

卒業論文

文字パターンに関する基礎的研究

東北大学工学部 電気・情報系
阿曾研究室 4年 水庭 威幸

1995年3月24日

目次

1	序論	3
1.1	研究の背景と目的	3
1.2	本論文の構成	4
2	文字認識	5
2.1	文字入力・前処理	5
2.2	特徴抽出	7
2.3	認識	7
2.4	候補出力	8
3	変形文字の生成手法	12
3.1	手書き文字の変形に関する過去の研究	12
3.2	本研究における変形文字の生成方法	15
4	変形実験	17
4.1	実験結果	17
4.2	考察	17
5	認識実験	19
5.1	辞書の作成方法	19
5.2	実験方法	19
5.3	実験結果	19
6	まとめ	24
	謝辞	25
	参考文献	26

第 1 章

序論

1.1 研究の背景と目的

情報処理技術の急速な発展にともない、計算機の扱う情報の量はますます増大し、多用化、複雑化している。しかし、計算機の発展の速度に比べ、人間と計算機間の情報交換手段はそれほど改善されていない。もし、我々が普段情報交換のために用いている、手書き文字、あるいは印刷文字を自動的に計算機に入力できれば、それは人間にとって、便利で容易な手段と言える。さらに、近年では OA 化により、ワープロやファックス、コピー機などの事務機器が産業から家庭に広がっており、大量の文書が出回っている。このような大量の文書の保存、変更、管理を計算機によって行なうという社会的要求が強まってきた。そういう意味において、急速に発展している情報化社会のなかで、文字認識技術の重要さは音声、画像技術と並んで、ますます重みが増してくると考えられる。

文字認識についての研究は三十年前から行なわれ、種々の成果が得られている。現在、特定の分野においては OCR (Optical Character Reader) が実用化されている。しかし、現状では認識率 100% の文書認識システムは完成されておらず、どうしても誤認識を生じてしまう。誤認識がある場合には、修正に手間がかかり過ぎるために多くの労力が必要とされる。このため、大量の文書認識の自動化が困難となっており、文書認識のさらなる高性能化が要求されている。

このような要求に対する一つの手段として、大量のサンプルを用いることにより、文字パターンの特徴量空間での分布形状を知ることができれば、誤認識を最小におさえること

ができると考えられている。しかし、現状ではサンプル数が少ないため、文字パターンの特徴量空間での分布形状を推定することができない、という問題がある。

本研究では、このような問題に対処するため、印刷文字の変形パターンを大量に自動生成し、文字認識性能の向上につなげることを目的とする。まず、変形文字の生成手法について述べ、次に、実際に変形させた文字を、見た目と実験により評価する。

1.2 本論文の構成

本論文の構成は次の通りである。

第 1 章 序論であり、本研究の背景と目的を述べる。

第 2 章 文字認識アルゴリズムについて説明する。

第 3 章 変形文字の生成手法について説明する。

第 4 章 実際に変形させた文字を示し、見た目での評価を行なう。

第 5 章 変形パターンを加えて作成した辞書を用いて認識実験を行ない、結果を検討する。

第 6 章 本研究のまとめ、今後の課題について述べる。

第 2 章

文字認識

文字認識のアルゴリズムには、大きく分けて次の二つの方法がある。

- パターンマッチング法
- 構造解析法

パターンマッチング法は、パターン同士の重なり具合で評価し、認識を行なう。このため、文字の多少の変形やノイズには強いが、類似文字の識別は難しい。また、計算機上で容易に高速で実現できる等の特徴を持つ。

これに対して構造解析法は、線分の接続関係や位置関係などの文字構造に着目し、構造の類似性で認識を行なう。この手法は類似文字の多い漢字の認識や、個人のくせが顕著である手書き文字の認識に有効であるとされているが、特徴量の定義や抽出が難しく、また計算機上での処理に時間がかかる等の問題も多い。

以後、本研究で用いるパターンマッチング法について説明する。図 2.1 に文字認識のアルゴリズムを示す。

2.1 文字入力前処理

- 文字入力

認識される文字データは文書の形でイメージスキャナによって入力される。入力されたイメージデータは、切り出し処理により、各文字毎に切り出される。

- 前処理

切り出された文字データには、実際にマッチングに使われる特徴量を求めるために前処理が施される。前処理は、ノイズ除去・スムージング、正規化、細線化、線素化の4つの処理によって構成されている。

- ノイズ除去・スムージング

入力画像（切り出されたデータ）は印刷時のプリントの状態やイメージスキャナなどの性能によって、ノイズや線分上の凹凸が少なからず発生している。これらは認識精度に悪い影響を及ぼすため、まず入力画像に対してノイズ除去を行なう。本研究では2×2ドット以下の孤立点をノイズと見なし除去することにする。スムージングは、周囲8近傍を参照し、中心のドットを決定する。実際には3×3ドットのマスクを用いて行なう。

- 正規化

正規化は、もとの入力イメージ画像を一定の大きさ（本研究では64×64ドット）に拡大又は縮小する処理であり、これによって文字の大きさや領域の位置ずれによる影響を吸収する。正規化の手法には大きく分けると、線形正規化と非線形正規化の二つの手法に分けられるが、本研究では線形正規化を用いている。

- 細線化

細線化は、通常数ドットの太さを持った文字の線分を1ドットの幅に変換する処理である。この処理によって、文字の線分幅の違いを吸収する。本研究では、各画素においてその周りの3×3のマスクを用いて、着目するドットの近傍の8ドットの連結状態を見ながら細線化していく Hilditch の方法を用いる。

- 線素化

線素化は、細線化された図形の各画素について、黒画素であればその周囲3×3の小領域を参照し最も自然な方向と考えられる4つの縦「|」、横「-」、斜め45°「/」、斜め135°「\」、のうちの一つの線素に対応させる。

前処理の各処理による効果を図 2.2 に示す。

2.2 特徴抽出

パターンマッチング法では、処理の高速化、パターン分離の効率化などのために、文字パターンを特徴量という数値ベクトルに変換する。この過程を特徴抽出という。以降は本研究で用いる方向線素特徴量 [1]、について説明する。

- 方向線素特徴量

方向線素特徴量の抽出法は図 2.3 のように、まず、 64×64 ドットの線素化文字を 8ドット間隔に縦横を分割する。次に、左上から 16×16 ドットを半分ずつ重複させて 49 領域（左上を 0 とし、左から右、上から下へ順に並ぶ）ができる。1 領域のベクトルは 4 方向のカウンタからなる 4 次元ベクトルであり、1 領域内の重みは図 2.3 の下図のようになっている。よって 1 文字当たりの次元数は $N = 196$ (49×4) 次元のベクトルとなる。

2.3 認識

ここでは最も一般的な手法である全数整合法を説明する。全数整合法は、各文字の標準パターンと特徴抽出で得られた未知入力文字の特徴量で評価値（距離）を求め、近いものから順に識別する方法である。アルゴリズムの簡潔性やある程度の高い認識率を得られることから、一般的に用いられている。本研究で用いる辞書、評価値について以下で説明する。

- 辞書の作成方法

パターンマッチング法では、入力されたパターンと標準パターンとを比較し、一番似ている（距離的に近い）標準パターンのコードを結果として出力する。この標準パターンは辞書と呼ばれ、予め全ての登録文字に対して用意しておかなければならない。辞書作成法としては、文字 k のサンプルパターンの集合 $S(k)$ の重心ベクトルが一般的である。

- 評価値

未知入力文字の特徴量と標準パターンとの類似性や整合性の評価は、評価値により行なわれる。評価値の例としてよく用いられるものとして、ユークリッド距離（実際は

その 2 乗) がある。ここで、 \mathbf{x} 、 \mathbf{u} をそれぞれ未知入力ベクトルと、ある文字の標準パターンベクトルとすると、次のように定義される。

$$D(x, y) = \sum_{i=1}^n (x_i - u_i)^2$$

2.4 候補出力

評価計算をした後、評価値の小さい順に、第 1 位候補、第 2 位候補、…、第 N 位候補を出力する。

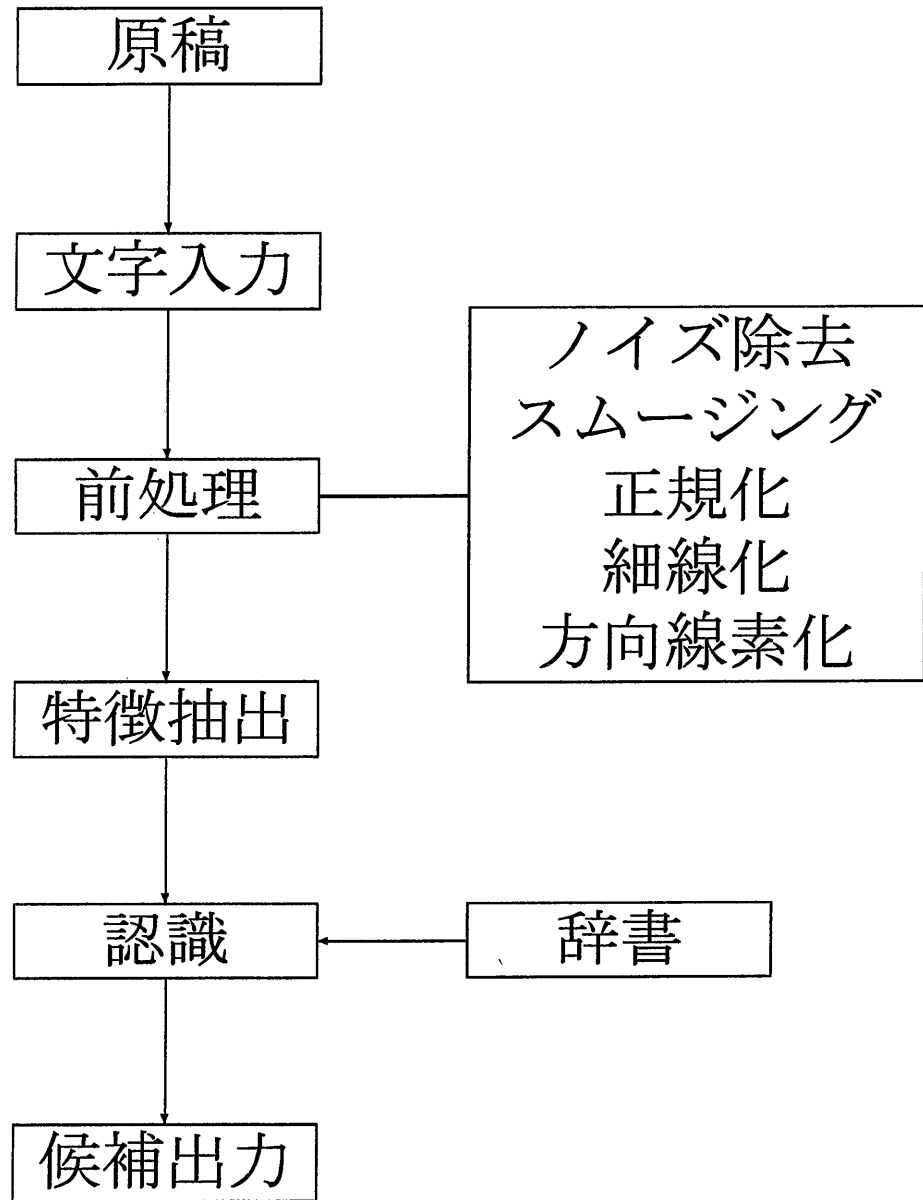
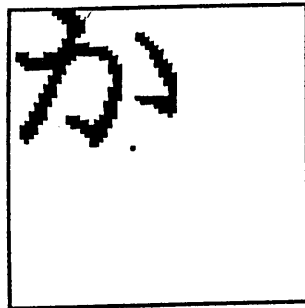
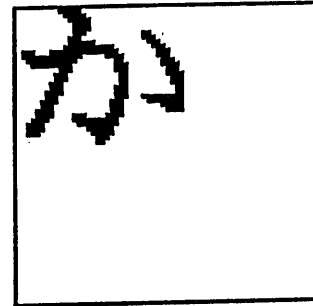


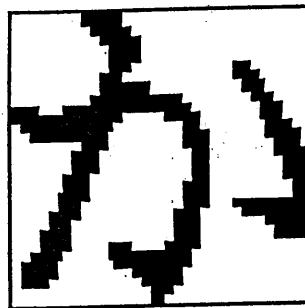
図 2.1 文字認識アルゴリズム



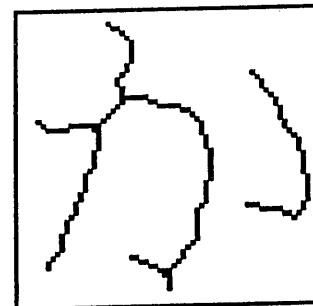
入力画像



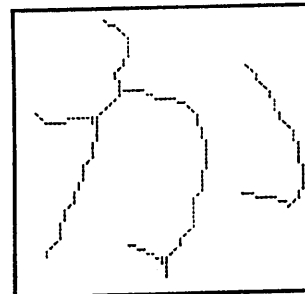
ノイズ除去・スムージング



正規化

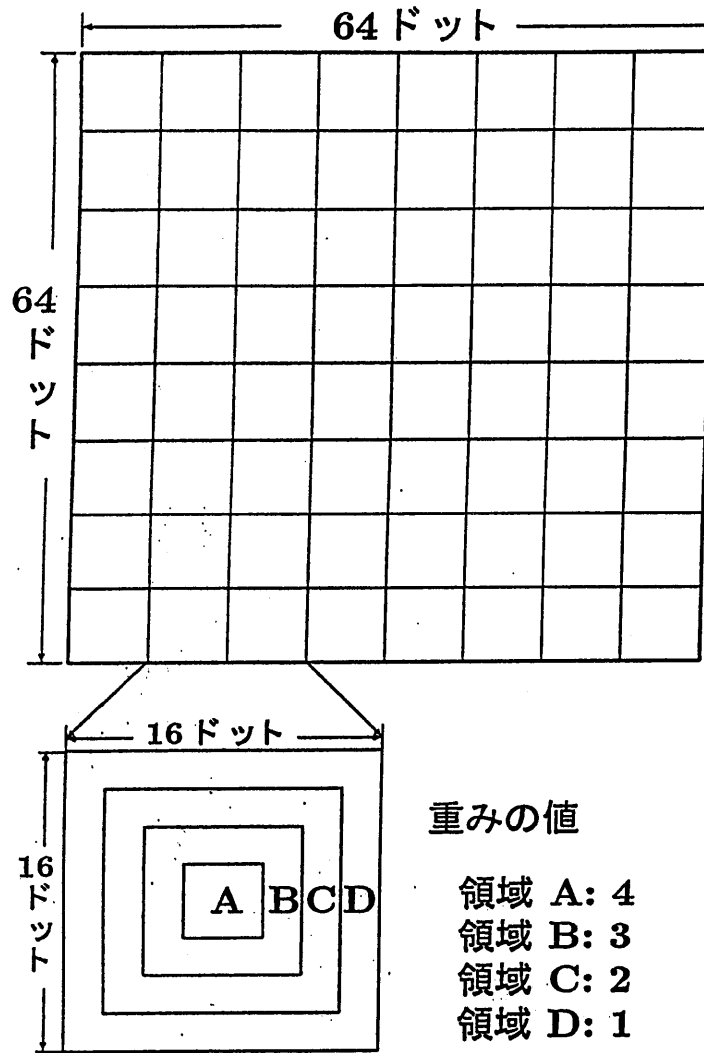


細線化



線素化

図 2.2 各処理後のイメージ



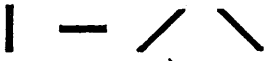
方向線素: 
方向線素特徴量: $V_m = (V_{m_1}, V_{m_2}, V_{m_3}, V_{m_4})$
where $m = 1 \sim 49$
小領域の数: 49(7×7)
方向線素特徴量の次元数: 196(4×49)

図 2.3 方向線素特徴量

第 3 章

変形文字の生成手法

3.1 手書き文字の変形に関する過去の研究

手書き文字については、

- 識別辞書作成に要する人手と時間
- OCR の識別性能評価

の二つの問題に対処するため、過去に変形文字の生成について研究が行なわれている。本研究の参考にした方法 [2] について、以下で説明する。

文字パターンの書かれている 2 次元平面 ($x-y$ 平面) をメッシュ状に分割し、文字パターンを各メッシュ上に存在する点の集合として表す。図 3.1 に示すように、変形文字 (図中灰色) は標準字形の細線パターン (図中黒色) 上の各点が摂動を受けて生成され则认为る。すなわち、標準字形上の各点に筆順に従って 1 から N までの番号を割り当て、 i 番目の点の座標を (X_i, Y_i) 、その点から r_i の摂動を受けて移った点を (X'_i, Y'_i) とすると、

$$\begin{aligned} X'_i &= X_i + r_i \cos \theta_i \\ Y'_i &= Y_i + r_i \sin \theta_i \quad (1 \leq i \leq N) \end{aligned} \quad (3.1)$$

となる。ここで θ_i は r_i が x 軸となす角度、 N は標準字形上に存在する点の総数である。摂動の方向としては、文字線と直角な方向、すなわち法線方向に限ることにする。式 (3.1) の r_i の生成方法としては、 r_i が前後の値 $r_{i\pm 1}$ 、 $r_{i\pm 2}$ 、 \dots と強い相関を有することを考え、次のような方法を用いる。すなわち、 $[-1, 1]$ の値を取る一様乱数 f_i を発生させ、それを図 3.2 に示すような離散的ガウス関数 g_k で畳み込む。従って r_i は

$$r_i = \sum_{k=-W/2L}^{k=W/2L} f_{i+k} \cdot g_k \quad (3.2)$$

となる。 L は x または y 方向に隣接するメッシュ間の距離を表す。ここでガウス関数 g_k は

$$g_k = A \cdot c \cdot \exp[-18k^2L^2/W^2] \quad (3.3)$$

ただし

$$c = \left(\sum_{j=-W/2L}^{j=W/2L} \exp[-36j^2L^2/W^2] \right)^{-1/2} \quad (3.4)$$

式(3.4)は f_i の分散値と r_i の分散値を等しくするための正規化係数である。式(3.2)はガウス関数によって定まる Low pass filter により一様乱数を平滑化し前後との相関を持たせたものである。平滑化の度合を表すのが式(3.3)の W 、すなわちガウス関数の広がり、 W までの範囲を畳み込みの対象にしたことを示している。 W が大きい程 r_i は滑らかな変化を示す。一方、式(3.3)における A は摂動の振幅を表しており、 A が大きい程、標準字形からのずれは大きくなる。標準字形としては、細線パターンを筆記情報とともにあらかじめ用意しておく。筆記情報は筆記者がその文字を書く場合、どこからどこまでを一筆で書くか、ということを示すものである。例えば図 3.1 のパターン "5" は弧 P_2P_3 を含む $P_1P_2P_3$ を第 1 のストローク、 P_1P_4 を第 2 のストロークとする、二つのストロークより成っている。摂動はストローク毎に独立に与えるので、図 3.1 の P_1 点において切れが生ずることがあっても、 P_2 点において P_1P_2 と P_2P_3 にストロークが分離されることはない。なお、連続する二点間に隙間が生じた場合は補間を行なう。

以上のような方法で生成された変形文字を用いて、認識実験を行なった結果、実際の手書き文字には及ばないものの、それに近い結果が得られている [3]。

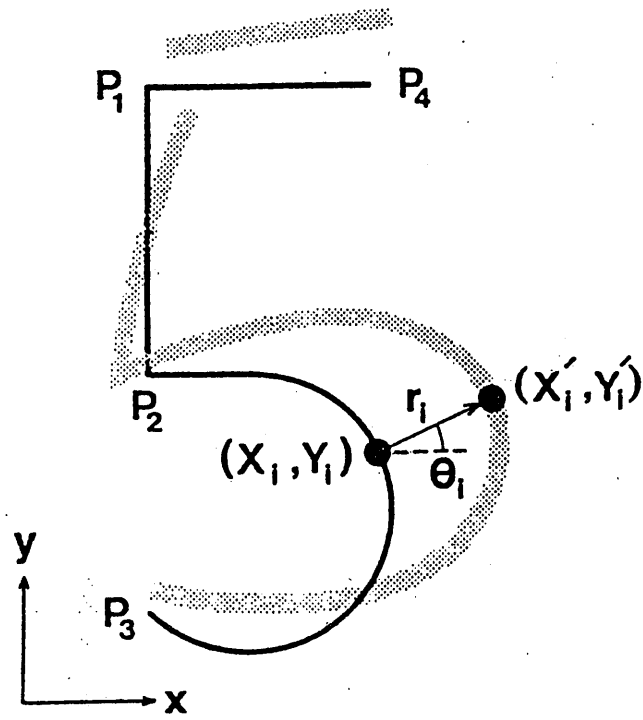


図 3.1 変形文字パターンの発生

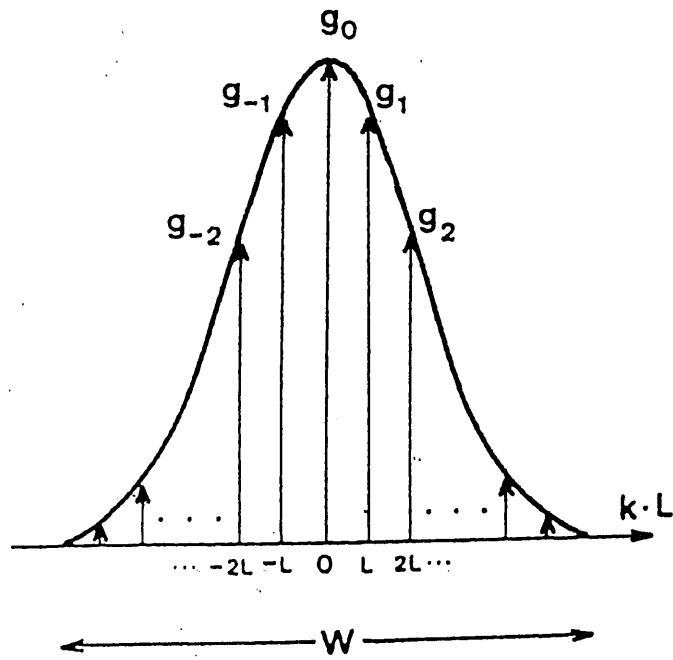


図 3.2 平滑化のためのガウス関数

3.2 本研究における変形文字の生成方法

本研究では印刷文字について考えるため、

- 印刷文字らしい変形を考える
- 筆記情報を得るのが困難

以上二つの理由により、筆記情報を考慮せずに変形文字を生成する手法を考えた。一つめの「印刷文字らしい変形」というのは、印刷文字の場合、手書き文字とは違って変形させたとき図 3.1 の「5」の $P_1P_2P_3$ と P_1P_4 のように二つの部分に分かれてしまうような変形があってはならない、ということである。具体的な変形方法としては、3.1 節で説明した手書き文字についての方法を、点の順番を考えているという意味で一次元的なものとする、細線パターン上の点と点との関係を二次元的に考える、というのが本研究の方法である。つまり、図 3.3 をある文字の細線パターンの一部とし、今、注目する点を真中の黒い点とすると、そこから二次元的に、距離が近い点の影響ほど強く受けるように振動させるわけである。

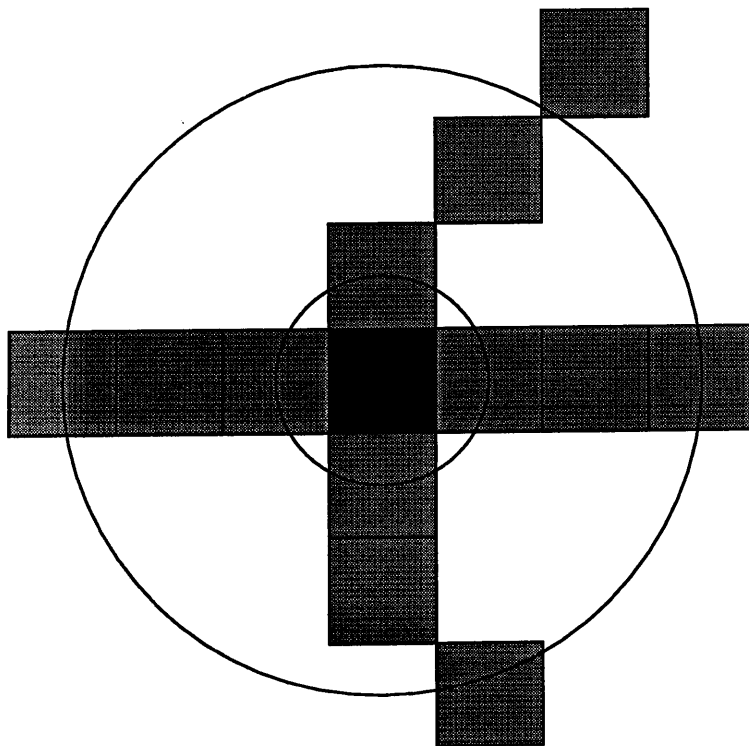


図 3.3 二次元的にみた細線パターン

これにより、摂動の大きさ r_i の式 (3.2)、ガウス関数 g_k の式 (3.3) をそれぞれ以下のように改善することにした。

$$r_{i,j} = \sum_k \sum_l f_{i+k,j+l} \cdot g_{k,l} \quad (3.5)$$

$$g_{k,l} = A \cdot c \cdot \exp[-18(k^2 + l^2)/W^2] \quad (3.6)$$

また、3.1 節で説明した方法の中に出てきた L (メッシュ間の距離) というパラメータは、本研究では考えないことにし、3.1 節での W/L を改めて W とおくことにした。これにより、摂動させる点との相関を考える範囲を、画素単位で表すことができるようになった。よって式 (3.5) における k, l の範囲は、 $(-W/2 \leq k, l \leq W/2)$ となる。

第 4 章

変形実験

4.1 実験結果

前章で説明した変形方法に従い、明朝体の平仮名「は」について、いくつかパラメータを変えて実際に変形を行なってみた結果を図 4.1 に示す。なお、 $A=0.0$ のときが標準細線パターンである。

4.2 考察

A を大きくすると標準字形からのずれが大きくなり、 W を大きくすると滑らかになることが実際にわかれると思う。 A を 2.0 に変えただけでもかなりの変形が見られ、 A が 8.0 のときには W を 64.0 まで上げて滑らかにはならないことがわかった。特に、もとの標準パターンにおいても近傍点間での変化の度合が比較的激しい部分は、 A を大きくすると、より一層変化の度合が激しくなっていると言える。また、 W が 4.0 以下では A が 2.0 のときでさえ、かなり不連続な変形が見られることから、文字点間の相関を考える場合かなり離れた点まで考慮しなければならないことがわかった。

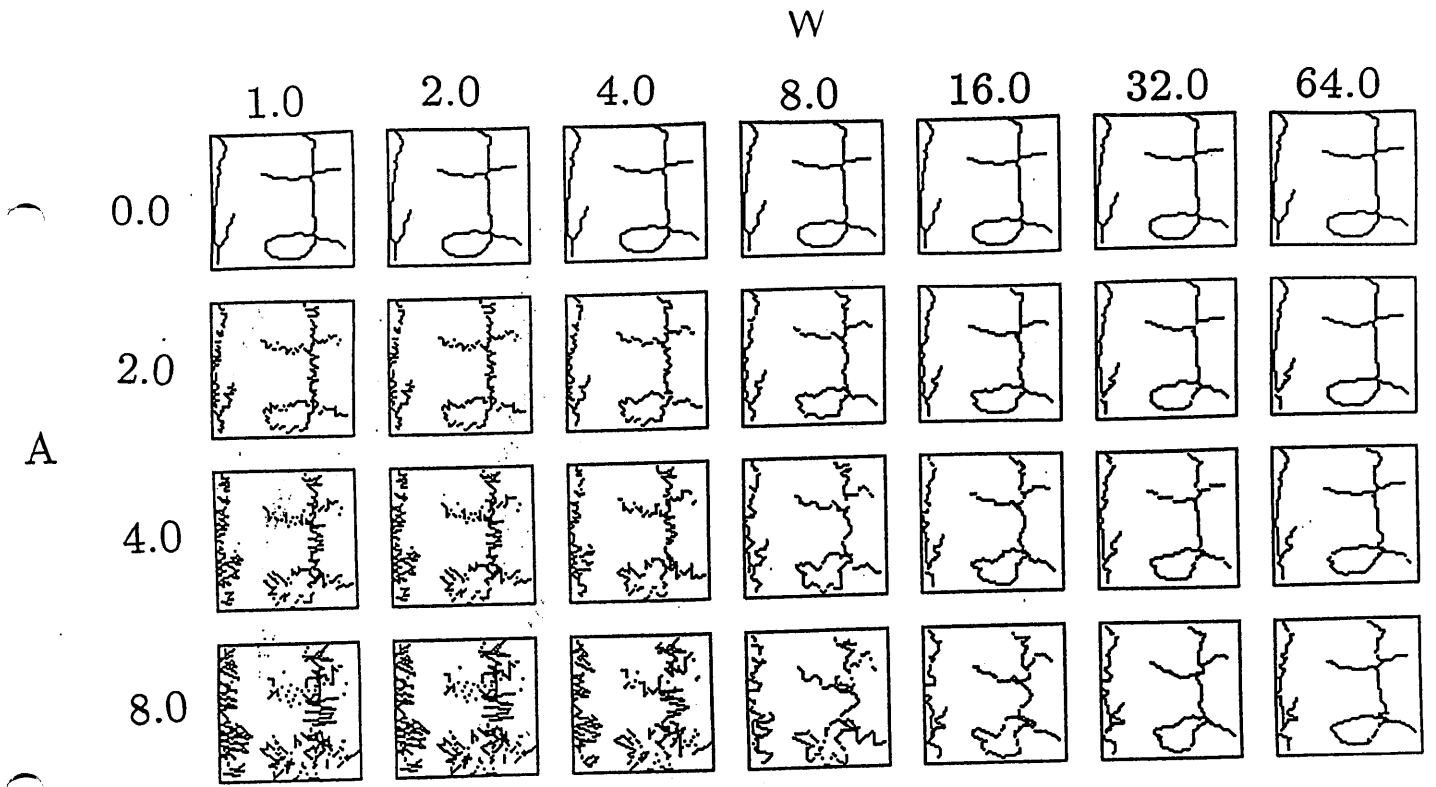


図 4.1 明朝体の平仮名「は」についての変形結果

第 5 章

認識実験

5.1 辞書の作成方法

3個の明朝体フォントから変形パターンをそれぞれ4個まで(計12個まで)生成し、それらを変形前の3個のフォントに加えていき辞書を作成する。

5.2 実験方法

前節で説明した方法により作成した辞書を用いて、フォント4個についてのオープン実験を行ない認識率を調べた。カテゴリ数は231(英数字、平仮名、片仮名)で、距離尺度にはユークリッド距離を用いた。なお、今回の実験で出した認識率は英字の「W」と「w」や、平仮名の「つ」と「っ」などの文字は同じとみなして出したものである。例えば、「つ」の第1位候補が「っ」になったとしても、それは正しく認識されたものとした。

5.3 実験結果

- 実験結果 1

辞書作成時に用いた変形パターン数を増やしていったときの認識率の変化を示したものが図 5.1 である。横軸は変形パターン数である。変形パターンは $A=4.0$ 、 $W=20.0$ に固定して生成したものであり、横軸が3のときが変形パターンを加える前の認識率である。なお、グラフ右下の説明書きは4個の未知フォントを示している。

- 実験結果 2

$W=20.0$ に固定して A を適当に変えていったときの認識率の変化を示したものが図 5.2 である。結果は変形パターン 1 2 個を加えて作成した辞書を用いたものであり、横軸が 0 のときが変形パターンを加える前の認識率である。

- 実験結果 3

$A=4.0$ に固定して W を適当に変えていったときの認識率の変化を示したものが図 5.3 である。辞書は実験 2 の要領で作成したものをを用いた。なお、グラフ右下の括弧内は変形パターンを加える前の認識率である。

実験結果 1 では一つの未知フォントを除けば同じような傾向の結果が得られた。辞書作成に用いたフォント数が 9 個のときの結果が最も良く、このとき変形パターンを加える前よりも 1、2 文字分認識率が上がっている。

実験結果 2 からは $A=4.0$ のときの結果が最も良く、このとき一つの未知フォントを除けば 1、2 文字分認識率が上がっていることがわかる。また、 A が大きくなるとともに認識率が下がる結果は見た目どおりの結果と言える。

最後に実験結果 3 では、未知フォントによって結果の傾向が異なったが、全体的に見ると $20 \leq W \leq 50$ の範囲では認識率に影響が出るほどの違いは見られなかった。

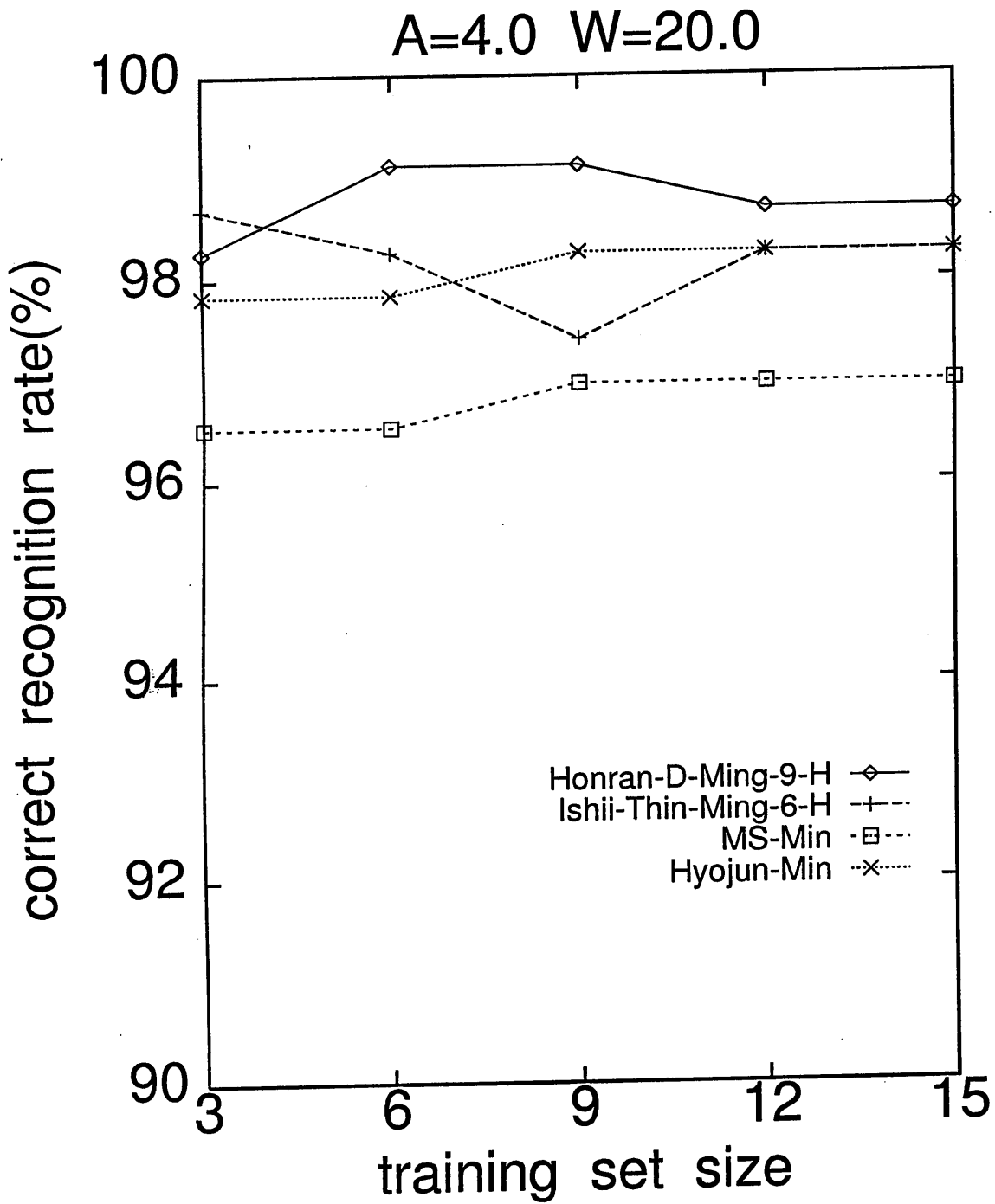


図 5.1 実験結果 1

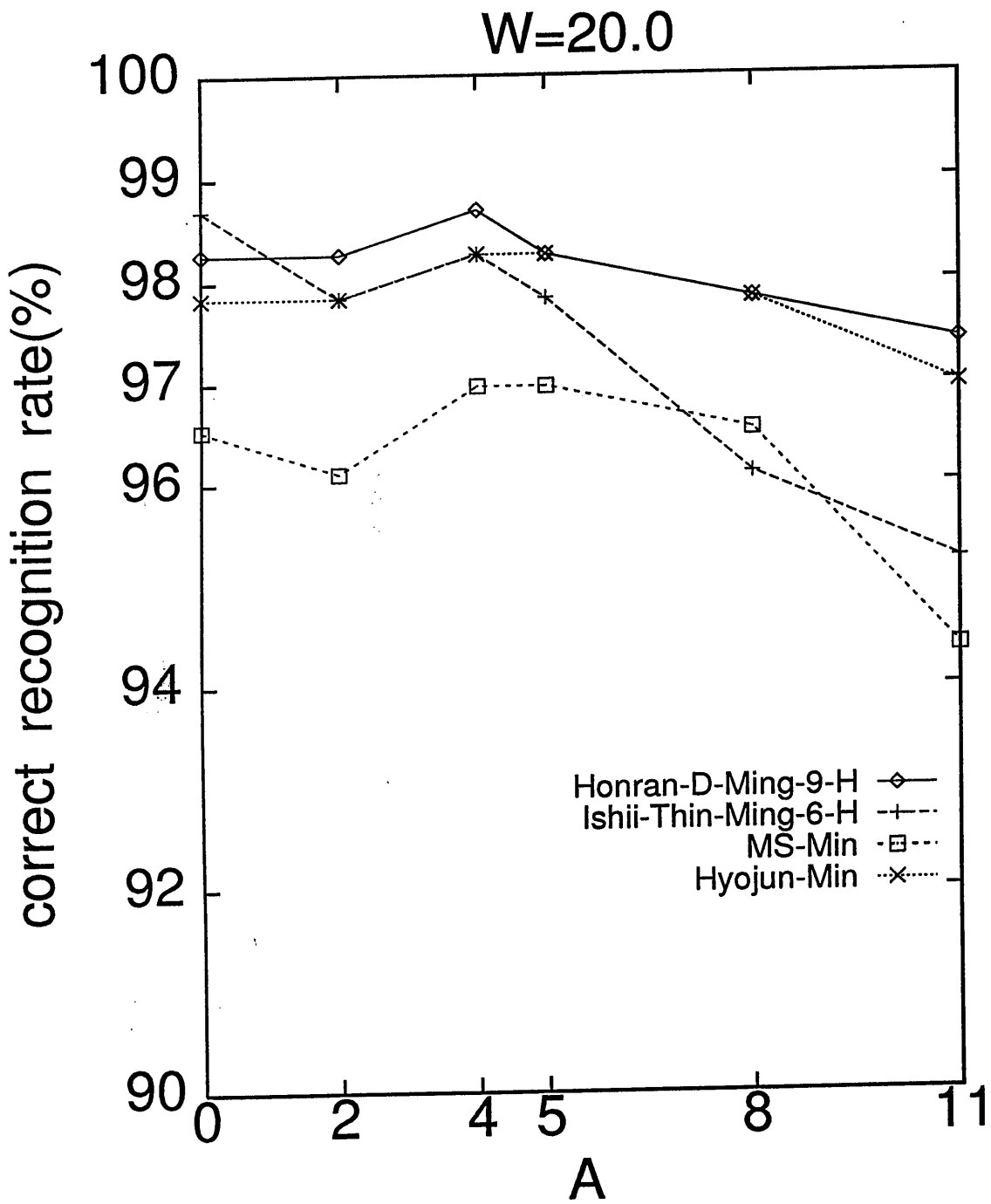


図 5.2 実験結果 2

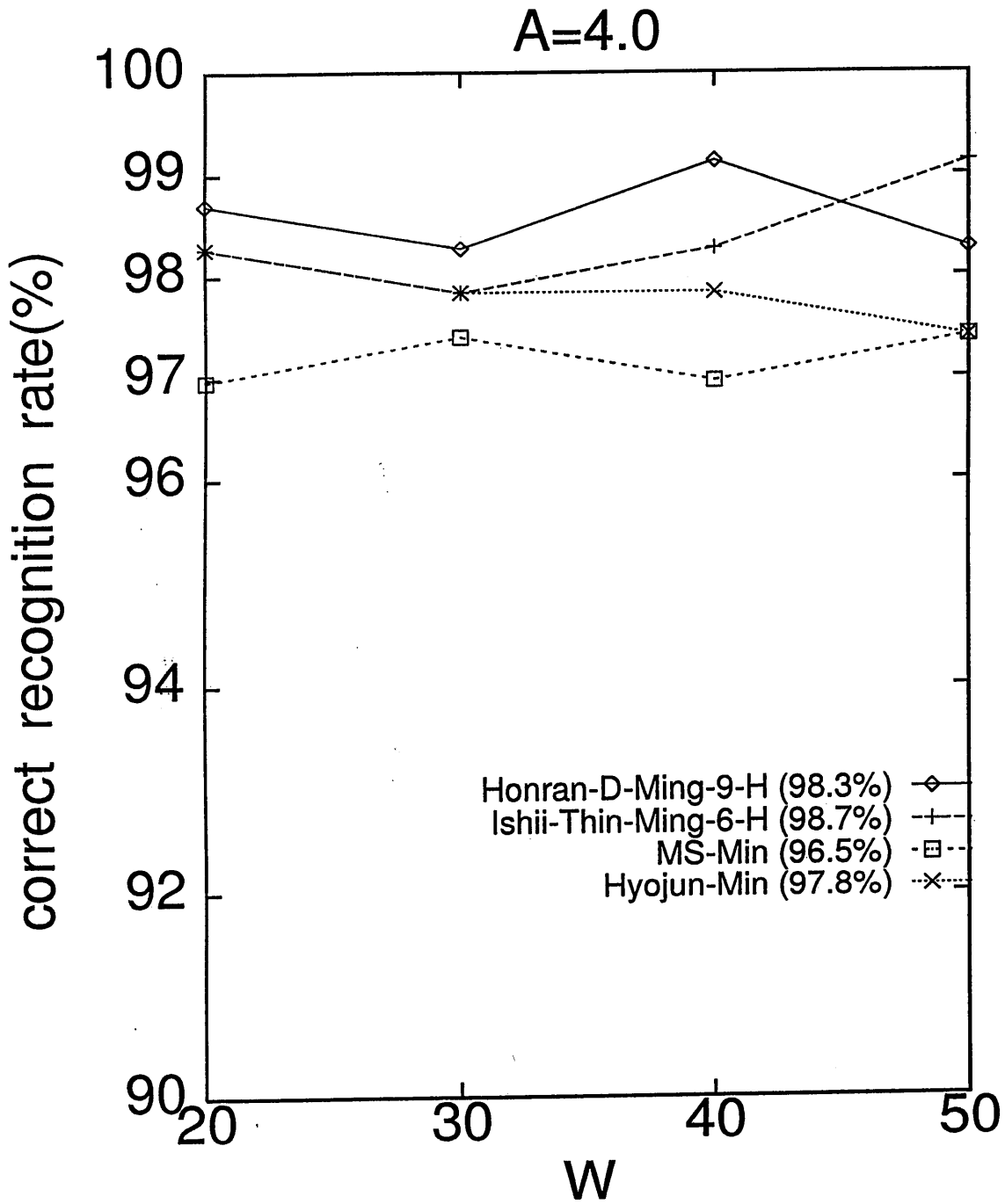


図 5.3 実験結果 3

第 6 章

まとめ

印刷文字について変形パターンを生成する方法を述べ、実際に変形させた文字を用いて基礎的な実験を行なった。実験からは、 A （摂動の振幅）の値が大きすぎると、 W （ガウス関数の広がり）の値を上げても変形パターンは滑らかにはならないことや、特定の変形パラメータ（ A 、 W ）においては、認識率が上がることがわかった。

また、実際の文字と変形文字との特徴量空間上での分布を比較することや、認識性能の向上につながる変形パラメータを見つけることが今後の課題として挙げられる。

謝辞

本研究を進めるにあたり、全般的な御指導を賜りました東北大学工学部阿曾弘具教授に心より感謝致します。

また、御討論、御協力をいただいた阿曾研究室の森 大毅氏、鈴木基之氏に心から感謝致します。

最後に多面に渡り御意見、御協力をいただきお世話になった阿曾研究室の皆様に感謝致します。

参考文献

- [1] 孫寧、田原透、阿曾弘具、木村正行：「方向線素特徴量を用いた高精度文字認識」
電子情報通信学会論文誌 (D-II) , Vol.J74-D-II, No.3, pp.330-339 (平成 3 年 3 月)
- [2] 石井健一郎：「変形文字パターン発生法とその応用」
電子情報通信学会論文誌、Vol.J66-D, No.11, pp.1270-1277 (1983 年 11 月)
- [3] 石井健一郎：「変形文字を用いた識別辞書の設計」
電子情報通信学会論文誌、Vol.J72-D-II, No.5, pp.669-677 (1989 年 5 月)

OHP 資料

文字パターンに関する基礎的研究

東北大学工学部通信工学科
阿曾研究室 4年 水庭 威幸

1 はじめに

- 背景
- 研究の目的

2 変形文字の生成手法

3 変形実験

4 認識実験

5 まとめ

6 今後の課題

1 はじめに

- 背景

文字認識高性能化の要求



文字パターンの分布を知る必要性

問題点： サンプル数が少ない

- 目的

印刷文字の変形パターンを大量に生成



文字認識性能の向上

2 変形文字の生成手法

- 手書き文字の変形に関する過去の研究(石井83)

筆記情報 : 一筆で書く範囲を示したもの

(X_i, Y_i) : i 番目の点の座標 ($1 \leq i \leq N$)

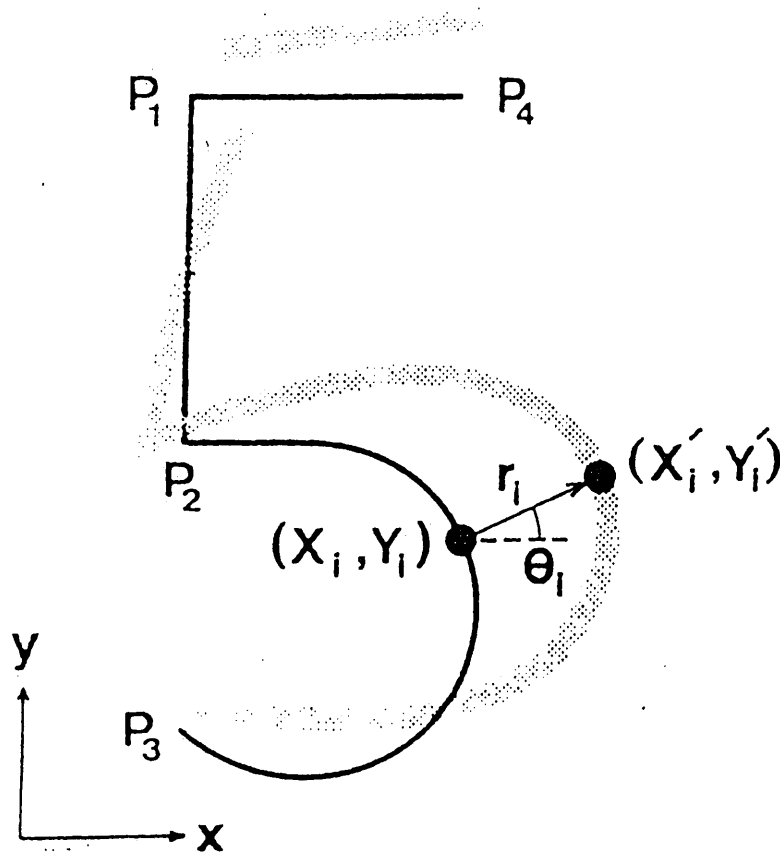
(X'_i, Y'_i) : i 番目の点が摂動を受けて移る座標

$$X'_i = X_i + r_i \cos \theta_i$$

$$Y'_i = Y_i + r_i \sin \theta_i$$

r_i : 摂動の大きさ

θ_i : 法線ベクトルが x 軸となす角度



$$r_i = \sum_{K=-W/2L}^{K=W/2L} f_{i+k} \cdot g_k$$

f_i : $[-1, 1]$ の値をとる一様乱数

g_k : 離散的ガウス関数

$$g_k = A \cdot c \cdot \exp[-18k^2 L^2 / W^2]$$

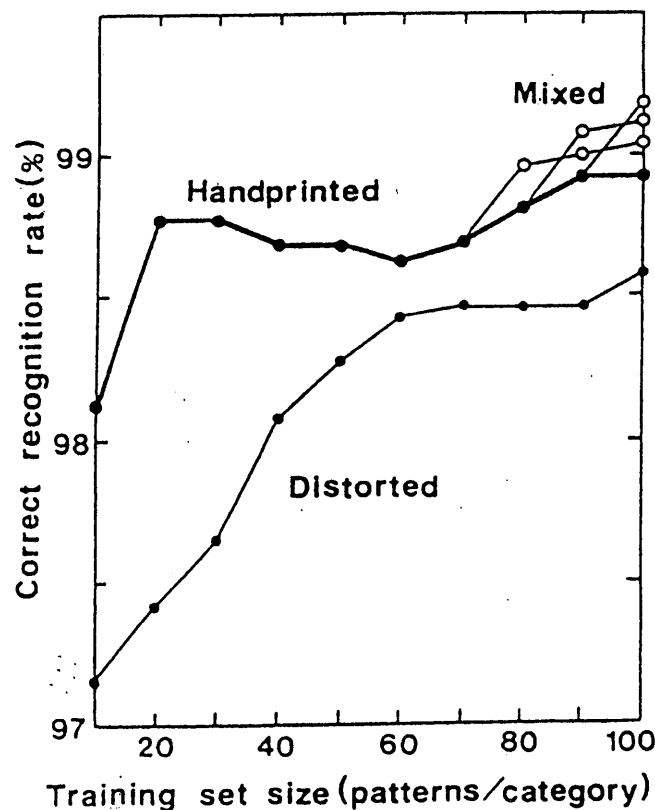
L : x または y 方向に隣接するメッシュ間の距離

A : 摂動の振幅

W : ガウス関数の広がり

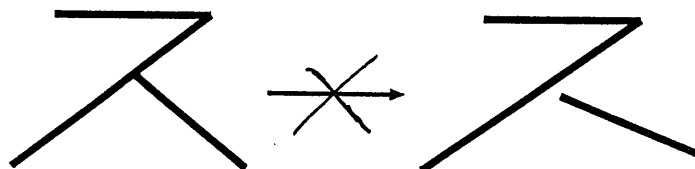
c : 正規化係数

$$c = \left(\sum_{j=-W/2L}^{j=W/2L} \exp[-36j^2 L^2 / W^2] \right)^{-1/2}$$



- 本手法

(1) 印刷文字らしい変形を考える



(2) 筆記情報を得るのが困難

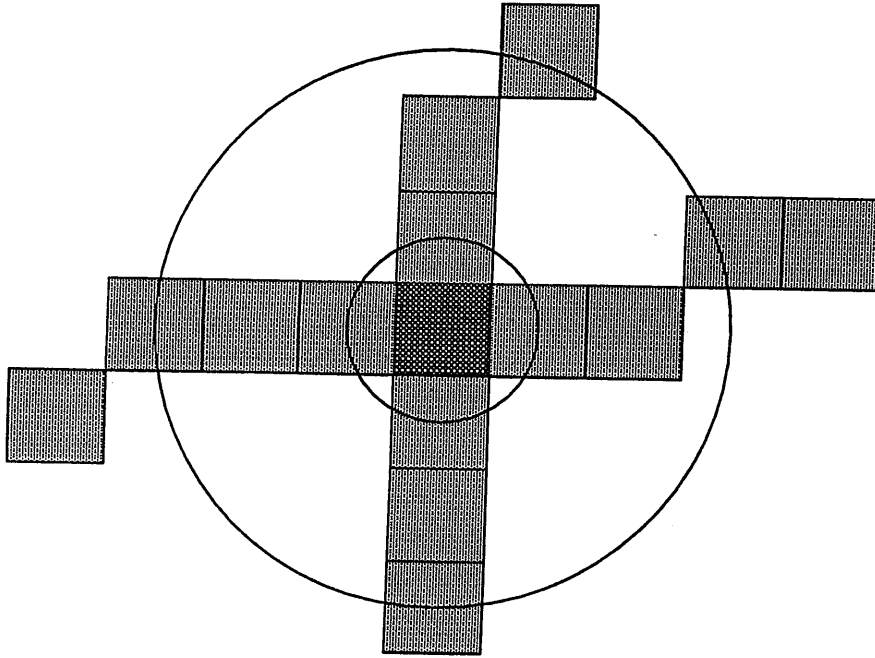
(1),(2) の理由



筆記情報を考慮せずに変形



細線パターン上の点と点との関係を2次的に考える



• 式の変更

$$r_i = \sum_{K=-W/2L}^{K=W/2L} f_{i+k} \cdot g_k$$

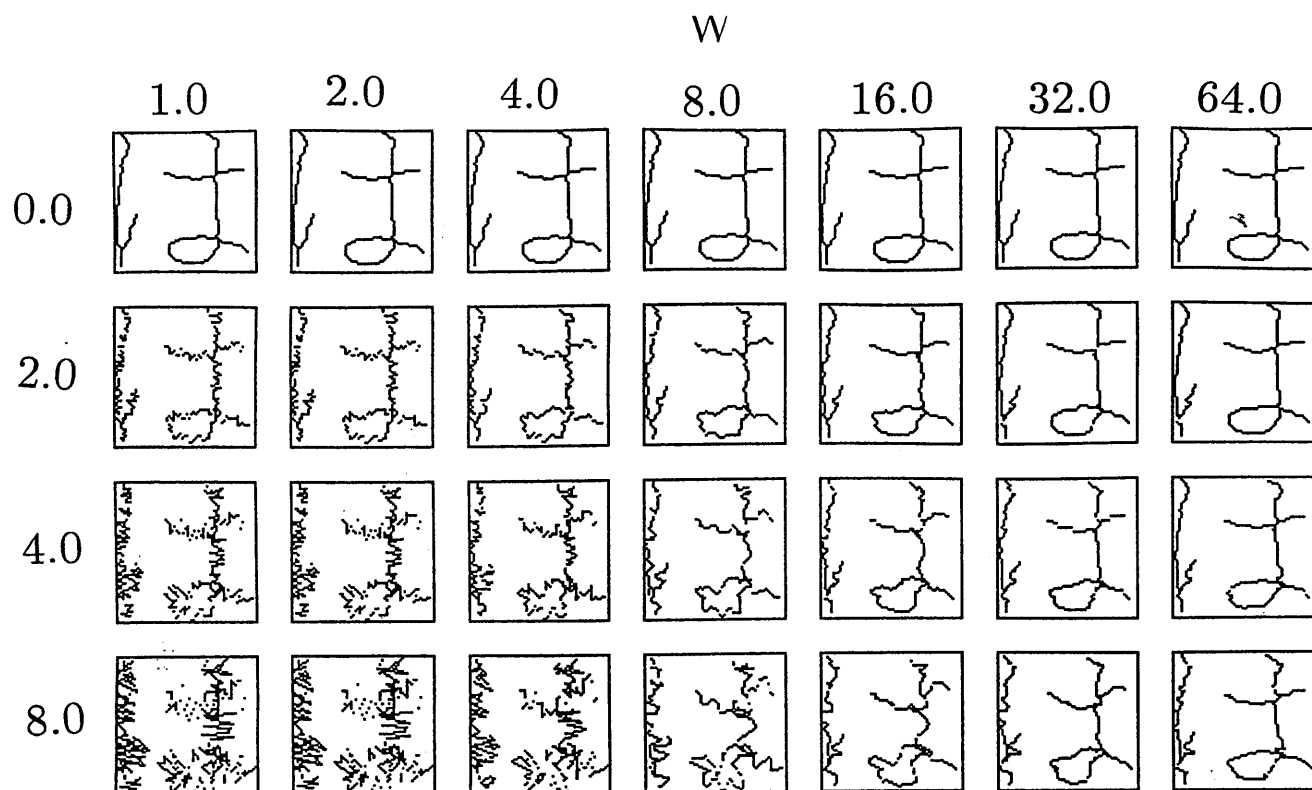
$$g_k = A \cdot c \cdot \exp[-18k^2 L^2 / W^2]$$

↓

$$r_{i,j} = \sum_k \sum_l f_{i+k,j+l} \cdot g_{k,l}$$

$$g_{k,l} = A \cdot c \cdot \exp[-18(k^2 + l^2) / W^2]$$

3 変形実験



A を大きくすると標準字形からのずれが大きくなる

W を大きくすると滑らかになる

4 認識実験

明朝体フォント 3 つから、変形パターンをそれぞれ 4 個まで (計 12 個まで) 生成



辞書作成



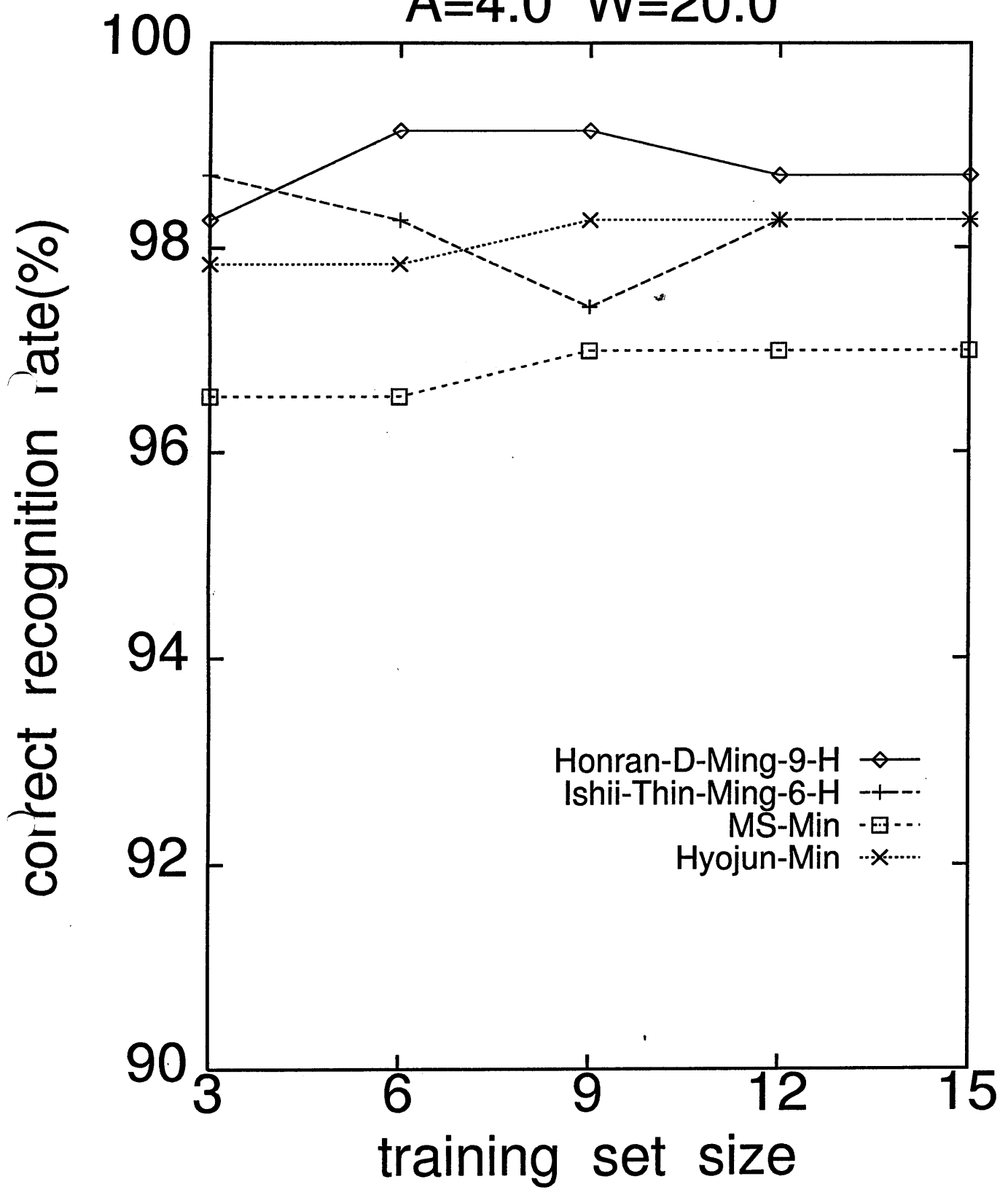
未知フォント 4 個について認識率を調べる

カテゴリー数 : 231(英数字、平仮名、片仮名)

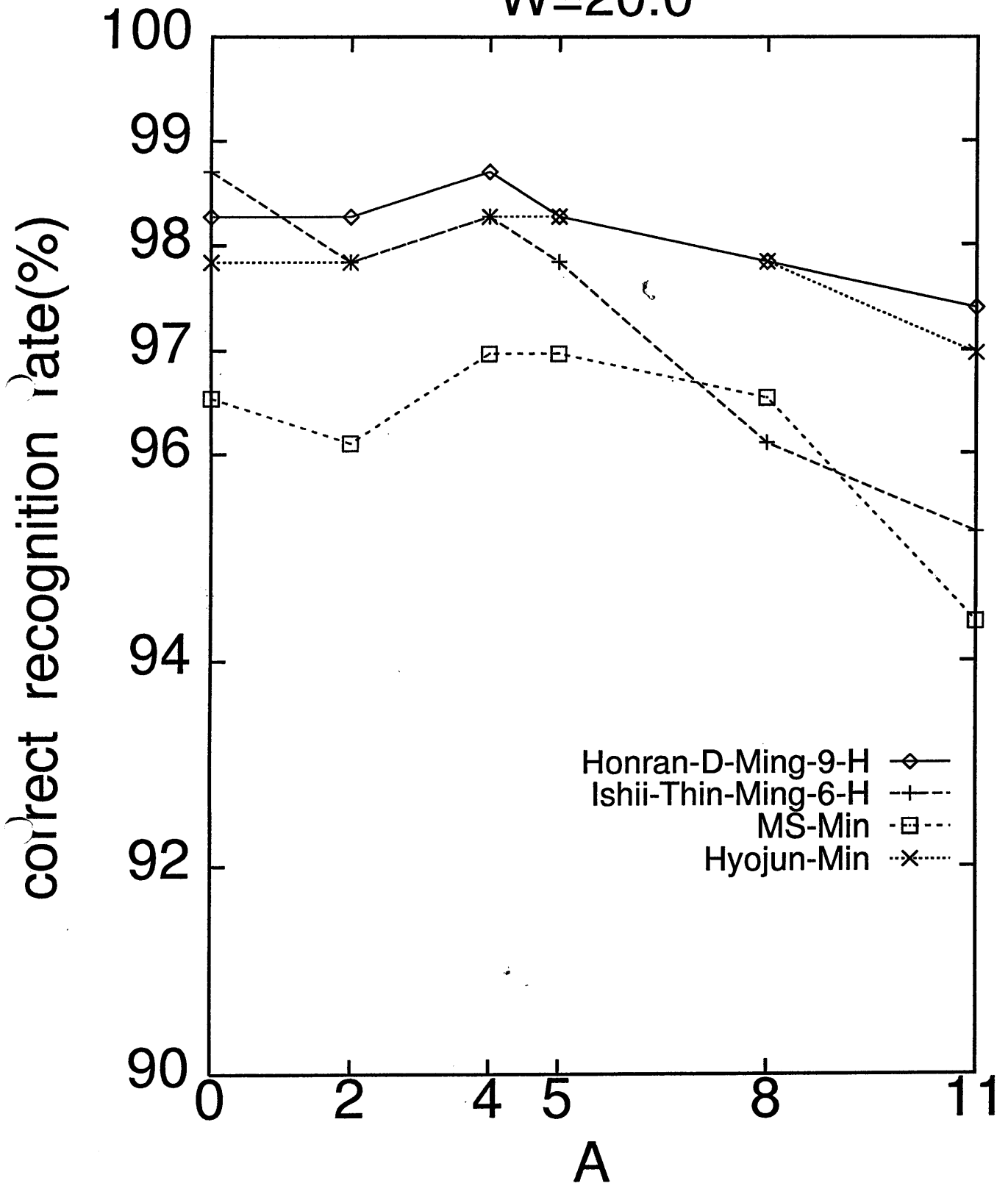
距離尺度 : ユークリッド距離

今回の実験で出した認識率は、英字の「W」と「w」や、平仮名の「つ」と「っ」などの文字は同じとみなして出したものである

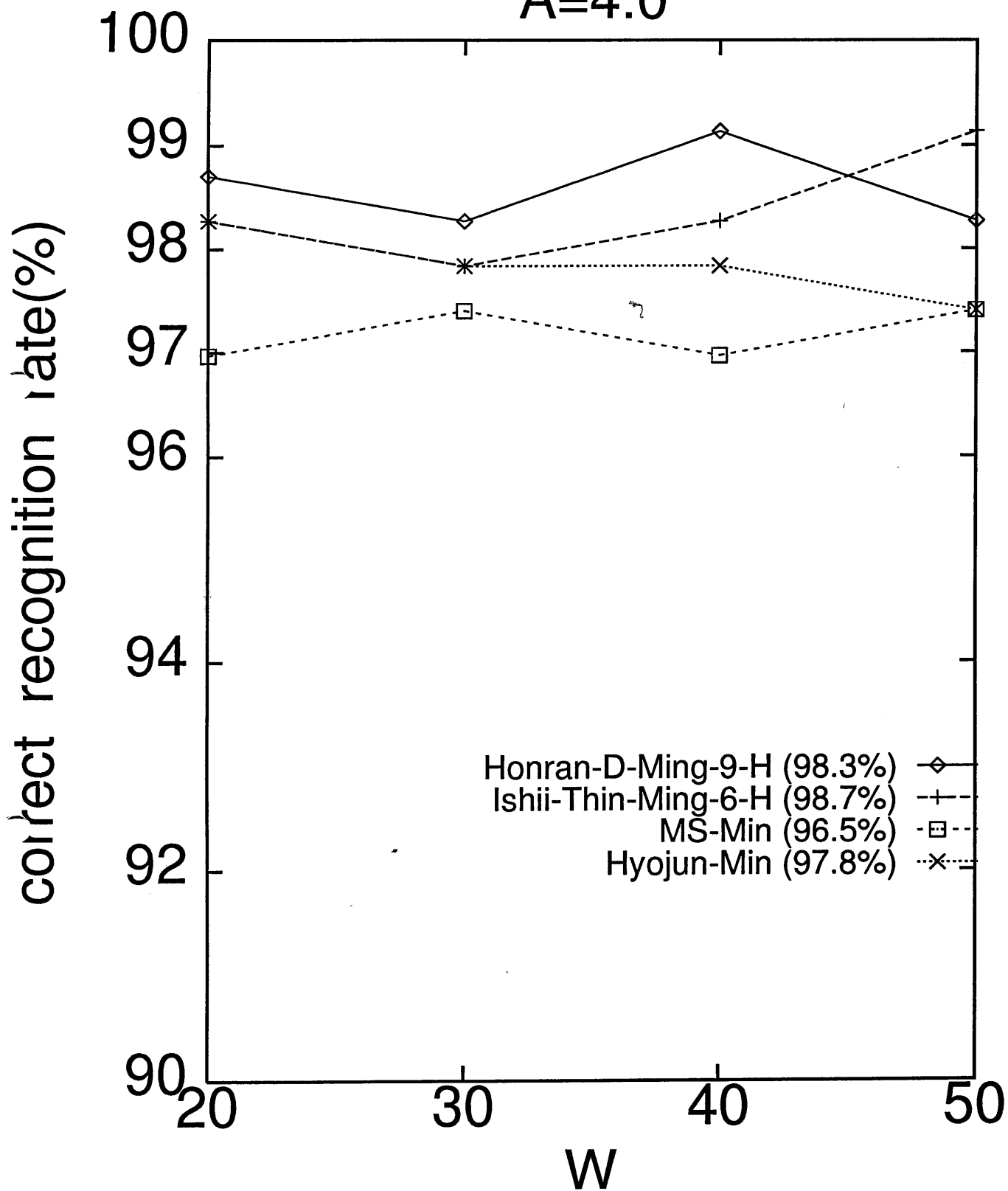
A=4.0 W=20.0



W=20.0



A=4.0



5 まとめ

- A の値が大きすぎると、 W の値を上げて
も滑らかにはならないことがわかった
- 特定の変形パラメータ (A, W) において
は認識率が上がることがわかった

6 今後の課題

- 実際の文字と変形文字との特徴量空間上
での分布の比較
- 認識性能の向上につながる変形パラメー
タを見つける