

Title	中間パターンを用いた二次元DPマッチングの高速化
Author(s)	野田, 陽
Citation	
Issue Date	2005-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/1903
Rights	
Description	Supervisor:党 建武, 情報科学研究科, 修士

修 士 論 文

中間パタンの選択的利用による二次元DPマッチン グの高速化

北陸先端科学技術大学院大学
情報科学研究科情報処理学専攻

野田 陽

2005 年 3 月

修 士 論 文

中間パタンの選択的利用による二次元DPマッチン グの高速化

指導教官 党 建武 教授

審査委員主査 党 建武 教授
審査委員 小谷 一孔 助教授
審査委員 赤木 正人 教授

北陸先端科学技術大学院大学
情報科学研究科情報処理学専攻

310082 野田 陽

提出年月: 2005 年 2 月

概要

本論文では、二次元 DP ワーピングを用いたパターン認識において、合成写像を利用することにより認識の高速化を行う手法を提案する。

画像パターン認識において、パターンに生じた変形を補償するための有効な手段として、二次元ワープ（弾性マッチング）の利用が検討されてきた。一般的に二次元ワープ問題は二次元 DP (2D-DP) を用いることで最適解が求まる。しかし、実用するには 2D-DP の計算量はあまりにも莫大である。

そこで本提案手法では、次の三段階を経て認識することを検討する。まず、入力に応じて特定のパターンを中間パターンとして選択する。次に、入力-中間パターン間写像と既に計算済みの中間-参照パターン間写像を用いて入力-参照写像を合成写像として得る。最後に、合成写像を利用して入力-参照パターン間距離を求める。この手法により、認識時に 2D-DP の計算を行う量を $\frac{\text{中間パターン数}}{\text{参照パターン数}}$ に減らす事ができ、高速化が可能となる。

また手書き文字認識実験において、30 クラス 90 参照パターン数のひらがな認識実験を行った。結果、従来の DP マッチング法と比較し認識率を落とすことなく、計算量を従来手法から 90% 近く削減した。

目次

第1章 序論	1
1.1 研究の背景と目的	1
1.2 本論文の構成	1
第2章 二次元DP	2
2.1 二次元ワープ決定問題	2
2.2 単調連続二次元ワープ	3
2.3 ピクセルワイズ二次元DP	3
2.4 写像制限パラメータ	3
2.4.1 一様性ペナルティ	4
2.4.2 単調性ペナルティ	4
2.4.3 絶対座標ペナルティ	5
2.4.4 縦方向ぼかしの適用	6
第3章 中間パターンを用いた合成写像	7
3.1 合成写像による高速化の原理	7
3.1.1 既存の認識手法	7
3.1.2 合成写像を用いた二次元DP	8
3.2 合成写像の作成法	9
3.3 二次元DPの写像制限パラメータと中間パターン	11
3.3.1 直接写像時	11
3.3.2 写像制限に関する考察	13
3.3.3 パスマックス	15
3.4 中間パタンの選択法	15
3.4.1 ミスマッチ度	15
3.4.2 入力-中間パターン間距離とミスマッチ度々	15
3.4.3 パスマックス数とミスマッチ度の関係	17
3.5 パスマックスと計算時間	17
3.6 中間パタンの認識時選択指針	20
3.7 中間パタンの生成法	20
3.8 合成写像時のパターン間距離定義	21
3.9 本手法の概略	21

3.9.1	前処理	21
3.9.2	認識時	21
第4章	文字認識による性能評価	22
4.1	実験条件	22
4.1.1	画像データの前処理	22
4.1.2	認識率実験	23
4.2	パスマックス数・写像制限・コスト関数に関する実験	23
4.3	計算量に関する考察	25
4.3.1	ビーム幅と計算量の関係	26
4.4	中間パターン選択精度と認識率の関係	28
第5章	結論	29
5.1	本研究の結果	29
5.2	今後の課題	29

目 次

2.1	二次元ワープ例	2
2.2	単調連続性条件化での写像範囲	3
2.3	連続単調制限下で単調性ペナルティが必要なケース	5
3.1	直接写像二次元DPと合成写像二次元DP	8
3.2	二次元DPでの写像例	9
3.3	合成写像例	10
3.4	写像の補完	10
3.5	冗長パス	11
3.6	「あ」から「あ」への写像結果	11
3.7	「あ」から「か」への写像結果	12
3.8	「あ」から「か」への写像結果 (写像制限なし)	13
3.9	「あ」から「あ」への写像	13
3.10	写像結果	14
3.11	「あ」→「か」→「あ」の写像	14
3.12	ぼかした画像の例	16
3.13	入力-中間パターン間距離 vs ミスマッチ度	16
3.14	パスミックス数とミスマッチ度の関係	18
3.15	ビーム幅とミスマッチ度の関係	19
4.1	各ぼかし度合いの一例	27

表 目 次

2.1	各ペナルティの重み	4
4.1	認識率実験結果	24
4.2	計算量 (秒)	26
4.3	N-Best 中間パターンに正解パターンが含まれる確率	27
4.4	極端にぼかして中間パターンを選択した場合	27

第1章 序論

1.1 研究の背景と目的

画像パターン認識において、パターンに生じた変形を補償するための有効な手段として、二次元ワープ（弾性マッチング）の利用が検討されている。一般的に二次元ワープ問題は二次元DP(2D-DP)を用いることで最適解が求まる。

しかし、2D-DPは総当たり法と比較すれば計算量は大きく削減されているが、対象となるパターンのサイズ、DPパス探索の計算量が $O(m^n)$ で増加するという問題がある。そこで、実時間内で非線形伸縮を行うために、ビームサーチ [1] や階層化 [2] によって準最適解を近似的に求める手法が提案されている。

これらの手法は2D-DPそのものの計算量を低減させるものだが、本論文では、2D-DPに基づくパターン認識法の計算量を低減する手法を提案する。

従来の認識法では、入力・参照パターン間の二次元DPを行い、そのときのDPコストでパターン間距離を定義することで、パターン認識を行う。入力パターンが与えられる毎に参照パターンとの二次元ワープを求める必要がある。そこで、本研究ではある参照パターン数よりもはるかに少ない数の中間パターンを設定し、入力-中間の写像を二次元DPで求めて、その入力-中間写像とすでに計算済みの中間-参照パターン間写像の合成を入力-中間パターンの近似値として利用することを考える。これにより、認識時に二次元DPを行う回数を著しく減らし、認識速度を向上させることが可能である。

1.2 本論文の構成

本論文では、二章では二次元DPの概念とアルゴリズムについて述べる。三章では合成写像を用いた高速化の概念・合成写像のアルゴリズムとその特性について述べる。四章では提案手法を用いての手書き文字認識の実験結果をしめし、それに関する考察を行う。五章では本稿全体の考察を行い、今後の課題を述べる。

第2章 二次元DP

2.1 二次元ワープ決定問題

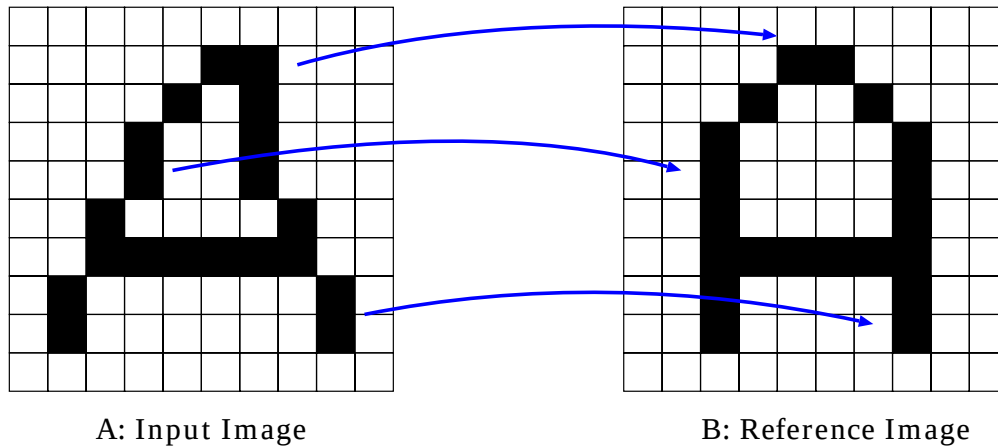


図 2.1: 二次元ワープ例

図 2.1 のように歪んだパターン同士の変形を補償するためにパターンをどのように変形させて、マッチさせれば良いかを決定する問題を二次元ワープ決定問題と言う。このように変形を補償することで、非線形に歪んだパターン同士のパターンマッチングにおいて、よりよい精度での認識ができる。

具体的に図 2.1 の入力画像を

$$A = a(i, j) | i = 1, 2, \dots, I, j = 1, 2, \dots, J \quad (2.1)$$

参照画像を

$$B = b(i, j) | i = 1, 2, \dots, I, j = 1, 2, \dots, J \quad (2.2)$$

とし、変形を補償するための写像関数を f としたとき。

$$D = \sum_{j=1}^N \sum_{i=1}^N \|a(i, j) - b(f(i, j))\| \quad (2.3)$$

目的関数 D を式 2.3 とし、これを最小にする写像関数 f を求める事で二次元ワープ決定問題を解く。理想的には、 A の画像を写像関数 f で変形することにより、 B と全く同じ画像になる。

2.2 単調連続二次元ワープ

不自然な写像を避けるために、写像関数に単調性と連続性の制限をかけた物を担当連続二次元ワープと呼ぶ。単調性とは、写像前後で隣接するピクセルの左右、上下関係が保たれる事をしめす。これにより、画像が反転するような写像を防ぐことができる。連続性条件とは写像前に隣接していたピクセルがお互いに極端に離れた場所に写像されることを防ぐ制限である。具体的には式 2.4 で示されるように、変形先の座標 $f(i, j)$ に制限を加える。

$$\begin{aligned} f(i, j) - f(i-1, j) &= \{(x, y) | x = 0, 1, 2, y = any\} \\ f(i, j) - f(i, j-1) &= \{(x, y) | x = any, y = 0, 1, 2\} \end{aligned} \quad (2.4)$$

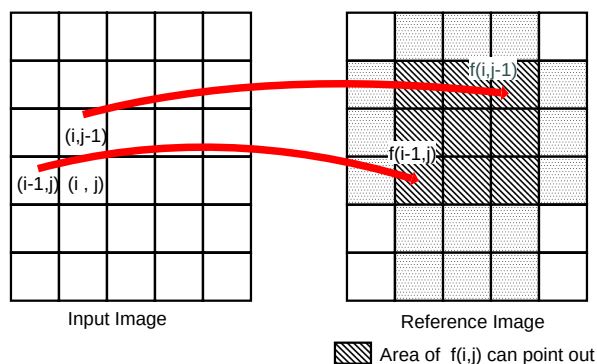


図 2.2: 単調連続性条件化での写像範囲

この二つの条件を満たす写像の例を図 2.2 に示す。

写像する前に隣接していた $(i-1, j)$, $(i, j-1)$ の写像先 $f(i-1, j)$, $f(i, j-1)$ によって、 $f(i, j)$ が写像できる範囲が制限されている。

2.3 ピクセルワイズ二次元 DP

入力画像 A をラスタスキャンしながらその順に従いながら (i, j) での写像 $f(i, j)$ を決定するという操作をコスト関数を D とした DP (Dynamic Programing: 動的計画法) 探索として解くものをピクセルワイズ二次元 DP と呼ぶ。

2.4 写像制限パラメータ

実際にピクセルワイズ二次元 DP を解く場合、たとえば $a(i, j) = 0$ のような真っ黒な画像同士の写像を決定する場合、どのような写像にしても D の値は常に 0 となりどのよう

α	一様性ペナルティの重み
β	単調性ペナルティの重み
γ	絶対座標ペナルティの重み

表 2.1: 各ペナルティの重み

な写像であっても最適解となってしまう。そこで式 2.5 のように写像形状に対するペナルティ P_{ij} を導入する。

$$D(A, B) = \sum_{j=1}^N \sum_{i=1}^N d(a(i, j), b(f(i, j))) + P_{ij} \quad (2.5)$$

これは (i, j) までラスタスキャンしてきて決定された写像 f がどのようなものであるかに対するペナルティであり、具体的には以下に示す、一様性・単調性・絶対座標ペナルティである。また本論文では、このペナルティ付きピクセルワイズ二次元DPの事を単に二次元DPと呼ぶ。

なお本論文では、各ペナルティの重み係数をそれぞれ表 2.1 のように α, β, γ と呼ぶ。

2.4.1 一様性ペナルティ

一様性ペナルティは平行移動ではペナルティがかからないが、画像を変形させてしまうような写像にはペナルティをかけるようなペナルティである。具体的には、式 2.6 の様に写像関数の一次微分の総和となる。

$$\begin{aligned} Pu_{ij} = & \|f(i, j) - f(i-1, j) - (1, 0)\| \\ & + \|f(i, j) - f(i, j-1) - (0, 1)\| \\ & + \|f(i, j) - f(i-1, j-1) - (1, 1)\| \end{aligned} \quad (2.6)$$

ただし、写像元画像と写像先画像のサイズが同じ場合である。サイズが違う場合は適宜座標変換が入る。

2.4.2 単調性ペナルティ

連続単調制限は、 (i, j) からの写像先と $(i-1, j), (i, j-1)$ からの写像先の関係しか規定していないため、図 2.3 のように単調性が部分的に崩れる場合がある。

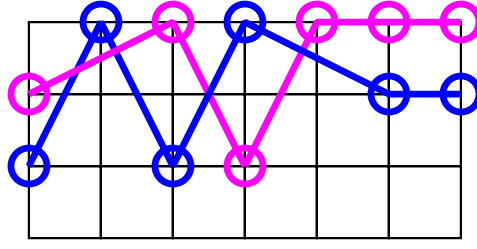


図 2.3: 連続単調制限下で単調性ペナルティが必要なケース

そのためのペナルティとして 2.7 式で示されるペナルティを導入する。

$$\begin{aligned}
 Pf_{ij} = & \kappa(Y_d \cdot X_l) \\
 & + \kappa(X_r \cdot Y_d) \\
 & + \kappa(Y_u \cdot X_r) \\
 & + \kappa(X_l \cdot Y_u)
 \end{aligned} \tag{2.7}$$

ただし

$$\begin{aligned}
 Y_u &= f(i, j-1) - f(i, j) \\
 Y_d &= f(i-1, j) - f(i-1, j-1) \\
 X_r &= f(i-1, j) - f(i, j) \\
 X_l &= f(i, j-1) - f(i-1, j-1)
 \end{aligned} \tag{2.8}$$

$$\kappa(x) = \begin{cases} x & (x \geq 0) \\ 0 & (x < 0) \end{cases}$$

2.4.3 絶対座標ペナルティ

写像関数によってどれだけ動いたかに関するペナルティで、式 2.9 で示されるように、写像もと座礁 (i, j) と写像先座標 $f(i, j)$ のノルムで決定される。

$$Pa_{ij} = \|(i, j) - f(i, j)\| \tag{2.9}$$

2.4.4 縦方向ぼかしの適用

二次元DPは横方向へのラスタスキャンを行う。そのため、ビーム探索を行うと横方向への探索は良く行われるが、縦方向への探索は十分になされない場合が多い。この状況を改善するために、縦方向に画像をぼかした成分をわずかに加えた。無論ぼかしすぎると情報量が落ちてしまい、認識率が落ちてしまうので、最適なパラメータを求める必要がある。今回実際に使ったパラメータに関しては4.1.1を参照。

第3章 中間パターンを用いた合成写像

3.1 合成写像による高速化の原理

3.1.1 既存の認識手法

1. すべての参照パターンに対して入力-参照パターン間で二次元 DP を行う。
2. 二次元 DP でえられたパターン間距離が最も近い物を解とする

前述のように、二次元 DP は計算量が多いので、ここでの主な計算コストは二次元 DP をおこなう部分である。以後この認識手法を、直接写像を用いた二次元 DP と呼ぶ。

3.1.2 合成写像を用いた二次元DP

これを解決するために、一次元のDPであるDTWにおいては森本ら [3] によって合成写像を用いた手法が提案されている。

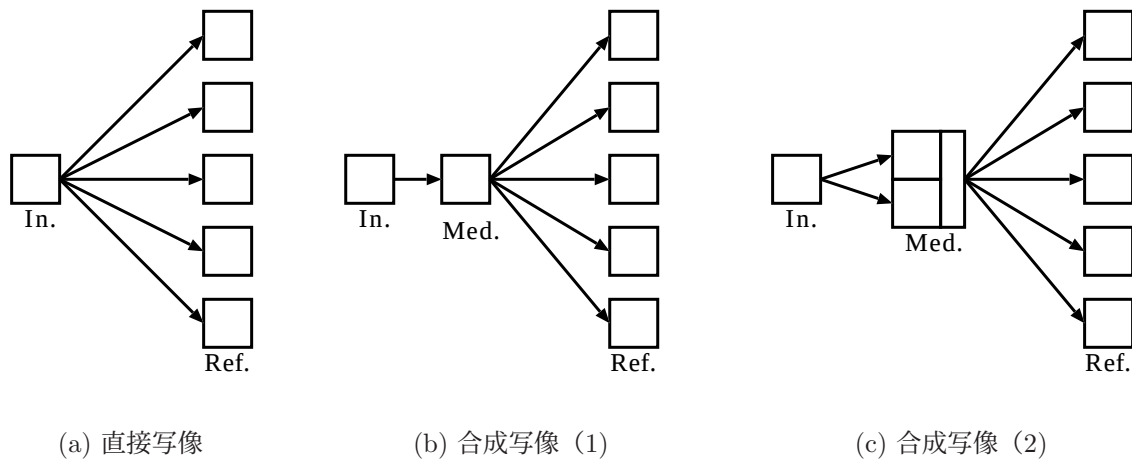


図 3.1: 直接写像二次元DPと合成写像二次元DP

1 個の中間パターンを用いた場合

まず、図 3.1(b) のように中間パターンを一つ用いた場合について示す。

1. 入力-中間パターン間で二次元DPを行う
2. 入力-中間パターン間で得られた写像と、予め計算済みの中間-参照パターンの合成写像を計算する
3. 合成された写像を入力-参照パターン間写像の近似値として利用し、パターン間距離を定義する
4. パターン間距離が最も近いものを解とする

このように、1 個の中間パターンを用いると二次元DPの計算を 1 回しかせずに済む。それにより計算量を劇的に削減できる。しかし、実際には 1 個の中間パターンでは認識精度が落ちてしまう事が一次元でのDPにおいて知られている。

複数の中間パターンを用いた場合

そこで、図 3.1(c) のように、複数の中間パターンを用いて信頼性を上げる方法を検討する。

1. 小数の中間パターンを選ぶ。
2. 入力-中間パターン間で二次元DPを行う
3. 入力-中間パターン間距離で得られた写像と、予め計算済みの中間-参照パターンの合成写像を計算する
4. 合成された複数の写像を重ね合わせる。
5. 重ね合わされた写像を全探索空間とみなして、二次元DPを再び行う
6. 5で得られたパターン間距離が最も短いものを解とする

このようにすると、合成後にまた二次元DPを行っているので非常に遅いように見える。しかし、重ね合わされた写像によって作られる探索空間は通常の二次元DPで探索する空間にくらべて非常に狭いので、ビーム探索時のビーム幅を非常に小さくできる。したがって合成後に行う二次元DPは、非常に高速に実行可能である。

本研究ではこのような、複数の中間パターンを用いた合成写像法を用いて二次元DPによるパターン認識の高速化を行う。

3.2 合成写像の作成法

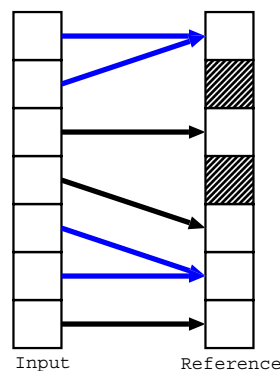


図 3.2: 二次元DPでの写像例

ピクセルワイズ法において作成された写像をの例を図 3.2 に示す。ここでは、簡単のため、1x7 ピクセルの画像の例を示す。

図 3.2 に示される様に入力画像側の各ピクセルに写像が来ているが、参照画像側の全ピクセルに写像が来ているわけではない。

ここで、中間パタンを用いて合成写像を作ると図 3.3 のように合成を行うと、参照パタンにおいて写像されないピクセルが増える。これはパタン間距離を定義する際に参照されないピクセルが増えることを意味する。

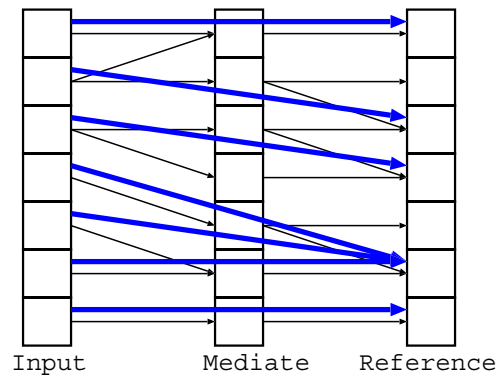


図 3.3: 合成写像例

このような状況を回避するために、冗長パスを利用する。冗長パスは、参照パターン中の写像が来ないピクセルを図 3.4 のように補間してから、合成写像を作成することによって作られた図 3.5 のような冗長な写像を持つ写像関数である。

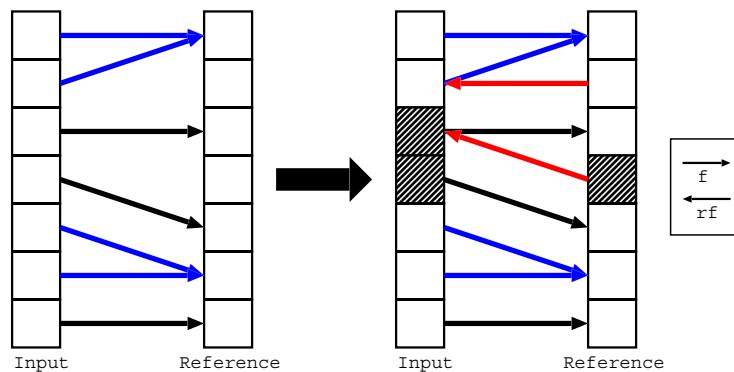


図 3.4: 写像の補完

このようにして得られた冗長パスを、写像可能な範囲を示す写像制限と見なし、二次元 DP を行うことで、合成後のパスを得る。この際通常の二次元 DP に比べ探索空間が極端に小さくなっている。そのため非常に小さなビーム幅のビーム探索で二次元 DP を実行できるため、高速に合成が可能である。

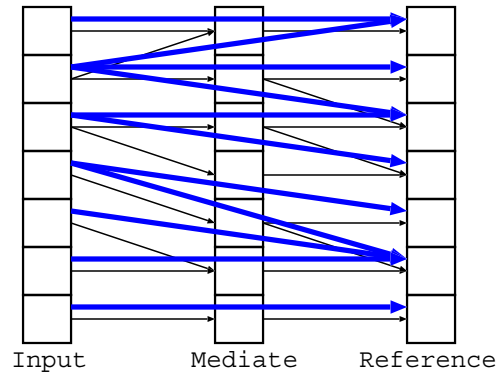


図 3.5: 冗長パス

3.3 二次元DPの写像制限パラメータと中間パターン

そのパラメータにおける写像結果の一例をここに示す。

上記に合成写像を作成する方法について述べたが、中間パタンの作成法や写像制限ペナルティについては触れなかった。ここでは中間パターンを作成する指針として、二次元DPで作られる写像に関して考察する。

3.3.1 直接写像時

まず、実験的にパラメータを変更しながら付録 4.1.2 認識率実験を行い、86.7%の認識を達成できるパラメータ ($\alpha = 10, \beta = 5, \gamma = 0$) を求めた。このときの、写像の一例を図 3.6, 3.7 に示す。



(a) 入力



(b) 逆写像後



(c) 参照

図 3.6: 「あ」から「あ」への写像結果

もし、二次元DPによる写像がきれいに白は白、黒は黒にマッチしているならば、図 3.6 のように、入力から参照への写像 f の逆写像を書けた画像 $f^{-1}(B)$ は入力画像に非常に似た画像になる。

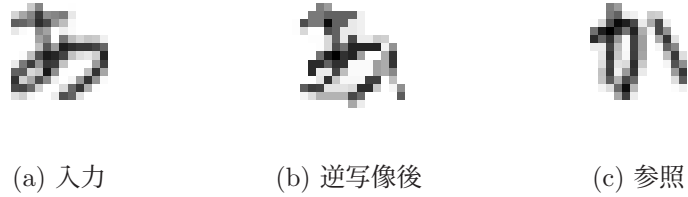


図 3.7: 「あ」から「か」への写像結果

しかし、「あ」から「か」の変形のように参照画像が入力画像を大きく変形しないと変形マッチでない場合。図 3.7 のように写像にミスマッチがおきる。直接二次元DPを行って認識する場合、このようなミスマッチは単に二次元DPでのコストが高くなる要因になるだけと考られる。これは、極端に形が違うクラス外パターンを排除するという意味では有効である。また、変形範囲を制限することにより、探索範囲を制限する事ができる。これは、ビーム幅を極端に大きくしなくても認識が可能な理由の一つと考えられる。

しかし、合成写像時に入力-中間、中間-参照パターン間でこのようなミスマッチが起きると入力-参照間写像が正しく推定されなくなり、パターン間距離を算出する際の誤差の原因となり、正解パターンとのパターン間距離が不当に遠くなったり、その逆がおきてしまう。

これは、単純に一つの間中パターンを利用して合成写像を作る場合、単純に直接写像時と同じパラメータを用いると、入力-中間パターン距離が入力-正解パターンの距離に匹敵するほど近い必要がある事を示す。もし、入力-中間パターンを入力-正解パターン間距離ほどに近づける事ができれば、その時点で中間パターンが正解である可能性が高くナンセンスである。したがって、多少離れた中間パターンにミスマッチのない写像を作れる様にする必要がある。

3.3.2 写像制限に関する考察

そこで、写像制限を全くかけない二次元DPで先ほどの「あ」から「か」への写像の実験を行ってみた。

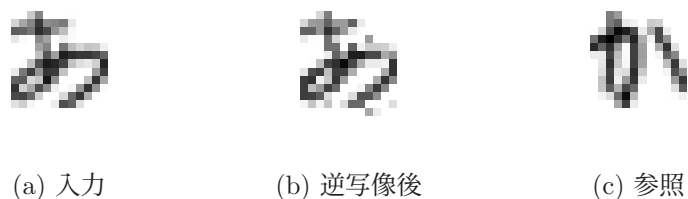
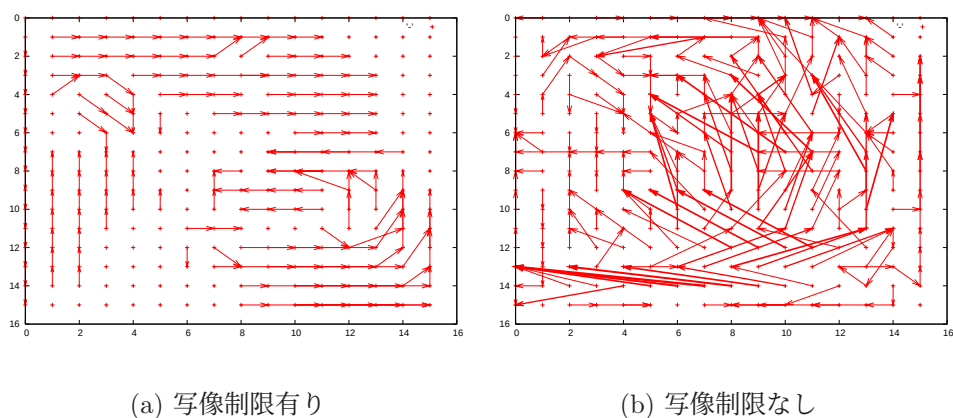


図 3.8: 「あ」から「か」への写像結果 (写像制限なし)

このように、ミスマッチが少ない写像が得られた。しかし、写像制限が弱くなるにつれて図 3.9、写像時の変形は不規則で変形量も大きくなる傾向がある。



矢印の元が写像元座標 (i, j) 、矢印の先が写像先座標 $f(i, j)$ を示す

図 3.9: 「あ」から「あ」への写像

そこで、変形量を小さく保ちながら、ミスマッチすくない画像を作れるパラメータ ($\alpha = 1, \beta = 1, \gamma = 5$) を「あ」を中間パターン「か」とおして別の文字の「あ」に写像する実験的で求めてみた。図 3.10, 3.11

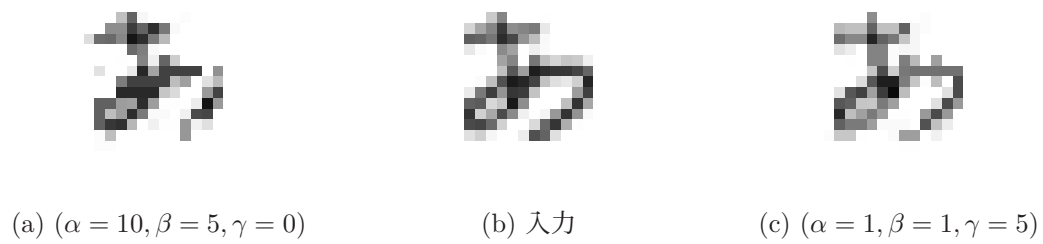


図 3.10: 写像結果

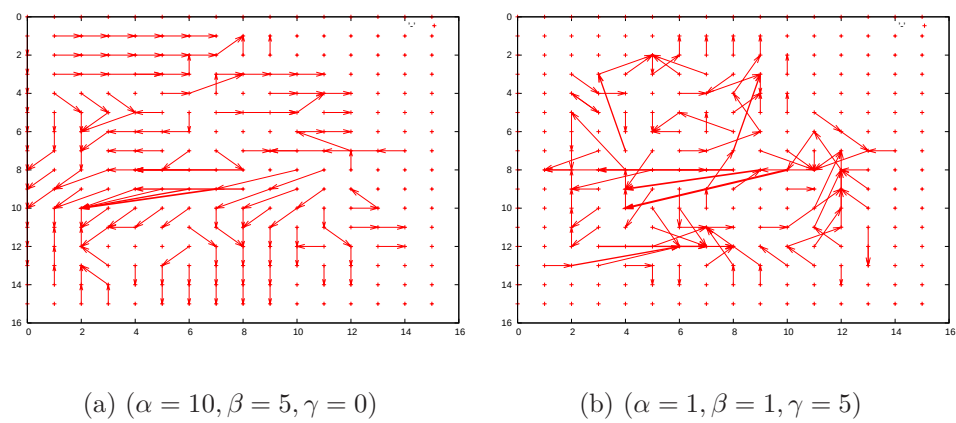


図 3.11: 「あ」 → 「か」 → 「あ」 の写像

3.3.3 パスマックス

写像制限を制御して、ミスマッチのある写像が作られるのを防ぐ方法のほかに、合成時にミスマッチのある写像を修正する方法が考えられる。これは3.1.2で述べたように中間パターンを複数用いる事で実現される。具体的には、まず複数の中間パターンで得られた複数の冗長パスを用意する。その複数の冗長パスを重ね合わせさらに冗長なパスを得る。最後に、この重ね合わせた冗長パスを全探索範囲とみなして、入力-参照パターン間の二次元DPを行う。この際、探索範囲は冗長パスで作られた探索範囲だけなので、通常の二次元DPにくらべてきわめて狭い探索範囲となる。したがって、狭いビーム幅で探索すれば十分であり。

このようにして特定の間中間パターンに起因する写像のミスマッチを補正する事ができる。

このように複数の中間パターンから得られた冗長パスを混合する事を、本論文では単にパスマックスと呼ぶ。

3.4 中間パタンの選択法

本手法では、多くの中間パターンから入力に適した中間パターンを適切に高速に選択することによって認識率の維持を図る。そこで、どのような基準で複数の中間パターンを選べばよいか？また、いくつ選べばよいかを検討する。

3.4.1 ミスマッチ度

どの程度ミスマッチを起こしているかの指標として、ここでミスマッチ度を導入する。ミスマッチ度は中間パターン M を入力-中間パターン間写像 f_{im} の逆写像で写像した $f_{im}^{-1}(M)$ と入力パターン A のノルム $\|A - f_{im}^{-1}(M)\|$ が100を超えたピクセル数を示すものとする。なお、今回の画像は256階調で、すべての画像は0から255までのフルレンジを使っている。

3.4.2 入力-中間パターン間距離とミスマッチ度々

たとえば、入力-中間パターンが全く同一のものであれば、合成したことに起因するミスマッチは全くない。予想として、入力-中間パターンが近ければ、同様にミスマッチが少ないことが予想される。そこで、入力-中間パターンが類似していれば入力パターンに適した中間パターンで有ると考え、入力-中間パターン間類似度基準で中間パターンを選択することを提案する。無論、認識時に入力-中間パターンの類似度を求める必要があるので、高速に求められる類似度が必要となる。そこで、本手法では計算が高速で簡便な手法として、画像同士のノルムを取り入力-中間パターンの類似度として用いる事を検討する。



(a) 原画像

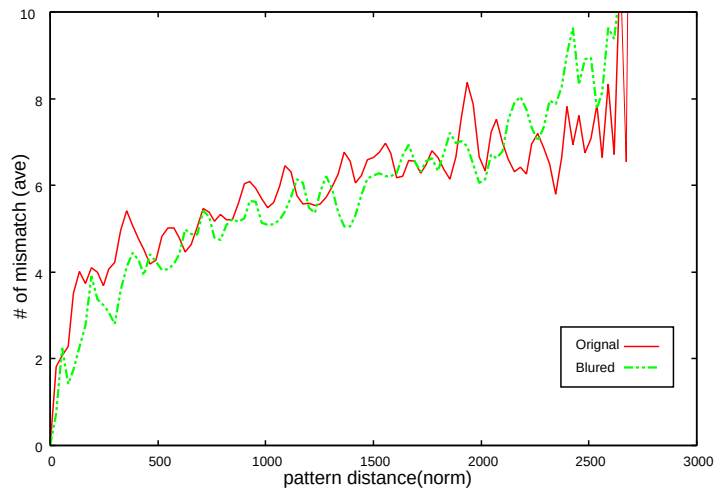


(b) ぼかしたもの

図 3.12: ぼかした画像の例

今回は類似度として、画像間のノルムを用いた物をと、図 3.12 のように、多少ぼかした後に画像間ノルムを求めた物を図 3.14 に示した。

なお、このときの合成時探索ビーム幅は 20、合成時の写像制限ペナルティは入力-中間、中間-参照パターン作成時のペナルティと同じでそれぞれ $(\alpha = 10, \beta = 5, \gamma = 0)$, $(\alpha = 1, \beta = 1, \gamma = 5)$ である。また、中間パターン選択時にぼかしの有無はあるが、合成写像を作成し、ミスマッチ度を求める際にはぼかし処理はいずれもかかっていない。



縦軸：各参照パターンへのミスマッチ度の平均 横軸：入力-中間パターン間ノルム
なお、参照パターンは入力と同クラスのもの

図 3.13: 入力-中間パターン間距離 vs ミスマッチ度

図 3.13 を見ると、ぼかしの有無にかかわらず入力-中間パターン間ノルムが小さければ、ミスマッチ度も小さい傾向が見て取れる。このこと、入力-中間パターン距離の類似度として画像間のノルムを用いて中間パターンを選択することがある程度有用であることがわかった。

また図 3.13 のぼかした場合とぼかしていない場合の違いについて注目すると。実際に中間パターンとして選出する、入力-中間パターン間距離が小さい領域において、ぼかした画像を使って選択した中間パターンを使った場合のほうがミスマッチ度が小さくなっているこのことにより、ぼかすことによって高速中間パターン選択時にミスマッチの少ない中間パターンを効率的に選択できる事がわかった。

これは、パタンの変形によって二次元DP的には類似していても、単純にノルムを取っただけのコストは高くなることがある。その変形分をぼかすことによってカバーしている考えられる。

3.4.3 パスミックス数とミスマッチ度の関係

3.3.3 でパスミックスによって写像のミスマッチ度が低減できると述べた。それを検証するために、中間パターンを類似度基準で選択し、パスミックスを行った。

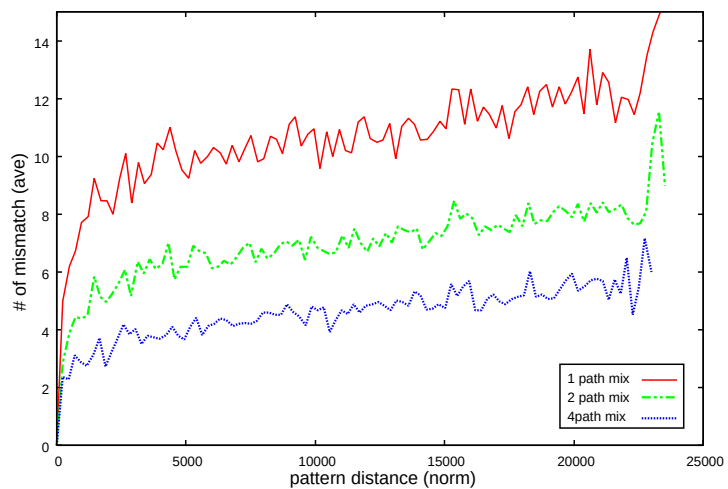
図 3.14 を見るとパスミックス数 1、すなわちパスミックスを行わない場合に比べてパスミックスを行うほうがミスマッチ度はさがっている。このことから、パスミックスに一定の効果があるという結果が得られた。また、図 3.14(b) の 2path mix 時と図 3.14(a) の 4path mix 時のマッチ度がほぼ同じであることが見とれる。このことから、写像制限が比較的きつえばあいでもパスミックス数を増やすことによってマッチ度を改善できる可能性があることが見て取れる。

3.5 パスミックスと計算時間

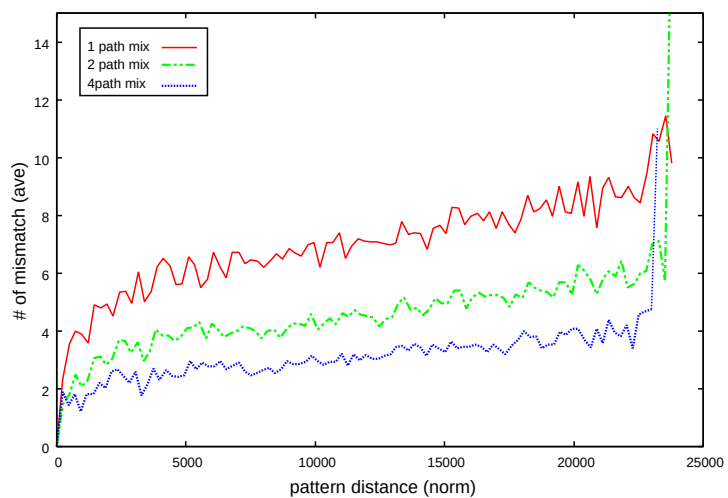
合成時の計算時間は主に合成時のビーム幅によって規程される。しかし、ビーム幅を狭くすると合成することに起因するミスマッチ度が上昇することが予想される。そこで、ミスマッチ度がビーム幅によってどのように変わるかを調べ、本手法の設計の指標にすることにした。これによって、計算時間とミスマッチ度のトレードオフを見ることが出来る。また、パスミックス数によってその関係が変わること予想される。したがって、パスミックスと、ビーム幅と、ミスマッチ度の関係をしらべた。

図 3.15 は、その実験結果である。各グラフの条件はパスミックス数 1~6 個の場合と、直接写像の場合である。各条件において、入力パターン数 270 に対して、同クラスの参照パターン 3 個ずつへの写像 270×3 個を生成しその際のミスマッチ度の平均を出している。

図 3.15 をみるとどちらの写像制限パラメータの場合もパスミックス数 4 程度、ビーム幅 10 程度のところで直接写像時のビーム幅 2000 に相当するミスマッチ度になっている。これは、この二つのパラメータにおいてはパスミックス数 4 程度、ビーム幅 10 程度で直接写像のビーム幅 2000 に匹敵するミスマッチ度の写像を合成できていること示している。これは、3.3.2 で示した写像制限なしの時のように無駄な写像が起きていない限り、直接写像じとほぼ同じ認識率が期待できることをしめしている。



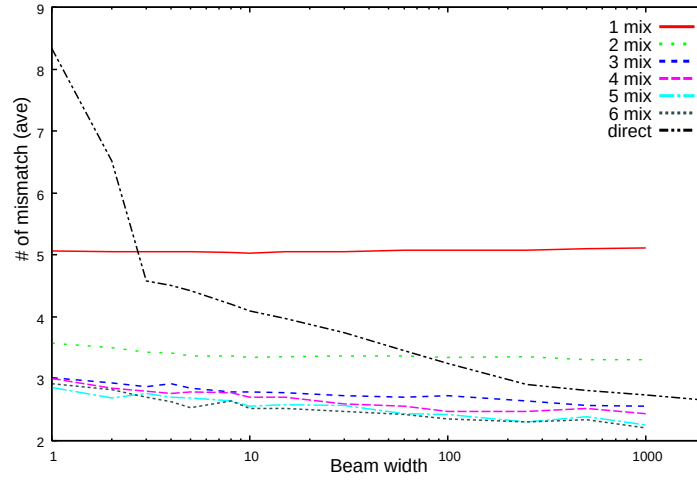
(a) ($\alpha = 10, \beta = 5, \gamma = 0$)



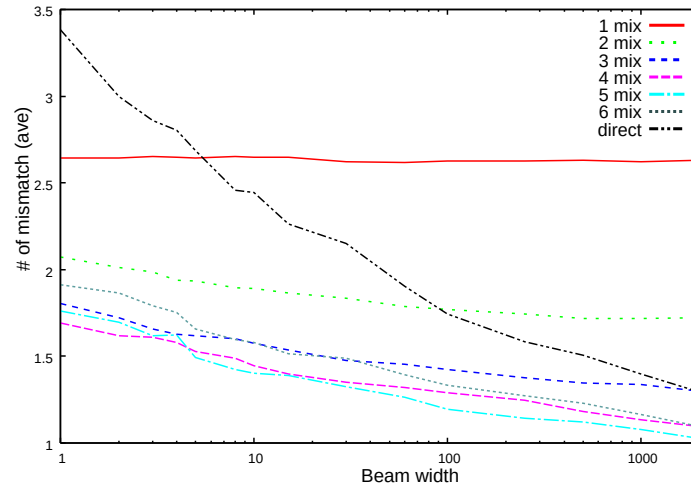
(b) ($\alpha = 1, \beta = 1, \gamma = 5$)

図 3.14: パスマックス数とミスマッチ度の関係

縦軸：各参照パターンへのミスマッチ度の平均 横軸：入力-中間パターン間ノルム
 なお、参照パターンは入力と同クラスのもの



(a) ($\alpha = 10, \beta = 5, \gamma = 0$)



(b) ($\alpha = 1, \beta = 1, \gamma = 5$)

図 3.15: ビーム幅とミスマッチ度の関係

縦軸：ミスマッチ度の平均 横軸:合成写像生成時二次元D Pのビーム幅

また、図 3.15(b) をみるとビーム幅が狭い時、パスミックス数が 4 になるまではミスマッチ度が単調に減っている。これはパスミックス数が増えることで写像先候補が増えミスマッチを補正できる可能性が増えた事によるものと思われる。しかしながら、それ以上パスミックス数が増えると、ビーム幅がせまい時には逆にミスマッチ度が増えている。これは、N-best 順に中間パターンを選んでおり、パスミックス数が増えるとそのぶんミスマッチを抱えた冗長パスがふえ、探索空間内にミスマッチ候補が増えてくる上に、ビーム幅が狭いためにそのミスマッチをうまく排除できない状態になっていると思われる。この傾向は、パスミックス数 6 のものがビーム幅 1 ではミスマッチの小ささで 4 番目なのに対し、ビーム幅が 2000 になると 2 位までに改善されていることからわかる。

同様の結果は図 3.15(a) においても確認された。

3.6 中間パタンの認識時選択指針

写像のミスマッチによる不当なパターン間距離増加を防ぐために、入力-中間・中間-正解パターンまでがミスマッチなく写像できる中間パターンである必要がる。通常、入力-正解パターンは非常に類似しているので、入力-中間パターン間類似度は中間-参照パターン間類似度とほぼ同じと考えられる。したがって、認識時に入力-中間パターン間の類似度を基準に中間パターンを選択すれば良い事がわかる。

3.7 中間パタンの生成法

前述のように、中間パターンを選択する際、入力-中間パターンが類似していることが重要である。そこで、多数の学習データから有限個の中間パターンを前処理段階で用意する際の選択基準について考えてみる。

様々な入力パターンに対して安定して、それなりの類似度の中間パターンを返すには、K-means を利用するのが妥当だと考えられる。ただし、実際の k-means アルゴリズムは重心を生成しなければならず適用は困難である。したがって 1 次元 DP での合成写像 [3] にも用いられた変形 k-means し、代表パターンを中間パターンとすることで選んだ。

また、参照パターンも同様に変形 k-means で選んだが、クラスごとに変形 k-means を行った。

3.8 合成写像時のパターン間距離定義

前述のとおり、合成写像時は直接写像時に比べて、ミスマッチ無く写像される変形範囲が広がっている。写像制限で広げる場合は、認識に最適な目的関数 D とは違う写像制限パラメータを用いて広げるので、目的関数 D がパターン間距離を定義するのに適した関数ではないことが予想できる。パスミックスで広げる場合も、本来変形できなかった領域まで中間パターンを介することによって変形することができるので D をそのままパターン間距離に使う強い理由はない。そこで、本手法においては、合成写像を求めた後に、その写像を用いて別の認識用目的関数 D_r にかけることによって認識を行う。

3.9 本手法の概略

ここで、本論文で最終的に提案する手法の概略をここに挙げる。

3.9.1 前処理

- 参照パターンを選ぶ。

これは k-means が重心を代表パターンにするのに対し、クラスタ内の他のパターンへの距離の総和が最小であるパターンを代表パターンとするものである。なお、パターン間距離はパターン間のノルムを利用している。

- 中間パターンを選ぶ。
- 中間参照パターン間二次元DPを行い写像を記録しておく。

3.9.2 認識時

1. 入力パターンに応じて中間パターンを数個選択する。
2. 入力-中間パターン間で二次元DPを行う。
3. 複数の中間パターンから得られた複数の冗長パスを用いてパスミックスする。
4. 重ね合わせられた冗長パスから、二次元DPを用いて1つの写像を選択する
5. 写像と、入力・参照画像からパターン間距離を算出

第4章 文字認識による性能評価

4.1 実験条件

4.1.1 画像データの前処理

サイズ正規化

本研究では実験データとして ETL-9b[7] データベースから、清音平仮名を用いた。まず、2階調文字データである ETL データベースから背景 255, 文字部 0 の 256 階調画像に変換した。次に、正規化のために、文字部分を矩形で切り出し、それをアスペクト比を崩す形で 14x14 ピクセルに伸張し、その周りに 2 ピクセル幅の白枠をとりつけた。

縦方向ぼかし

2.4.4 に述べたように、前処理として縦方向にぼかすことで、縦方向への探索範囲の不足を補う処理を行っている。今回の実験では実験的に認識率のよくなるパラメータを求めた。具体的には、縦方向に ± 3 ピクセル分の領域の平均値を取り、その値を 0.1 倍したものを元の特徴量に足すことによって縦方向ぼかしを実現している。

特徴量

今回は特徴量として、隣接ピクセルとの微分成分を考慮に入れた特徴量を用いた。これは画像 A のある点 (i, j) と画像 B のある点 (x, y) の特徴量間距離を式 4.1 で定義するものである。

$$\begin{aligned} d = & \|a(i, j) - b(x, y)\| \\ & + \|(a(i+1, j+1) - a(i-1, j-1)) - (b(x+1, y+1) - b(x-1, y-1))\| \\ & + \|(a(i+1, j) - a(i-1, j)) - (b(x+1, y) - b(x-1, y))\| \\ & + \|(a(i, j+1) - a(i, j-1)) - (b(x, y+1) - b(x, y-1))\| \end{aligned} \quad (4.1)$$

4.1.2 認識率実験

データセット

本実験での文字列認識実験はアルファベット順に清音平仮名 30 クラスを対象に行った。ETL データベースから各クラス 40 文字のうち、データベース収録順で前半 20 文字を参照パターンや中間パターン用の学習データ、後半 20 文字を認識実験時の入力パターンであるテストデータとした。

参照パターン選択

参照パターンを選択する手法としては、広く用いられている変形 k-means によるクラスタリングを用い、各クラス 3 つを選んだ。

中間パターンの選択

クラス区別なしで変形 k-means を行い、90 個を選んだ。

直接写像時認識実験

入力パターンとすべての参照パターン 90 個との間の二次元 DP を行い、そのときの目的関数 D の値をパターン間距離とし、一番パターン間距離が近い参照パターンを探すことで認識を行った。

4.2 パスミックス数・写像制限・コスト関数に関する実験

パスミックス数、写像制限、及び認識時のコスト関数 D_r を変更して認識実験を行った。実験条件は以下の通りである

パスミックス数: 1, 2, 及び 5

写像制限:

- A: 直接写像で認識率が良かったパラメータ ($\alpha = 10, \beta = 5, \gamma = 0$)
- B: ある程度離れたパターンにきれいな写像を作れるパラメータ ($\alpha = 1, \beta = 1, \gamma = 5$)

コスト関数:

- 写像作成に用いたものと同じパラメータ
- ($\alpha = 0, \beta = 0, \gamma = 50$)

以上の組み合わせで行った。

写像生成時 パラメータ	直接写像時 認識率	パターン間距離定義時 パラメータ	1	2	5	10
$A : \alpha = 10, \beta = 5, \gamma = 0$	86.7	same	79.3	83.3	89.3	88.5
		diff	77.8	84.8	90.7	90.0
$A : \alpha = 10, \beta = 5, \gamma = 0$ (縦方向ぼかしなし)	89.6	same	76.3	81.9	88.1	87.8
		diff ¹	71.5	80.3	85.2	86.3
$B : \alpha = 1, \beta = 1, \gamma = 5$	80.0	same	80.3	84.1	83.0	81.1
		diff	79.3	85.1	84.1	83.0

表 4.1: 認識率実験結果
直接写像時、及びビーム幅 1,2,5 時の認識率
same は写像生成時と同じパラメータ
diff は ($\alpha = 0, \beta = 0, \gamma = 50$)

なお。直接写像時は写像制限 A の場合 86.7%で写像制限 B のとき 80.0%であった
尚、縦方向をぼかさないう、写像制限 A をかけたとき、認識率が 89.6%となり、直接写
像で実験的にふったパラメータのなかでは最大であった。

合成写像を用いたときの最高認識率が 90.7% であり、直接写像の最高認識率 89.6%を
超える結果となった。これは、合成写像を用いて、認識率を落とすことなく高速化が可能
で有る事を示している。しかしながら縦方向ぼかしがない場合に限ってみると、合成写像
のほうが直接写像よりも認識率が低下している。縦方向ぼかしがない場合、探索時に縦方
向の隣接ピクセルの影響を受ける範囲が非常に狭くなってしまふ。そのため写像制限を縦
方向にだけ厳しくしたものと同じような状態になっているものと思われる。前に議論し
たとおり、合成写像では入力-中間パターン間での変形を許容できないほどの厳しい写像制
限はミスマッチ写像の原因となる。そのために認識率が低下していると思われる。

パスミックス数 10 に注目してみると、どのパターンでも認識率が低下し始めている。
これは 3.5 で示したように、パスミックス数に対してビーム幅が狭いためにミスマッチ度
が上昇し、結果としてパターン間距離の計算に誤差が増え、認識率が落ちてきている物と思
われる。

認識時の目的関数 D_r を写像生成用目的関数 D とは別の物 ($\alpha = 0, \beta = 0, \gamma = 50$) にし
たとき、ぼかしなしの場合を除いて認識率が上昇している。これは中間写像を利用可能す
るために写像制限を甘くした分、写像生成よう目的関数 D はパターン間距離定義には不向
きなものになってしまう傾向があることを示唆するものだとおもわれる。そのため特に合
成写像を利用した二次元 DP においては写像生成用目的関数 D と、パターン間距離定義用
コスト関数 D_r は別のパラメータを振って検討する必要性が有るといふ議論に一致する。

¹パターン間距離算出時もぼかしなし

パスミックス数2の時に注目すると、パスミックス数を増やせない場合も、写像制限をゆるめることで、ある程度の認識率を出すことが可能であることが見て取れる。これは写像制限を緩めるとマッチ度が上がる分離した入力-中間パターンしかなくても対応できる可能性が高くなるという3.3.2での議論に一致する。

4.3 計算量に関する考察

合成写像によって計算量がどの程度軽減されるかについてここで定式化する。

- T_d : 直接写像二次元D Pにかかる時間
- T_s : 合成写像二次元D Pにかかる時間
- $R_s = T_s/T_d$
- T_n : 画像間のノルムを算出するのにかかる時間
- N_R : 参照パターン数
- N_M : 中間パターン数
- N_m : 選択された中間パターン数

とすると、合成時の計算時間 $Time_{syn}$ 、直接写像の計算時間 $Time_{direct}$ はそれぞれ

$$\begin{aligned} Time_{syn} &= T_n N_M + N_m T_d + T_s N_R \\ &= T_n N_M + (N_R R_s + N_m) T_d \end{aligned} \quad (4.2)$$

$$Time_{direct} = N_R T_d \quad (4.3)$$

となるので、計算時間比は $N_M \leq N_R$ であるとする $T_n \ll T_d$ なので

$$\begin{aligned} Rate &= \frac{T_n N_M}{N_R T_d} + \frac{N_R R_s + N_m}{N_R} \\ &\simeq R_s + \frac{N_m}{N_R} \end{aligned} \quad (4.4)$$

となる。

ビーム幅	直接写像	1	2	5	10
1	-	0.039	0.043	0.049	0.056
10	0.43	0.056	0.071	0.11	0.15
100	1.4	0.23	0.45	0.85	1.28
1000	11	-	-	-	-
2000	24	-	-	-	-

表 4.2: 計算量 (秒)

直接写像、及びパスミックス数 1,2,5,10 での計算量
値は、Celeron 2.6GHz での実測値 (平均)

4.3.1 ビーム幅と計算量の関係

実際の計算量を求めるにあたって、 R_s を求める必要がある。実際の R_s は直接写像・合成写像のビーム幅及び、パスミックス数で決まる。そこで、ベンチマークを取りそこから実際の計算量を推定した。

表 4.2 をみると、合成時には同じビーム幅ならば、パスミックス数が大きいほうが処理時間がかかっている。これはビーム探索がかかっている場合でも探索空間のサイズの影響を受けるからである。

ビーム探索では、ある入力画像上の点からの写像先候補が複数ある場合、一時的にビーム幅よりも大きなサイズの候補のコストを評価し、その後ビーム幅を超える分だけ切り捨てる。探索空間が広いという事は写像先候補が多数ある点が多く存在するということでありその分処理時間がかかる。

合成写像時ビーム幅 2000、合成時ビーム幅 5、パスミックス数 5 という今回の例の場合、 $R_s = 0.006$ となり式 4.4 の R_s の項は無視できる程小さい。従って、計算時間は理論上 1/18 に短縮される。なお、実際の実験環境では大量の合成済み写像ファイルを取得する処理等があるので、さらに多少の時間を必要となる。そのため実機での速度は合成写像利用時に 173 秒になる。これにより実機上では 1/12.5 倍に短縮されたことが示された。

N-best	含有率
1	73.7
2	81.5
5	89.3
10	92.9

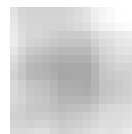
表 4.3: N-Best 中間パターンに正解パターンが含まれる確率



(a) 原画像



(b) ぼかしたもの



(c) 極端にぼかしたもの

図 4.1: 各ぼかし度合いの一例

N-best	含有率	合成写像認識率
1	59.9	76.7
2	63.3	83.7
3	67.8	86.3
4	70.0	88.9
5	72.6	90.7
6	74.1	90.4

表 4.4: 極端にぼかして中間パターンを選択した場合

4.4 中間パターン選択精度と認識率の関係

今回の場合、表に示す用にかかなりの確率で N-best で選んだ中間パターン内に正解パターンが存在している。そこで、正解パターンを中間パターンに含む程高精度の中間パターン選択でないと合成写像を利用した方法は使えないのでは無いかという疑問が浮かぶ。

そこで図 4.4 のように極端にぼかし、ノルムだけではパターン間距離の推定をするのを難しくしてみた。

結果として表 4.4 のように、N-best での正解パターン含有率は低くなった。しかし、このような状況下においても合成写像を利用して求めた認識率の低下はそこまで大きいものではなかった。これは、この合成写像という技術を使うにあたって必須となる中間パタンの高速選択においてどの中間パターンが選ばれるか？ということに対してある程度のロバスト性を持っていることを示している。これは文字以外への適用を考える場合重要な知見であろう。

第5章 結論

5.1 本研究の結果

一次元DP (DTW) で利用されていた合成写像を用いた高速化を二次元DPに適用し、実機上で12.5倍もの高速化が達成できた。また、パスミックス等の手法の導入により、直接写像を行うよりも認識率をわずかではあるが改善する事も可能となった。これにより、計算量を理由に実用とならなかった二次元DPが実用へと一歩進んだと言えるだろう。

また、このパスミックスをし合成写像段階で再選択するという行為は、

- 合成写像によって直接写像が近似可能である
- 全探索空間がビーム探索を必要とするほど広大である。

という二つの条件を満たせば、二次元DPに限らず有効であると考えられ、他の手法への転用も期待できる。

5.2 今後の課題

前述のように、入力-中間パターン間の類似度の推定精度が高速中間パターン選択時に悪い場合であってもある程度ロバストである。これは、文字のように画像間のノルムを取るだけである程度の認識率が出る様な用途でも利用できる可能性を示唆している。したがって、他のドメインのデータに適用できるか検証するのには大きな価値があるだろう。たとえば、骨画像処理 [4] などの医療画像処理などへの利用が期待される。また、今回は細かいパラメータの最適化については言及しなかったが、実験的であれ理論的にであれ、パラメータの最適化を行った上での評価が必要であろう。

参考文献

- [1] 内田誠一, 迫江博昭, “単調連続 2 次元ワープ法の検討”, 信学技報, PRMU96-81, 1996
- [2] 勝股充, 鈴木尚, 徳田恵一, 北村正, “単調連続 2 次元 DP アルゴリズムの階層化”, 信学論, Vol.J85-D2, No.9, pp.1382-1391, 2002
- [3] 森本 大樹 “ パターン間の合成写像を用いた非線形時間伸縮法の高速化 ” '04 JAIST
- [4] 福田 将彦 “ 二次元 warping を用いた頸部 X 線画像からの骨年齢推定に関する研究 ” '02 JAIST
- [5] 宮崎 洋光, 内田 誠一, 迫江 博昭 「粗密 DP による 2 次元ワープの高速化の検討」 の 国情報シンポジウム 2004
- [6] 松本 直樹, 内田 誠一, 迫江 博昭 「弾性マッチングを用いた画像パターン認識のための標準パターン設定法」 信学技報 2003
- [7] ETL9 手書き文字データベース <http://www.is.aist.go.jp/etlcdb/etln/etl9/etl9.htm>