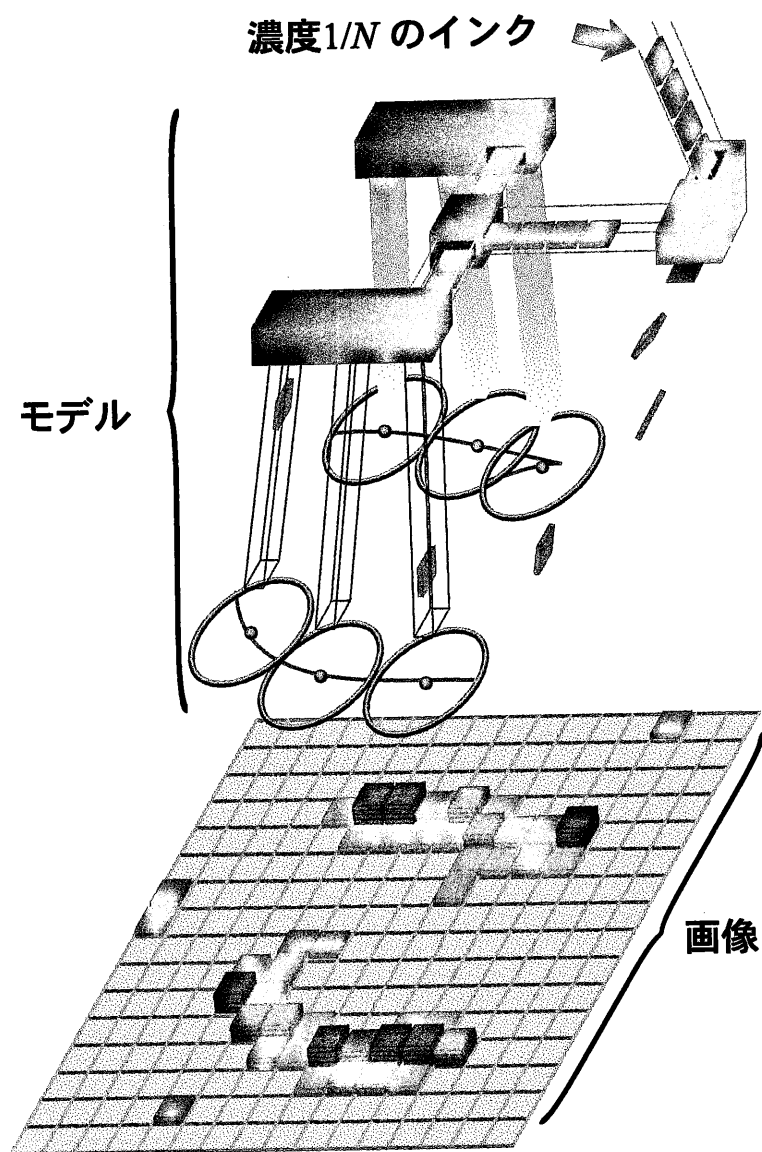


図 3.1: グレースケール画像生成機構

ゴリズムは、最急降下法と同様、解を逐次改良して反復探索する手法であるが、解空間が比較的複雑ではない場合における大域的最適解への良好な収束性や初期段階での速い収束性が知られており、音声認識でも隠れマルコフモデルの Baum-Welch アルゴリズムとして実用的に用いられている。

しかし、前章で述べた EM アルゴリズム単体では、インク発生器のサイズの値がイタレーションごとに小さくなっていくことを利用して粗密探索が実現されているが、文字が複数ストロークをもつなど複雑になると、これだけでは局所最適解に陥る場合がある。また切り出しミスなどによるノイズが存在すると、それも囲むように初期アフィン変換が設定され、ノイズが最後まで影響し、間違っただけの文字にフィットしてしまうことが起こる。数字のような簡単な構造の画像ではノイズを無視してフィットされることもあるが、平仮名のような複雑な構造の文字ではノイズの影響により局所最適解に陥ることが多い。

一方、局所最適解からの脱出を行う機構を導入したパラメータ探索手法として温度を導入して探索を確率的な振る舞いをさせた模擬徐冷法 [22], [23], [24] が提案され、画像理解の分野でしばしば用いられている [25], [26]。模擬徐冷法は Geman ら [23] によって温度スケジューリングを $T(k) = C/\log k + 1$ ($T(k)$:時刻 k の温度, C :定数) としたとき必ず大域的最適解に収束することが証明されたが、それには膨大な計算時間を要するという問題があるので、文字認識というタスクに用いるには適切ではないと考えられる。

局所最適解の脱出というアプローチをとるには確率的、ランダム的な探索という方法をとらざる得ず、探索時間の増大を逃れることはできない。そこで、探索空間自体を緩やかにすることで局所最適解の回避を試みることを考える。すなわち、まず緩やかにした探索空間における最適パラメータを求め、徐々に空間を複雑化して最適値を求めるというアプローチである。一般的にこのような手法は、統計物理学のアナロジーを使って温度という概念を導入し高温により大域的構造を表されたパラメータ空間から、徐冷により求める目的関数に近づけていく決定論的徐冷法と呼ばれ、クラスタリング問題 [27],[28] において有効性が確認されている。しかし、この手法は模擬徐冷法とは異なり、全ての最適化問題に対して汎用的に使用できるわけではない。本手法では、多重解像度法の考えを用いることによりこれを実現する。まず、低解像度の画像からフィッティングさせ、徐々に解像度をあげながら、フィッティングさせていく方法である。以下に、アルゴリズムを述べる。

1. 入力画像から解像度を $1/8, 1/4, 1/2, 1/1$ に下げた 4 つの画像を作成する。ここで 4 つ目の画像は原画像である。
2. フィッティング対象の画像を最も低い解像度の画像とする。
3. フィッティング対象の画像中のある閾値以上の画素集合、すなわち Z を囲む矩形を求め、 A, X を初期化する。
4. インク発生器のサイズを一つの画素の大きさの 1.5 倍にセットする。
5. EM アルゴリズムを適用して、目的関数の変化が θ_{tot} になるまでフィッティングを行なう。
6. フィッティング対象の画像を一つ高い解像度の画像とする。既にフィッティング対象の画像が原画像ならば、アルゴリズムを終了する。
7. 4 に戻る。

図3.2に多重解像度法を使ったフィッティングアルゴリズムの処理の流れを示す。図に示すように、まずステップ(1)で $\{X, A\}$ の初期化を行い、ステップ(2)–(5)で低い解像度の画像から順にフィッティングさせている。各ステップの説明の下にある図は画像「そ」に対してモデル「そ」にこのアルゴリズムを適用したときのそれぞれのステップの様子である。EMアルゴリズムは解の近くでは収束が遅くなる傾向にある。各ステップに対してEMの完全な収束を待つと計算量が増大してしまうのでステップ(2)–(4)では $\theta_{\text{tot}} = 0.01$ を用い、ステップ(5)のみ $\theta_{\text{tot}} = 0.002$ を用いた。

3.3 認識システム

図3.3に多重解像度法を導入した伸縮変形モデルを用いた認識システムの流れを示す。この認識システムでは、入力画像の2値化、画像正規化、特徴抽出といったプロセスは不要である。まず、入力された画像から、解像度を落とした画像を作成する。図では64x64の画像が入力されたとし、その画像から、32x32, 16x16, 8x8の3つの画像を作成している。この3つの画像と原画像はそれぞれの字種のモデルの整合器に送られる。整合器では前節で述べた多重解像度法を用いたフィッティングを行う。

3.4 評価実験

多重解像度法を用いたフィッティング手法の有効性を確認するため、手書き文字データベース ETL8B[16] と ETL9B[17] の平仮名を使って、認識実験を行なう。

3.4.1 ETL8B による評価実験

ETL8B で用いたサンプルや本システムの認識率の評価方法は2.6.1と同じである。

フィッティングアルゴリズムとしては、多重解像度法を用いない手法(手法A)と多重解像度法を用いた手法(手法B)との2種類について行った。また、2.6.1と同じく4種類の変形エネルギーについて比較実験を行った。

図3.4に認識率の平均を手法別に示す。いずれの変形エネルギーを用いた場合も手法Aよりも手法Bで高い認識率が得られ、多重解像度法を用いることの効果が確認された。また、縦横比を考慮した $E_{\text{def}}^{(a+cp)}$ の結果とストロークの端点間の間隔を考慮した $E_{\text{def}}^{(r+cp)}$ の結果は $E_{\text{def}}^{(cp)}$ を用いたものより、良好な結果が得られた。また、縦横比とストロークの端点間の間隔の両方を組み入れた $E_{\text{def}}^{(a+r+cp)}$ はこれら全てを上回る認識性能を示し、手法Bでは偶数番号をテスト用とした場合に99.29%、奇数番号をテスト用とした場合に98.91%、平均99.10%という非常に高い認識率が得られた。

標準パターン側のトップダウン的な変形を用いる手法のうち、同条件で認識性能が報告されている単純連続2次元ワープ法では、同じデータベースを偶数番号をテスト用、奇数番号を学習用に用いて97.0%が得られると報告されている[7]が、本手法はこれをはるかに

上回る認識性能を示した。

3.4.2 ETL9B による評価実験

ETL9B で用いたサンプルや本システムの認識率の評価方法は 2.6.2 と同じである。実験は 3.4.1 と同様に、手法 A と手法 B について行った。また、4 種類の変形エネルギーを比較した。結果を図 3.5 に示す。この結果も ETL8B 同様、多重解像度法を適用した手法 B の方が手法 A を上回り、最高で 98.20% の認識率が得られた。

図 3.6, 図 3.7 に本手法を用いてフィッティングさせた例を示す。図 3.6 は文字画像「か」にモデル「か」を使ってフィッティングさせた場合で、図 3.7 は文字画像「む」にモデル「む」を使った場合である。左から、初期状態、途中の状態、最終状態を示しているが、いずれもうまくフィットされている様子が確認される。

図 3.8 (a) は文字画像「ね」にモデル「ね」を多重解像度を使用せずにフィッティングさせた場合であるが、縦棒が縦棒にうまくフィットされず、インク発生器のサイズが大きいま目的関数が収束している。この文字は誤認識された。このように複数ストロークが似ている部分を含むような文字は数字のような簡単な構造の文字と比べ多くの局所最適解を持つてしまう。これに大域的な真の最適解を得ることを保証していない EM アルゴリズムを適用した結果、局所最適解に陥ったものと考えられる。多重解像度法を使用しない場合は、複雑な文字ほどこのような場合が起こりやすく、これが誤認識の要因の一つになっている。一方、多重解像度法を使用した図 3.8(b) は、局所最適解を回避し、良好な解を取得できた。その結果、正しく認識することができた。多重解像度法を使用しない場合は、パラメータの初期値として、文字の大きさや位置に応じてモデルの大きさや位置を揃えるようにアフィン変換を設定する。そのため、複雑な探索空間に対して真の最適解から遠い初期値から探索を開始しなければならない場合が多いと考えられる。これに対して、多重解像度法を用いる場合は入力された画像が複雑な探索空間をなしていても、解像度を十分に下げた画像から EM アルゴリズムを始めるため、その画像の空間の複雑性は入力画像に比べてある程度緩和していると考えられる。まずこの解像度を下げた画像に EM アルゴリズムを適用することによってその複雑性が緩和された空間における最適なパラメータを得ることができる。ここで得られたパラメータを初期値として次に少し解像度を上げた画像に EM アルゴリズムを適用してフィッティングさせる。このときその解像度の画像における最適解にパラメータが収束される。このように徐々に解像度を上げることによって個々の解像度の画像が比較的最適解に近い初期値から探索を開始することができる。最終的に入力画像にフィッティングさせる段階ではかなり大域的最適解に近いパラメータから最適パラメータの探索を開始できる。そのため、複雑な文字画像でも良好な解を得られることができたと考えられる。

切り出しミスなどによるノイズの影響も調査した。本手法ではモデルが入力画素 Z の大きさに合うようにアフィン変換の初期値が定められるが、図 3.9 (a) のようなノイズがある文字もこれを囲むように定められてしまう。「に」は左側の縦棒が分離しているため、このノイズの影響で第 1 ストロークがフィットできない。図 3.9 (b) は多重解像度を使用した

場合であるが、よい初期値から始められ、フィットすることができた。多重解像度画像を使用する場合も、初期アフィン変換は 3.2 で述べた方法で設定されるが、 Z はある閾値以上の濃度の画素のみ含めるので、結果的にノイズのような小さな点は低解像度画像では無視されることになる。

他の字種のモデルが文字画像の黒画素領域を包含するようにフィッティングされて誤認識されることがあった。図 3.10 は文字画像「な」にモデル「は」を使ってフィッティングさせた場合で、図 3.11 は文字画像「ら」にモデル「も」を使った場合である。いずれもモデルが文字のストロークを包含するようにフィッティングされている。これは、文字を識別する際の評価関数に含まれるフィットエネルギーが未知パターンとモデルとの類似度として扱うには不十分なのが原因と考えられる。何故なら、黒画素の同時発生確率の対数尤度で定義されているため、入力パターンのストローク以外の場所に多くのインク発生器が残されていたとしても、フィットエネルギーにはほとんど影響がない。そのため、異字種のモデルでも入力パターンを包含するような文字構造ならば、そのパターンを発生しやすくなってしまふ。

これを正しく認識させるには白画素上のインク発生器の存在に大きな負荷がかかるような関数を評価関数に追加すれば解決できると考えられる。しかし、そのような関数を加えるには入力文字画像とモデルが同じ字種の場合、正しくフィットするという前提が必要である。モデルのストロークが入力文字のストローク以外にも余分に伸びていたりすると白画素上のインク発生器の存在に大きな負荷がかかるような関数の値は大きくなり、かえって誤認識してしまう可能性がある。そこで、入力文字画像に対して同字種のモデルがうまくフィットされているか調査した結果、他のストロークが悪影響を与え、うまくフィットされていない場合が存在していることが分かった。図 3.12 は文字画像「う」にモデル「う」を使ってフィッティングさせた場合で、図 3.13 は文字画像「か」にモデル「か」を使ってフィッティングさせた場合で、図 3.14 は文字画像「ら」にモデル「ら」を使ってフィッティングさせた場合である。図 3.12 ではモデル「う」の第 1 ストロークが文字画像の第 2 ストロークに引っ付いてしまっている。図 3.13 ではモデル「か」の第 1 ストロークが文字画像の第 2 ストロークに引っ付いてしまっており、さらにモデル「か」の第 3 ストローク文字画像の第 2 ストロークに引っ付いている。図 3.14 ではモデル「ら」の第 1 ストロークが文字画像の第 2 ストロークに引っ付いてしまっている。さらに第 2 ストロークがうまくフィットされていない。このように同字種のモデルがインク発生器を白画素上に多数残してフィッティングアルゴリズムを終了してしまった場合があった。このような場合、白画素上のインク発生器の存在に大きな負荷がかかるような関数を評価関数に追加するとそのような関数の値が大きくなってしまい、誤認識されると予想される。故に、図 3.10 や 図 3.11 のような他の字種のモデルが文字画像の黒画素領域を包含する場合を正しく認識させるには、白画素上のインク発生器の存在に大きな負荷がかかるような関数の追加に加えて、さらなるフィッティング手法の改良が要求される。

3.5 まとめ

本章ではモデルを文字画像にフィッティングさせる際、局所最適解に陥ることを防ぐため、フィッティング手法に多重解像度法の考えを導入し、安定に良好な解が得られるアルゴリズムを提案した。このアルゴリズムの有効性を確認するため、電子技術総合研究所提供の手書き文字データベース ETL8B[16] と ETL9B[17] の平仮名を使って、認識実験を行った。その結果、多重解像度法を用いたフィッティング手法の有効性が示され、多重解像度法を使わない場合、複雑な文字にはしばしばパラメータ探索計算のかなり初期の段階で局所最適解にトラップされることがあったが、多重解像度法を使って段階的にパラメータを更新することによってこれを改善することができた。しかし、他の字種のモデルが文字画像の黒画素領域、つまり入力文字のストロークを包含することによって誤認識をしてしまう場合があった。これを正しく認識させるには白画素上のインク発生器の存在に大きな負荷がかかるような関数を評価関数に追加させれば解決できると考えられる。しかし、前章で述べたフィッティングには白画素上のインク発生器への対処が不十分なため、同字種のモデルのストロークが余分に伸びてフィッティングされることがある。このため、他の字種のモデルが入力文字のストロークを包含しても正しく認識させるために、白画素上のインク発生器の存在に大きな負荷がかかるような関数を評価関数に追加するにはまずさらなるフィッティング手法の改良が必要であると考えられる。

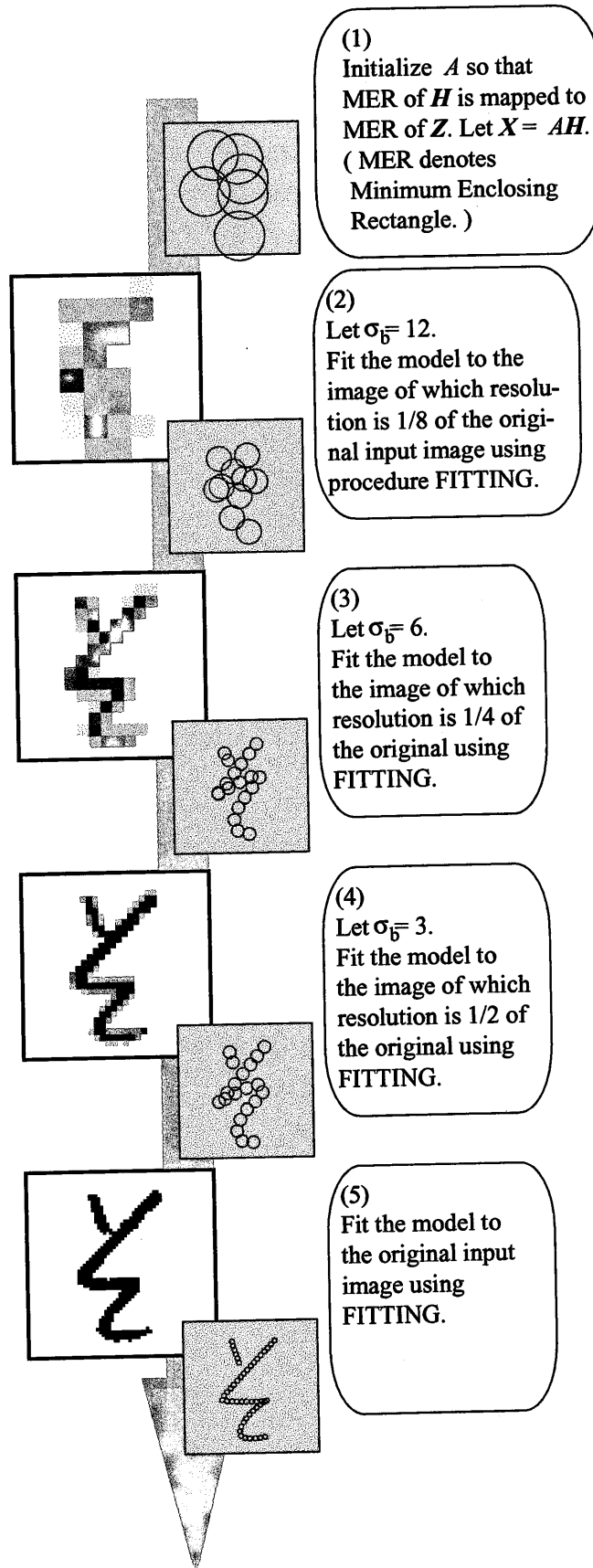


図 3.2: 多重解像度法によるフィッティングアルゴリズム

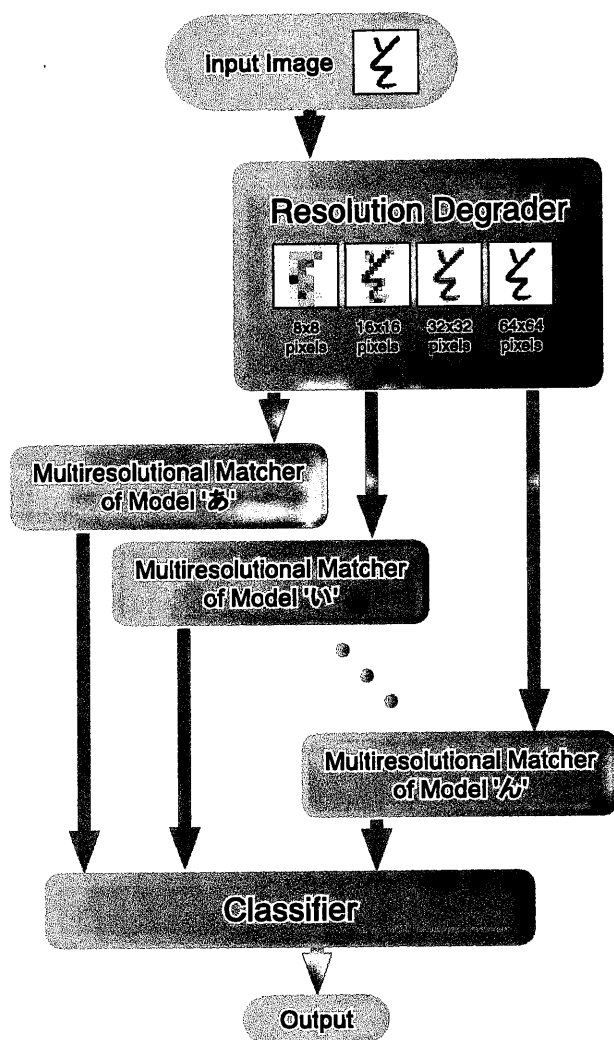


図 3.3: 認識システムの概要

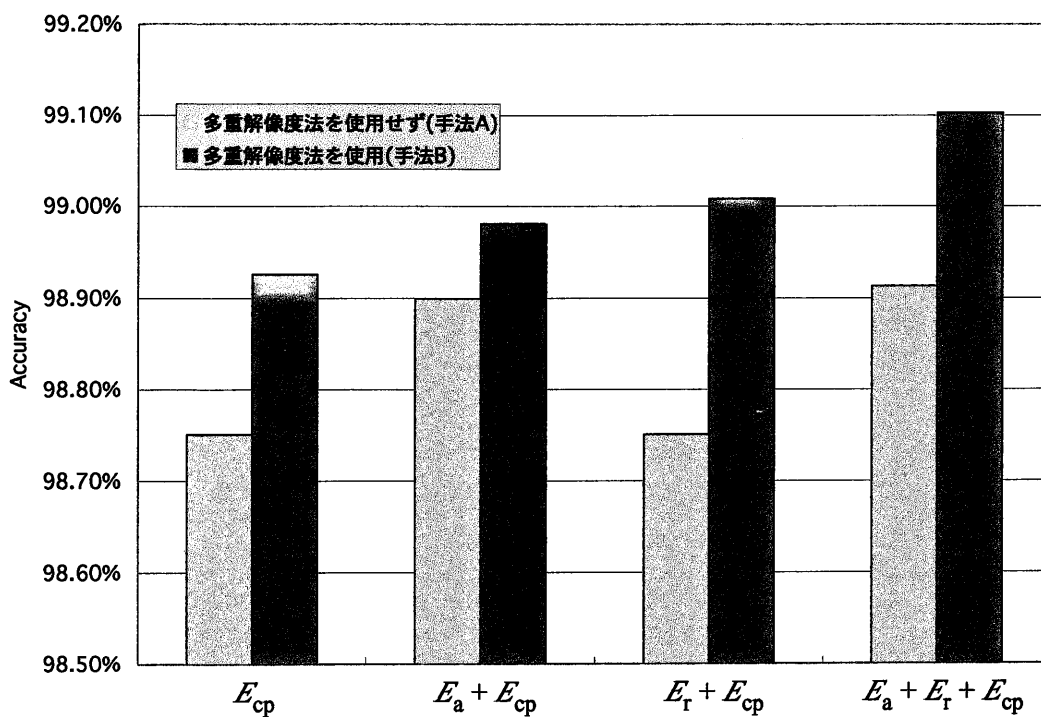


図 3.4: ETL8B による認識率

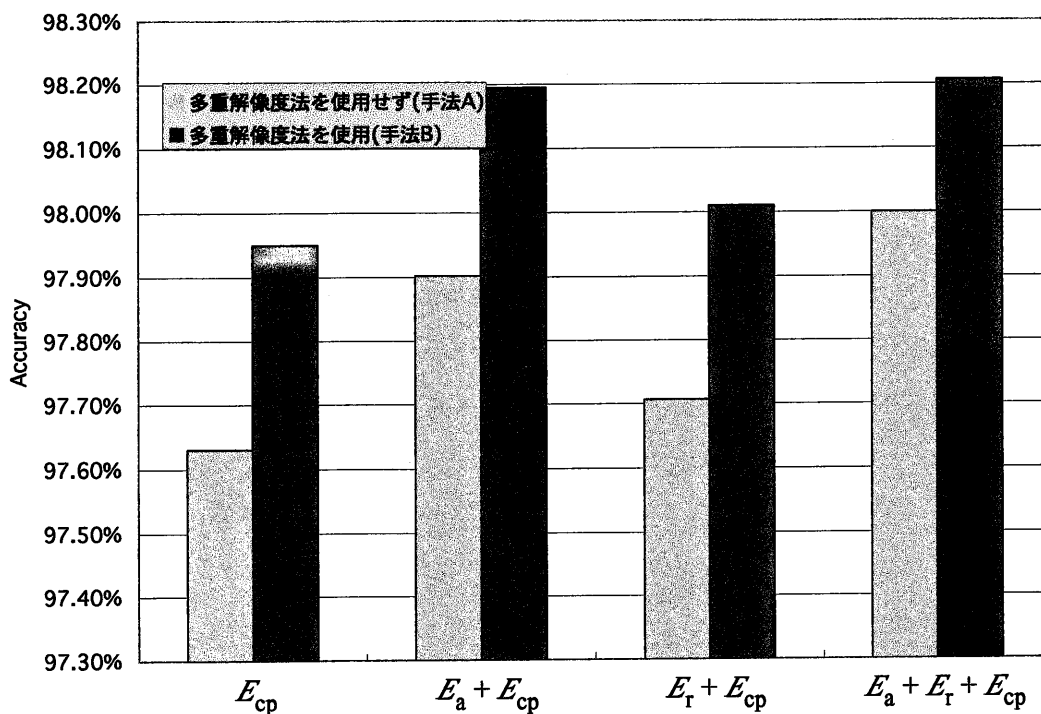


図 3.5: ETL9B による認識率

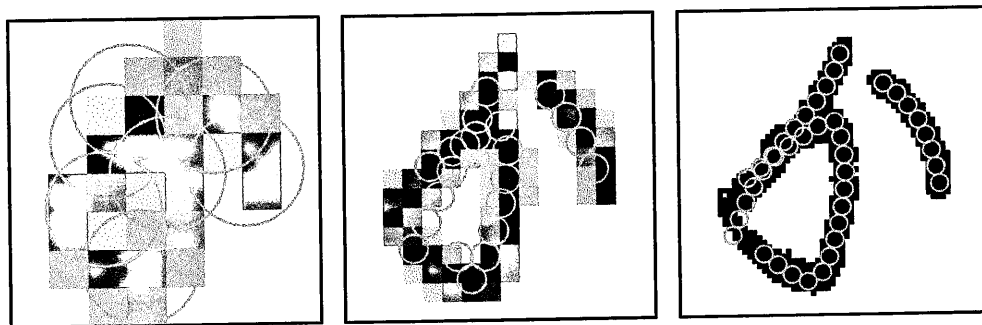


図 3.6: フィッティング例 その1

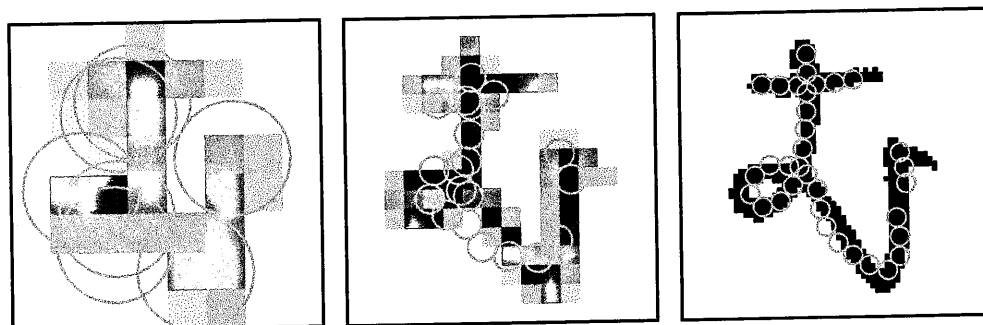
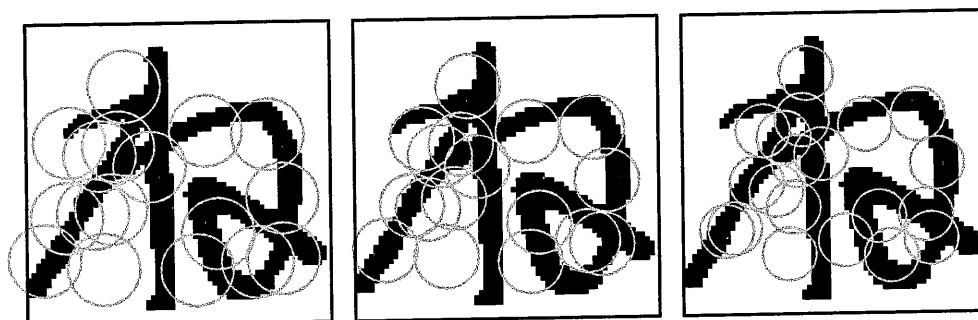
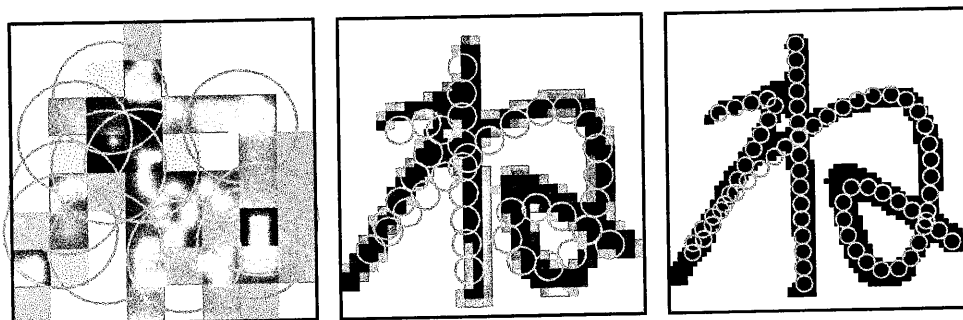


図 3.7: フィッティング例 その2

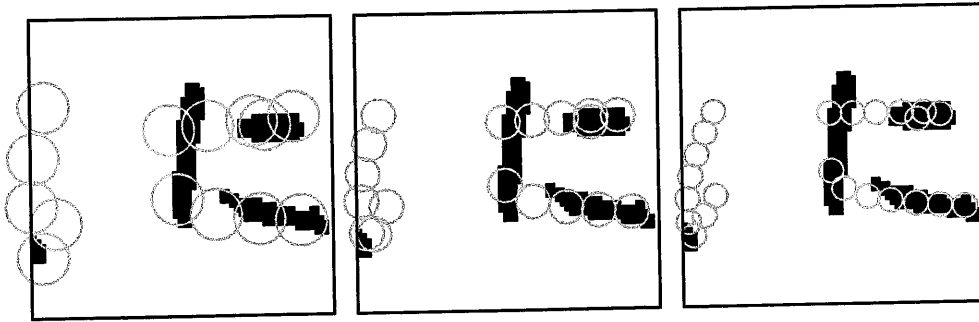


(a) Without using multiresolutional approach

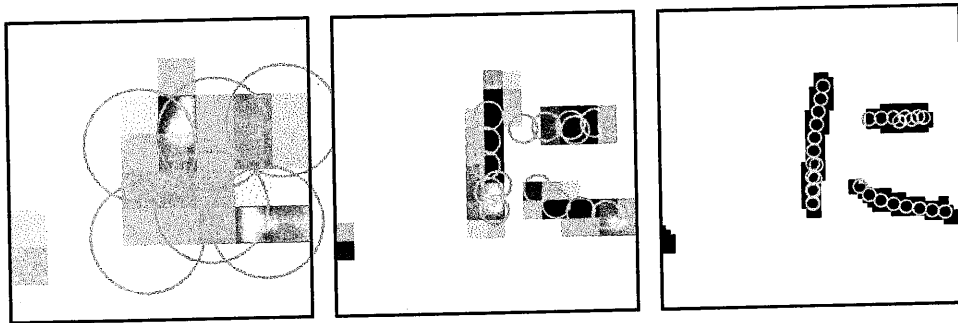


(b) Using multiresolutional approach

図 3.8: 多重解像度法による改善例 その1



(a) Without using multiresolutional approach



(b) Using multiresolutional approach

図 3.9: 多重解像度法による改善例 その2

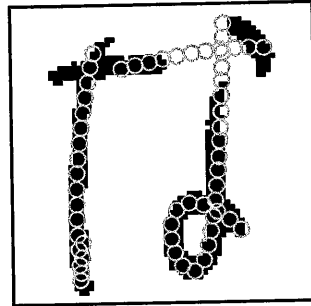


図 3.10: 誤認識例 その1

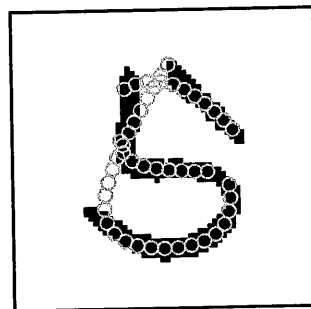


図 3.11: 誤認識例 その2

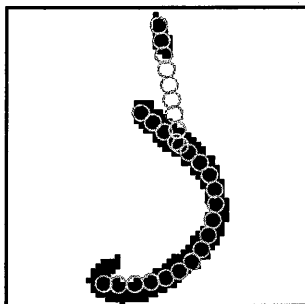


図 3.12: うまくフィットされていない例 その1

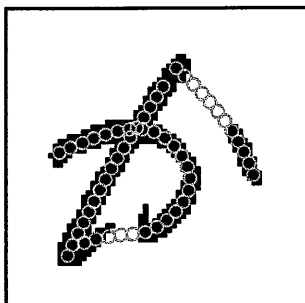


図 3.13: うまくフィットされていない例 その2

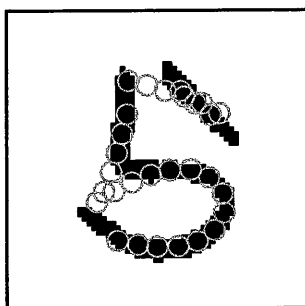


図 3.14: うまくフィットされていない例 その3

第4章

形状修正アルゴリズム

前章では、多重解像度法を用いてフィッティング手法を提案した。しかし、さらに改良が必要なことが分かった。前章の手法では他の字種のモデルが入力文字のストロークを包含することによって誤認識をしてしまう場合があった。これを正しく認識させるには白画素上のインク発生器の存在に大きな負荷がかかるような関数を評価関数に追加すれば解決できると予想される。しかし、調査の結果、同字種のモデルがインク発生器を白画素上に多数残してフィッティング過程が終了してしまう場合があることが分かった。このため、入力文字のストロークを他の字種のモデルが包含しても正しく認識させるために、白画素上のインク発生器の存在に大きな負荷がかかるような関数を評価関数に追加するだけでなく、フィッティング手法をさらに改良する必要がある。

本章では、フィッティング後のモデルの形状を修正し、うまくフィットさせる手法を提案する。また、識別用評価関数に改良を加え、さらに高精度に認識する手法を提案する。これらの提案手法の有効性を評価実験によって確認する。

4.1 形状修正アルゴリズム

本節では高精度なフィッティングを行うために、ストロークの形状を修正するアルゴリズムを提案する。このアルゴリズムを形状修正アルゴリズムと呼ぶ。形状修正アルゴリズムは以下の手続き FITTING_WITH_SHAPE_MODIFICATION によって行われる。

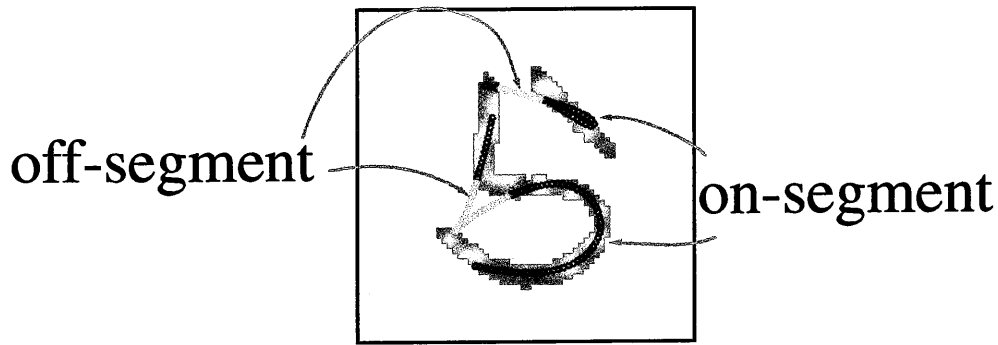


図 4.1: on/off-segment の特定

```

procedure FITTING_WITH_SHAPE_MODIFICATION()
begin
  フィッティング; ... (1)
  for  $i = 1$  to  $N_{\text{stroke}}$  do begin
    on-segment と off-segment を特定する; ... (2)
    for  $o_i \in O_i$  do begin
      冗長に伸びたストロークを短縮する; ... (3)
      on-segment 間を短縮し, 再結合する; ... (4)
       $o_i$  から最小二乗法を使って  $X^{(o_i)}$  を求める ... (5)
    end;
     $O'_i := \{X^{(o_i)} \mid C_{\text{len}}(o_i) \leq \min_{o \in O_i} (o) + \theta_{\text{len}}\}$  ... (6)
  end;
   $C' := \{X \mid C_{\text{MDL}}(X) \leq \min_{X' \in C} (X') + \theta_{\text{MDL}}\};$ 
  where  $C := O'_1 \times \dots \times O'_{N_{\text{stroke}}}$  ... (7)
  for  $X \in C'$  do 再フィッティング ... (8)
end.

```

まず, 多重解像度法によるフィッティングを行ったスプライン曲線 (1) のどの部分が黒画素領域上にあるか調べるため, インク発生器をスプライン曲線上に並べる. D_b が閾値以上ならば, そのインク発生器は黒画素領域上にあるとする. 黒画素領域上のインク発生器集合を on-segment と呼び, それ以外を off-segment と呼ぶことにする ((2), 図 4.1).

このとき, 第 i ストロークのスプライン曲線から n 個の on-segment に分かれたとする. この n 個のうちどれが対応するストローク上にあり, どれが異なるストローク上にあるか分からないので, すべての組み合わせ, つまり, $2^n - 1$ 個の組み合わせ集合 O_i について考慮する必要がある (図 4.2 参照).

この組み合わせそれぞれに対して, 冗長に伸びたストロークを短縮し ((3), 図 4.3), on-segment 間を切断して, 直線で再結合する ((4), 図 4.4). この修正された曲線に近づくように制御点 $X^{(o_i)}$ を最小二乗法を用いて計算する ((5), 図 4.5).

これをモデルのすべてのスプライン曲線に対して行い、それぞれの組み合わせを統合する。統合された組み合わせの妥当性を検証するため、MDL(Minimum Description Length) 原理に基づいた基準(式 (4.1)) により絞り込みを行う (7) ¹。

$$C_{\text{MDL}}(c) = E_{\text{def}} + E_{\text{fit}} + |\mathbf{B}| \log |\mathbf{Z}| \quad (4.1)$$

組み合わせ集合 C の中で $\min_{c \in C} C_{\text{MDL}}(c)$ との差が閾値以内のもの C' に対して、再フィッティングを行う (8)。形状修正アルゴリズムを伴ったフィッティングは図 4.6 のようにいくつかの仮説を出力して終了する。

4.2 認識システム

図 4.7 に形状修正アルゴリズムを導入した伸縮変形モデルを用いた認識システムの流れを示す。整合器では前節で述べた形状修正アルゴリズムを用いたフィッティングを行う。整合器は一つのモデルに対して複数の仮説を出す。識別器は後述の識別用評価関数が最小になる仮説を持つ字種のモデルを見つける。

4.3 文字の識別

3.1 で述べたフィットエネルギーを未知パターンと文字との類似度として扱うには不十分である。何故なら、 \mathbf{Z} の各濃度の同時発生確率の対数尤度で定義されており、最終的に文字を識別する際には黒画素の同時発生確率の対数尤度となる。従って、入力パターンのストローク以外の場所に多くのインク発生器が残されていたとしても、フィットエネルギーにはほとんど影響がない。そのため、異字種のモデルでも入力パターンを包含するような文字構造ならば、そのパターンを発生しやすくなってしまう。[14] ではこの問題に対し、字種ごとにヒューリスティックを設けて対処している。本論文では、白画素上のインク発生器の存在に大きな負荷がかかるようなエネルギー E_{white} を定義する。

$$E_{\text{white}} = -\frac{1}{|\mathbf{B}|} \sum_{b \in \mathbf{B}} \log D_b \quad (4.2)$$

$$D_b = \sum_{k \in \mathbf{Z}} \lambda_k N(\mathbf{z}_k | \mathbf{s}_b, \sigma_b^2) \quad (4.3)$$

$-\log D_b$ はインク発生器 b の周辺に黒画素がないとき、非常に大きな値となる。従って、そのようなインク発生器が存在すると、 E_{white} は大きくなる。識別器は E_{white} を用いて改良

¹ここで全組み合わせに対して、 $C_{\text{MDL}}(c)$ を計算するのはかなりの計算コストがかかるので、あらかじめストローク毎に以下の基準で大分類しておく (6)。

$$C_{\text{len}}(o) = w_{\text{on}} l_{\text{on}} + w_{\text{off}} l_{\text{off}} + \min_A N(A^{-1} \mathbf{X} | \mathbf{H}, \Sigma)$$

但し、 $l_{\text{on}}(l_{\text{off}})$ は on(off)-segment の長さで、 $w_{\text{on}}, w_{\text{off}}$ は重み定数である。on-segment の組み合わせ集合 O に対し、 $\min_{o \in O} C_{\text{len}}(o)$ との差が閾値以下の組み合わせ O' のみを残す。

した次の識別用評価関数が最も小さい仮説を持つ字種のモデルを見つける。

$$E_{\text{rec}} = w_{\text{def}}E_{\text{def}} + w_{\text{fit}}E_{\text{fit}} + w_{\text{white}}E_{\text{white}} \quad (4.4)$$

この評価関数をトータルエネルギーと呼ぶ。 w_{def} , w_{fit} , w_{white} は重み係数で予備実験により決める。

4.4 評価実験

本章で提案する認識システムの有効性を確認するため、電子技術総合研究所提供の手書き文字データベース ETL9B を使って、認識実験を行なう。

フィッティングアルゴリズムとして、多重解像度法を用いない方法(手法A)、多重解像度法を用いる方法(手法B)、さらに形状修正アルゴリズムを適用した方法(手法C)の3種類について行った。また、評価関数(式(4.4))における w_{white} を 0.0 とした場合と 8.0 とした場合を比較して、それぞれの手法における E_{white} の影響を調べた。

図 4.8 に認識率の平均を示す。また、形状修正アルゴリズムを用いた手法 C ではさらに良好な結果が得られた。 E_{white} を用いた場合、手法 A, B では用いない場合よりも認識率が下がったが、手法 C ではより高い認識率が得られた。

図 4.9 (a) は文字画像「う」にモデル「う」を形状修正アルゴリズムを使用せずにフィッティングさせた場合である。このような場合、白画素上に配置されたインク発生器に大きな負荷をかける目的で導入された E_{white} が悪影響をおよぼし、誤認識してしまう。一方、形状修正アルゴリズムを使用すると、(b) のように、うまくフィットさせることができ、結果 E_{white} を有効に用いることができ正しく認識された。同様に 図 4.10 (a) は文字画像「か」にモデル「か」を形状修正アルゴリズムを使用せずにフィッティングさせた場合で、(b) は形状修正アルゴリズムを使用した場合である。図 4.11 (a) は文字画像「ら」にモデル「ら」を形状修正アルゴリズムを使用せずにフィッティングさせた場合で、(b) は形状修正アルゴリズムを使用した場合である。図 4.10(a), 図 4.11 (a) は誤認識したが、形状修正アルゴリズムを使用した図 4.10(b), 図 4.11 (b) では正しく認識した。

図 3.10 や 図 3.11 も E_{white} により正しく認識された。

誤認識された文字の例を 図 4.12 に示す。中には、(a) のように人間でも読むのが困難な文字もあった。正解は「な」だが、本システムでは「え」と誤認識された。また、(b) のように「ろ」と見分けが付きにくい「う」などの誤認識があった。モデル「う」「ろ」両方とも小さな変形でフィットし、モデル「ろ」の方が僅差で評価値が小さく誤認識された。これを解決するには、さらなるヒューリスティックを加える必要があると思われる。

4.5 まとめ

本章ではフィッティング後のモデルの形状を修正し、フィッティング性能を向上させる手法を提案した。また、識別用評価関数に改良を加え、さらに高精度に認識する手法を提案した。これらの提案手法の有効性を評価実験によって確認した。

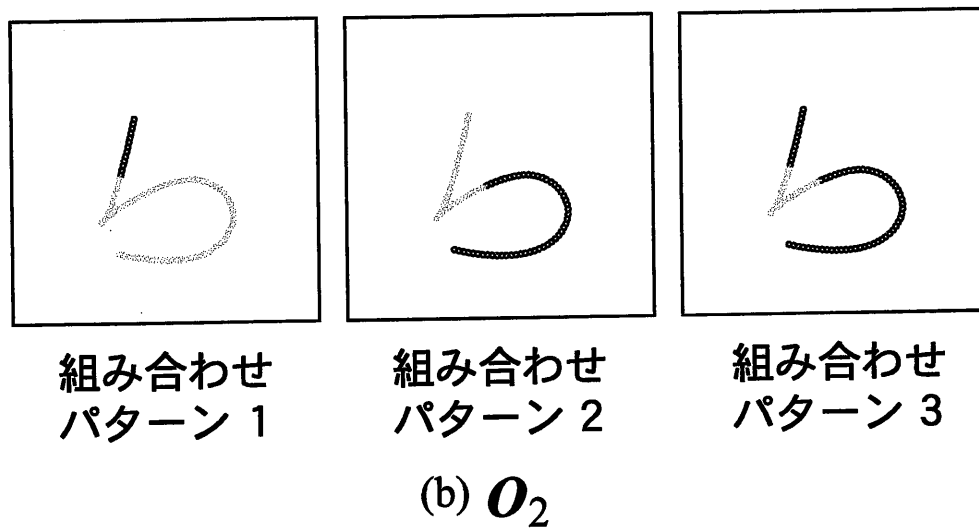
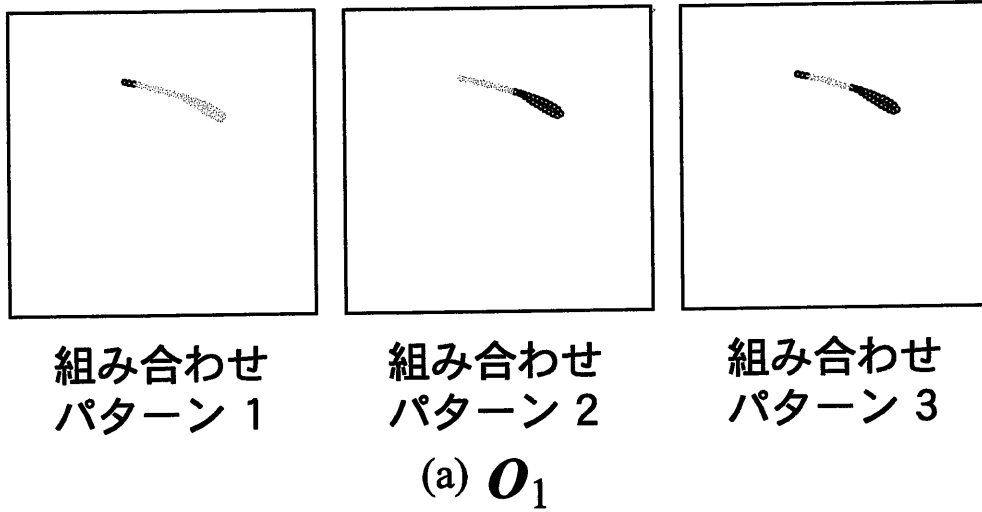


図 4.2: on-segment の組み合わせ集合

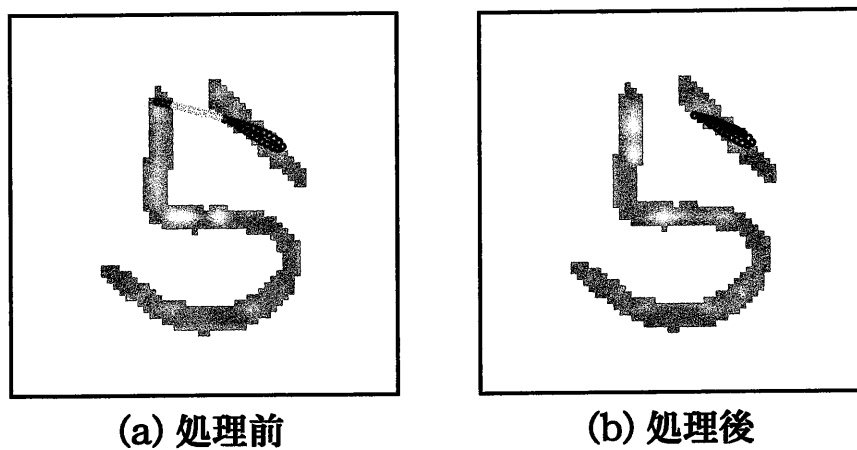


図 4.3: 冗長なストロークの短縮

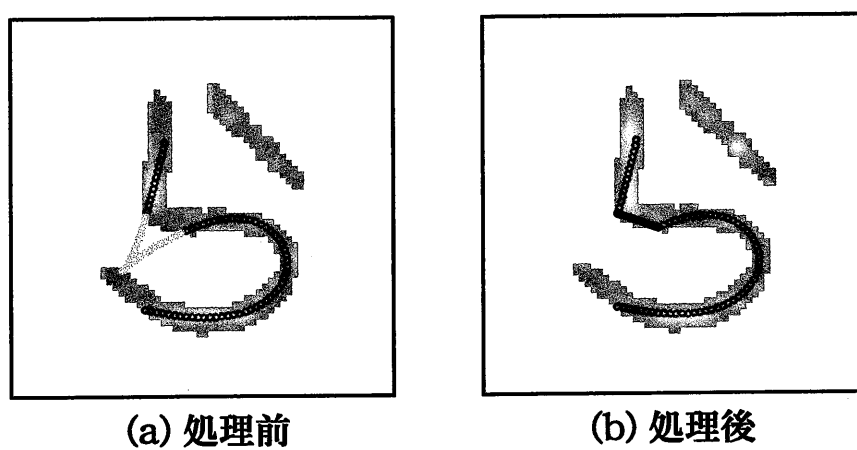
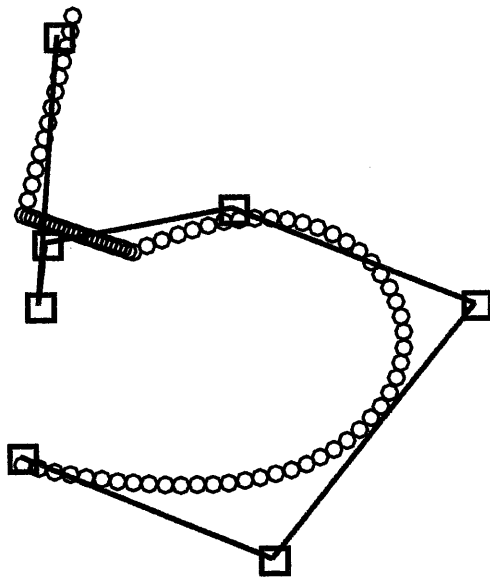


図 4.4: on-segment 間の再結合

図 4.5: 最小二乗法による $X^{(o_i)}$ の計算

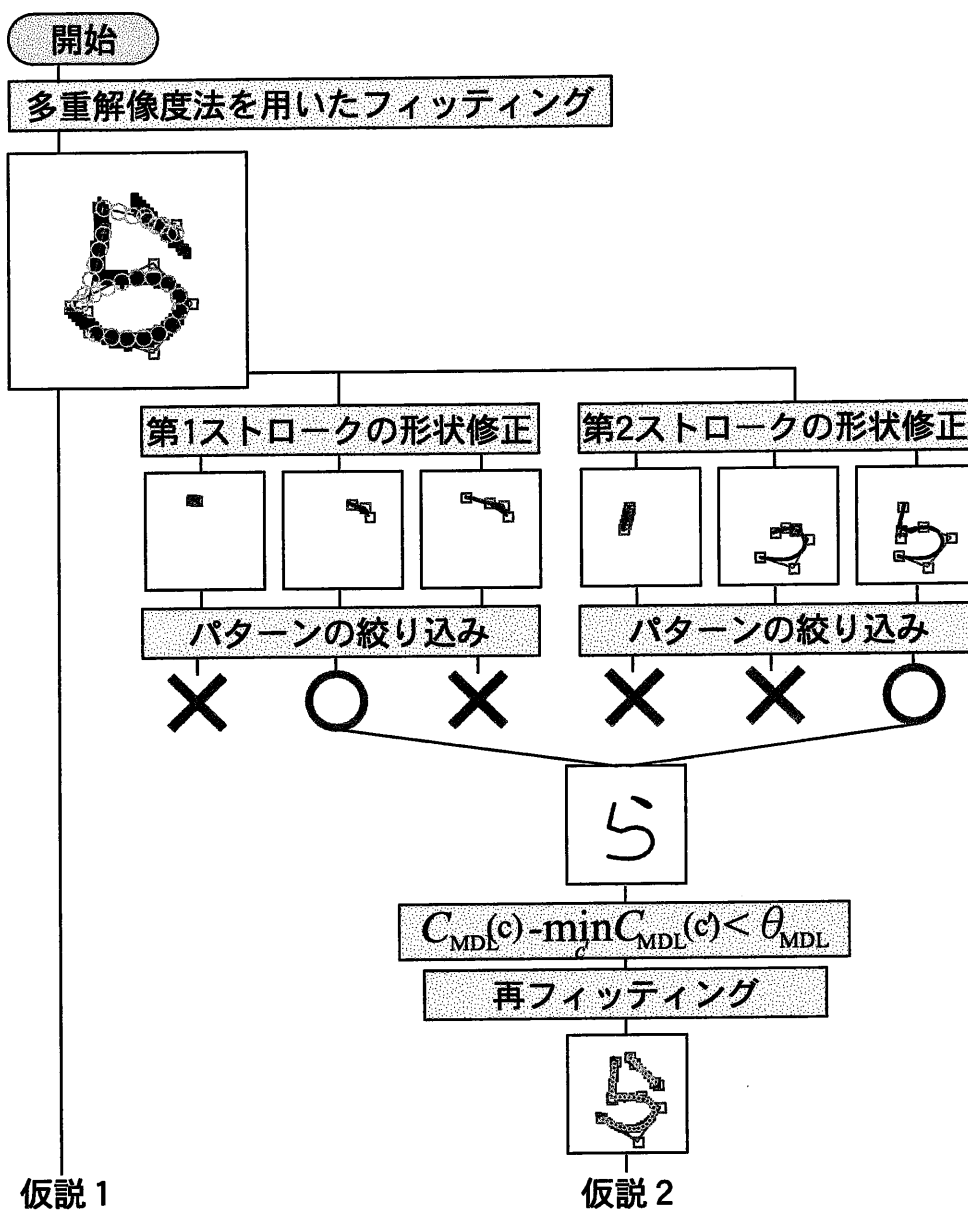


図 4.6: 形状修正アルゴリズムを伴ったフィッティング

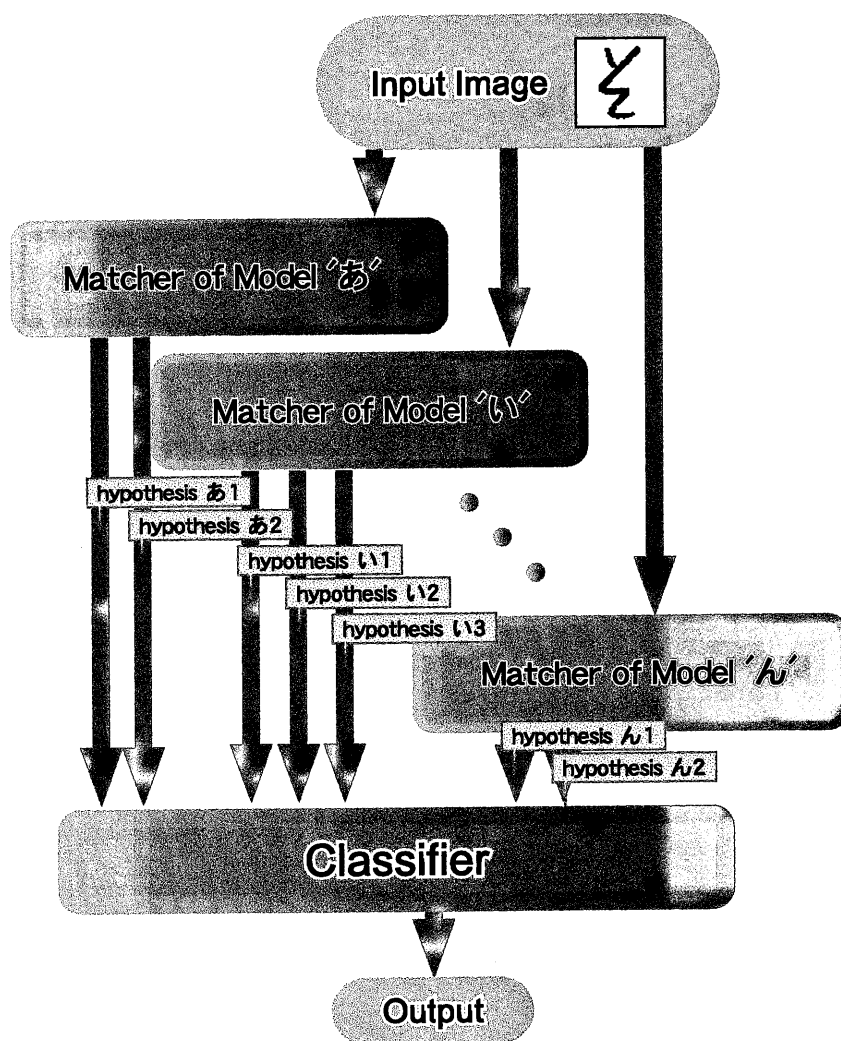


図 4.7: 認識システムの概要

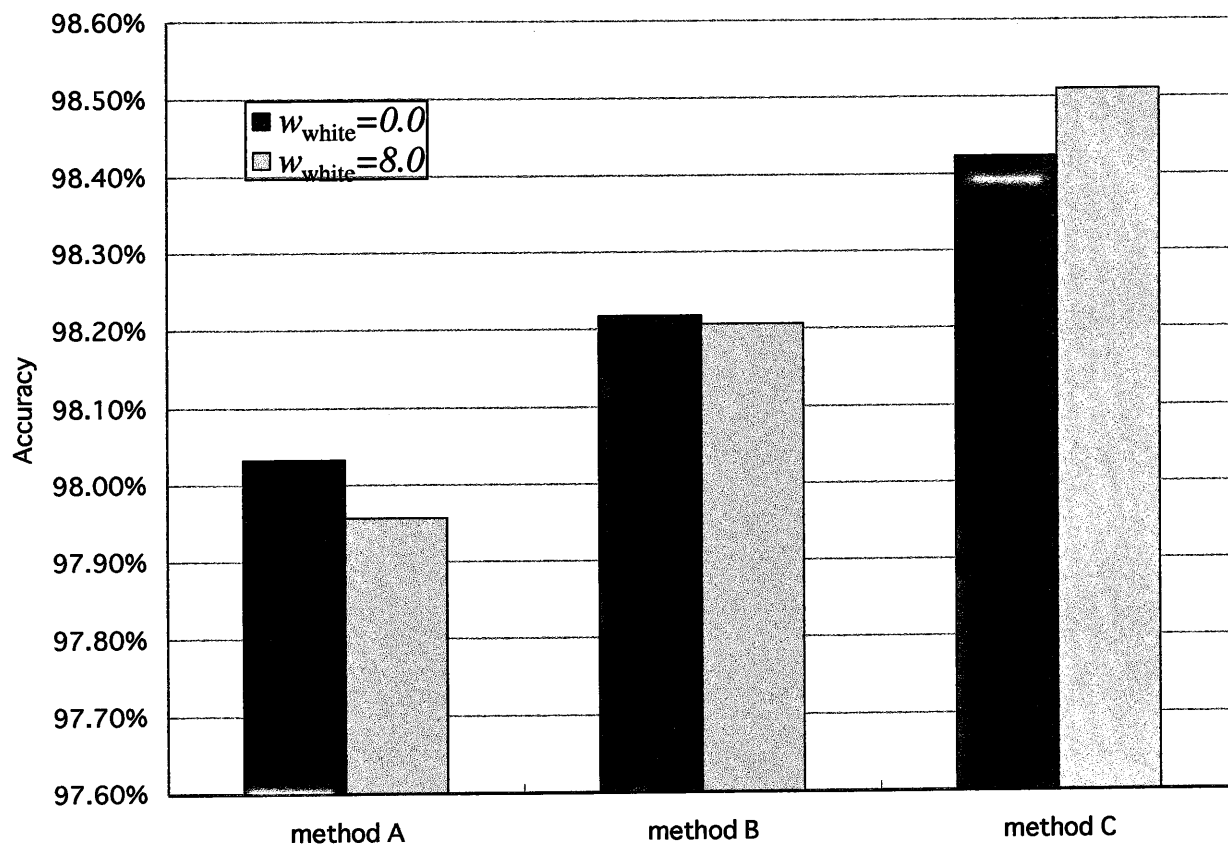


図 4.8: ETL9B による認識率

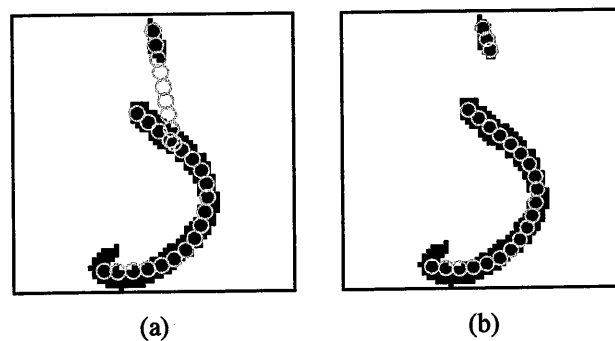


図 4.9: 形状修正アルゴリズムの効果 その1

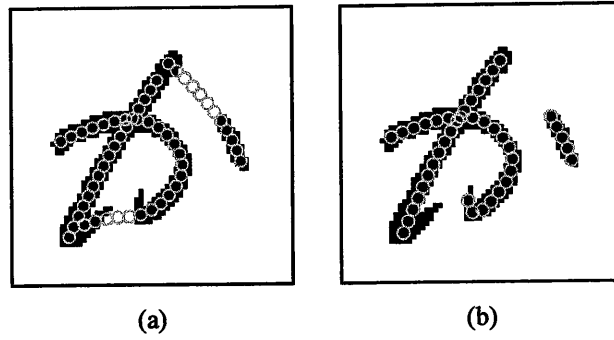


図 4.10: 形状修正アルゴリズムの効果 その2

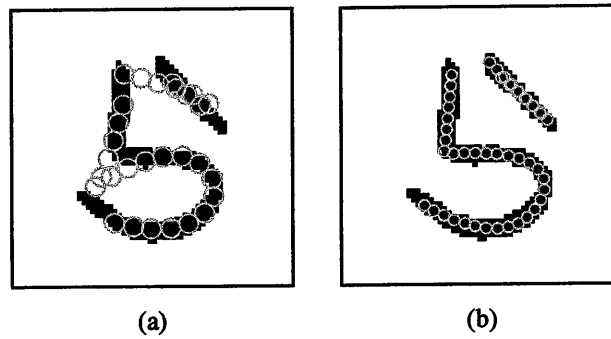


図 4.11: 形状修正アルゴリズムの効果 その3

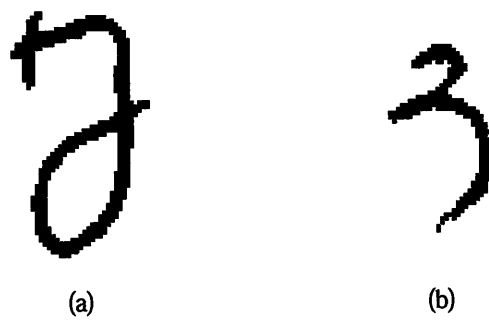


図 4.12: 誤認識された文字の例

第5章

結論

5.1 本論文の成果

本論文で得られた成果をまとめると、以下のようになる。

第2章では、では複数のストロークを持つ文字を少数のパラメータで表現できる変形に柔軟な複数ストローク対応型伸縮変形モデルを提案した。また、このモデルを使った有効な文字の識別手法を提案した。提案した複数ストローク対応型伸縮変形モデルの性能について調査し、また、識別用評価関数におけるモデルの変形の程度をはかる各項についての有効性を確認するため、平仮名を使って認識実験を行った。その結果、識別用評価関数の各項の有効性が示された。

第3章ではモデルを文字画像にフィッティングさせる際、局所最適解に陥ることを防ぐため、フィッティング手法に多重解像度法の考えを導入し、安定に良好な解が得られるアルゴリズムを提案した。認識実験の結果、多重解像度法を用いたフィッティング手法の有効性が示され、多重解像度法を使わない場合、複雑な文字にはしばしばパラメータ探索計算のかなり初期の段階で局所最適解にトラップされることがあったが、多重解像度法を使って階層的にパラメータを更新することによってこれを改善することができた。

第4章ではフィッティング後のモデルの形状を修正し、フィッティング性能を向上させる手法を提案した。また、識別用評価関数に改良を加え、さらに高精度に認識する手法を提案した。これらの提案手法の有効性を評価実験によって確認した。

これらの成果により、高精度な文字認識システムを構築するために、変形に頑健な伸縮変形モデルを実現し、その評価を行うことができたといえる。

5.2 今後の課題

本論文により、大きな変形を持つ文字にも高精度にフィッティングできるモデルが確立された。しかし、文字を識別するにはこれで本当に十分なのであろうか。本論文では一貫して文字の変形を変形エネルギーと呼ばれる関数で評価してきた。特に文字骨格の変形の度合いを調べるのに、線形変換を使って局所的な変形のみを取り出している。これにより、大きさ、回転、位置ずれを無視し、純粋な文字形状の歪みのみを抽出することが可能になった。だが、文字は大局的な変形は線形変換のみで表すことができるのだろうか。数字程度なら、確かに線形変換のみで十分だろうが、文字が複雑化すればするほど部分的に部位が拡大されたり縮小されたりすることも十分に考えられる。つまり、文字によっては大局的に非線形な変形をなしているものも考慮する必要があるだろう。しかし、どの字種がどの程度の自由度を持つ非線形変換を要するのか、決める基準が必要である。

本論文で提案するような反復計算を伴う手法は変形への頑健性が期待できるが、第1章で紹介したような重ね合わせ法を主体としたボトムアップ的システムのような高速性は得られない。近年のコンピュータの高速化には目を見張るものがあるが、現在のコンピュータの性能では文字認識というタスクを行うのには計算量が多いかもしれない。この計算量の問題も解決する必要がある。

実際に国内で書かれる文章に平仮名だけではなく漢字が多く含まれているのはいうまでもないが、本論文では基礎研究として平仮名のみを取り扱ってきた。本論文で提案する伸縮変形モデルは平仮名程度の複雑さには耐えうるが、漢字のような非常に複雑な文字にそのまま適用するのは難しいと考えられる。これも解決しなければならない問題である。

謝辞

本研究を進めるにあたり、研究生活全面にわたり数々の御指導とともにこの研究の機会を与えて下さった東北大学大学院工学研究科教授 阿曾弘具氏に深く感謝致します。特に、ゼミにおいては貴重なご意見を賜り、ありがとうございました。

研究室に配属されてから3年間始終一貫して心からのご指導と多くの享受を賜りました、東北大学大学院工学研究科助教授 大町真一郎氏には心から厚く感謝致します。先生のご鞭撻なくして本研究の成果はあり得なかったと思います。御礼申し上げます。

暖かいお心遣いをいただいた東北大学大学院工学研究科教授 川又政征氏、本研究にご興味をお示しいただいた東北大学大学院情報科学研究科教授 堀口剛氏は私の大きな活力となりました。感謝いたします。

日々の研究におきまして、計算機環境を整えていただいた菅谷至寛氏に感謝致します。また、宇都宮大学工学部助手 森大毅氏は計算機環境の整備にとどまらず、研究室の規範としてモラル向上にご貢献され、氏のおかげで安心して研究に勤しむことができました。この場を借りて謝意を表します。有益なご意見をいただいた佐藤俊治氏にも感謝申し上げます。氏には多岐にわたって度々相談申し上げましたが、そのたびに的確なご助言をいただきました。広島大学総合科学部助手 黒岩丈介氏には4年ゼミ、研修Aにおいて心からのご指導をいただきました。研究生活をご一緒できた期間は短かったですが、今日に至っても感謝の念が絶えることはありません。李 信行氏は私に世界的な視野をお与えくださいました。ありがとうございます。東北文化学園大学助手 孫方氏には研究生活全般に渡って貴重なご助言を戴きました。また、様々な相談に対する親身のようなご対応には感激しております。誠にありがとうございました。

最後に、多岐に渡ってお世話になった阿曾研究室の皆様、今日に至るまで絶え間ないご支援をいただいた私の両親に感謝致します。

参考文献

- [1] N. Kato, M. Suzuki, S. Omachi, H. Aso and Y. Nemoto: “A handwritten character recognition system using directional element feature and asymmetric mahalanobis distance”, *IEEE Trans. Pattern Analysis and Machine Intelligence*, **21**, 3, pp. 258–262 (1999).
- [2] R. G. Casey: “Moment normalization of handprinted characters”, *IBM Journal of Research and Development*, pp. 548–557 (1970).
- [3] H. Yamada, Yamamoto and T. Saito: “A nonlinear normalization method for handprinted kanji character recognition — line density equalization —”, *Pattern Recognition*, **23**, pp. 1023–1029 (1990).
- [4] T. M. Ha and H. Bunke: “Off-line, handwritten numeral recognition by perturbation method”, *IEEE Trans. Pattern Analysis and Machine Intelligence*, **19**, 5, pp. 535–539 (1997).
- [5] A. K. Jain and D. Zongker: “Representation and recognition of handwritten digits using deformable templates”, *IEEE Trans. Pattern Analysis and Machine Intelligence*, **19**, 12, pp. 1386–1391 (1997).
- [6] 水上, 古賀, 鳥岡: “変位抽出を行なう手書き文字認識システム”, *信学論 (D-II)*, J80-D-II, **19**, 12, pp. 1386–1391 (1997).
- [7] 内田, 追江: “単調連続 2 次元ワープ法によるオフライン手書きひらがな認識”, *信学技法*, PRMU98-163, pp. 103–110 (1998).
- [8] M. Nakagawa, T. Yamagida and T. Nagasaki: “An off-line character recognition method employing model-depend pattern normalization by and elastic membrane model”, *Proc. ICDAR '99*, pp. 495–498 (1999).
- [9] T. Wakahara: “Shape matching using LAT and its application to handwritten numeral recognition”, *IEEE Trans. Pattern Analysis and Machine Intelligence*, **16**, 6, pp. 618–629 (1994).

- [10] L. Lam and C. Y. Suen: "Structural classification and relaxation matching of totally unconstrained handwritten zip-code numbers", *Pattern Recognition*, **21**, 1, pp. 19-31 (1988).
- [11] 山本: "弛緩整合法による手書き教育漢字認識", *信学論*, J85-D, 9, pp. 1169-1174 (1982).
- [12] M. D. Revow, C. K. I. Williams and G. E. Hinton: "Using generative models for handwritten digit recognition", *IEEE Trans. Pattern Analysis and Machine Intelligence*, **18**, 6, pp. 592-606 (1996).
- [13] D. MacKay: "Bayesian interpolation", *Neural Computation*, **4**, 3, pp. 415-447 (1992).
- [14] K. Cheung, D. Yeung and R. T. Chin: "A Bayesian framework for deformable pattern recognition with application to handwritten character recognition", *IEEE Trans. Pattern Analysis and Machine Intelligence*, **20**, 12, pp. 1382-1388 (1998).
- [15] A. P. Dempster, N. M. Laird and D. B. Rubin: "Maximum likelihood from incomplete data via the EM algorithm", *J.R. Statistical Society, Series B*, **39**, pp. 1-38 (1977).
- [16] S. Mori, K. Yamamoto and M. Yasuda: "Research on machine recognition of hand-printed characters", *IEEE Trans. Pattern Analysis and Machine Intelligence*, **6**, 4, pp. 384-405 (1984).
- [17] T. Saito, H. Yamada and K. Yamamoto: "On the database ETL9 of handwritten characters in JIS Chinese characters and its analysis", *Trans. IEICE J68-D(4)*, pp. 757-764 (1985).
- [18] P. Huber: "Robast Statistics", Wiley (1981).
- [19] X. L. Meng and D. B. Rubin: "Maximum likelihood estimation via the ECM algorithm: a general framework", *Biometrika*, **80**, pp. 267-278 (1993).
- [20] A. K. Jain, Y. Zhong and S. Lakshmanan: "Object matching using deformable templates", *IEEE Trans. Pattern Analysis and Machine Intelligence*, **18**, 3, pp. 267-278 (1996).
- [21] A. D. Bimbo and P. Pala: "Visual image retrieval by elastic matching of user sketches", *IEEE Trans. Pattern Analysis and Machine Intelligence*, **19**, 2, pp. 121-132 (1997).
- [22] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Reller and E. Teller: "Equations of state calculations by fast computing machines", *J. Chem. Phys.*, **21**, pp. 1087-1092 (1953).
- [23] S. Geman and D. Geman: "Stochastic relaxation, Gibbs distribution and the Bayesian restoration in images", *IEEE Trans. Pattern Analysis and Machine Intelligence*, **6**, 6, pp. 721-741 (1984).

- [24] S. Kirkpatrick, C. D. Gelatt and M. P. Vecchi: "Optimization by simulated annealing", *Science*, **220**, pp. 671-680 (1983).
- [25] M.-P. D. Jolly, S. Lakshmanan and A. K. Jain: "Vehicle segmentation and classification using deformable templates", *IEEE Trans. Pattern Analysis and Machine Intelligence*, **18**, 3, pp. 293-308 (1996).
- [26] G. Storvik: "A Bayesian approach to dynamic contours through stochastic sampling and simulated annealing", *IEEE Trans. Pattern Analysis and Machine Intelligence*, **16**, 10, pp. 976-986 (1994).
- [27] K. Rose, E. Gurewitz and G. C. Fox: "A deterministic annealing approach to clustering", *Pattern Recognition Letters*, **11**, 9, pp. 589-594 (1990).
- [28] J. Puzicha and J. Buhmann: "Multiscale annealing for grouping and unsupervised texture segmentation", *Computer Vision and Image Understanding*, **76**, 3, pp. 213-230 (1999).

研究業績

学術論文

加藤毅, 大町真一郎, 阿曾弘具,
“伸縮変形モデルを用いた手書き文字認識,” 電子情報通信学会論文誌 (D-II) (投稿中).

査読付き国際会議

Tsuyoshi Kato, Shin'ichiro Omachi and Hiroto Aso
“Precise Hand-printed Character Recognition Using Elastic Models via Nonlinear Transformation,” 15th International Conference Pattern Recognition (投稿中).

学会発表等

加藤毅, 大町真一郎, 阿曾弘具,
“平仮名認識のための複数ストローク対応型伸縮変形モデル,” 電気関係学会東北支部
連合大会, 2G-18, p.280, 1998年8月.

加藤毅, 大町真一郎, 阿曾弘具,
“複数ストローク対応型伸縮変形モデルを用いた平仮名認識,” 信学技法, PRMU98-162,
pp.95-102, 1998年12月.

加藤毅, 大町真一郎, 阿曾弘具,
“多重解像度法を用いた文字認識用伸縮変形モデルの高精度化,” 信学技法, PRMU99-
130, pp.213-220, 1999年11月.