

手書き文字列認識を目的としたワープによる文字傾きの補正法

平 英 二* · 石田敏之* · 内田誠一** · 迫江博昭**

Slant Correction of Handwritten Word Using Two-Dimensional Warping

Eiji TAIRA, Toshiyuki ISHIDA, Seiichi UCHIDA and Hiroaki SAKOE

(Received December 15, 2000)

Abstract: Slant correction of characters is necessary in the segmentation of a handwritten word into component characters. Conventional slant correction techniques estimate the average slant angle of component characters and correct only uniform slant resulting residual error for each character. In this paper, a slant correction technique which can well correct nonuniform slant is proposed. In the present technique, the slant correction problem is formulated as a non-linear mapping problem of slanted strokes onto vertical straight lines. Then a dynamic programming-based two-dimensional warping algorithm is applied to optimize the mapping. The effectiveness of the present technique was shown by experiments.

Keywords: Slant correction, Two-dimensional warping, Dynamic programming, Word recognition, Segmentation

1. はじめに

文字単位の標準パターンに基づく文字列の認識では、前処理としてもしくは認識処理と並行して文字のセグメンテーションが必要となる。活字列の場合は単純に文字間の垂直スペースを探索することでセグメンテーションが可能である。しかし、手書き文字列の場合は文字の傾きや続け書きによりそのような垂直スペースが存在しない場合が多い。そこで本論文では、傾いた文字を立体に直す、傾きの補正処理を提案する。

手書き文字列の傾き補正法は、これまでもいくつか検討されている。Bozinovicら¹⁾は、横ストロークを除去し、残った縦ストロークの平均角度を文字列の傾きとする手法を提案した。Guillevicら²⁾やKavallieratouら³⁾は、投影ヒストグラムから文字列の平均の傾きを推定する手法を提案した。Simonciniら⁴⁾やMadhvanathら⁵⁾、Dingら⁶⁾は、文字の輪郭線のチェーンコードから傾きを推定する手法を提案した。これらの従来法はいずれも文字列全体にわたる平均的文字傾きを推定して補正している。しかし、各文字毎に見るとこのように推定した傾きでは補正が不十分な文字もあれば、補正過多のために逆方向に傾いてしまうことも起こりうる。そのような場合、部分的な傾きも補正できるような手法が望まれる。

本論文では、2次元-2次元非線形写像(2次元ワープ)を用いた文字傾きの補正法を提案する。本手法は部分的な傾きも補正可能である。原理的には、傾き補正の問題を画像の非線形写像問題としてとらえ、2次元ワープのアル

ゴリズム⁷⁾を応用することで最適な変形結果、すなわち傾き補正画像を求めようというものである。ここで変形の方針は、長い縦ストロークの傾斜を垂直に補正することを基本とし、文字の形を崩さないように画素位置の単調連続性を保存することとする。

以下では、本手法の基本原則を述べ、実際に手書き英文字列に適用した実験を行ない、本手法の有効性を確認する。

2. ワープを用いた文字傾きの補正法

本論文で提案するワープを用いた文字傾きの補正法では、入力とする2値文字列画像内から傾斜した長い縦ストロークを検出しながら、その傾きを単調連続関数により近傍に伝播させることで、検出したストロークを含む文字やその付近の補正を行なう。

2.1 傾き補正問題の定式化

入力画像を $A = \{a(i, j) | i = 1, \dots, M, j = 1, \dots, N\}$ とする。ここで A の傾き補正処理後の画像が存在したと想定し、それを $B = \{b(x, y) | x = 1, \dots, M, y = 1, \dots, N\}$ とする。ここで $a(i, j), b(x, y)$ の値は0(黒画素, 文字線)もしくは1(白画素, 背景)とする。また M, N はそれぞれ横, 縦のサイズを表す。

傾き補正は2次元-2次元写像(ワープ) $B \rightarrow A$ を最適化し、その逆写像を A に施すことで B を求める。しかし、実際にはワープの最適化段階において B は存在しない。そこで B は傾きのない垂直ストロークによりなる画像であるという性質を仮定して、その性質を手掛かりに B, A 間のワープを最適化する。

平成12年12月15日受付

* 知能システム学専攻修士課程

** 知能システム学部門

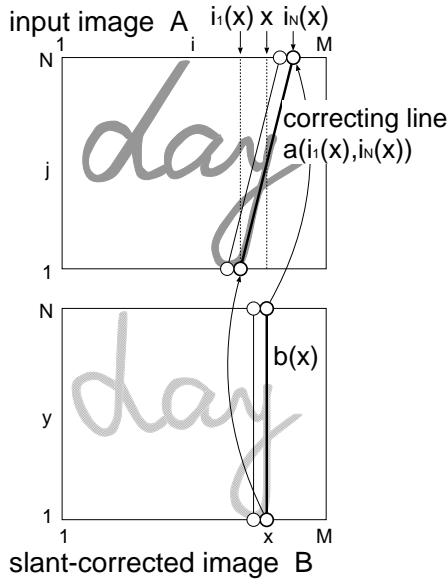


Fig.1 Slant correction using two-dimensional warping.

まず準備として B の第 x 列を $b(x) = [(x, 1), \dots, (x, y), \dots, (x, N)]$ と表し, $b(x)$ のワープによる A 上での像を補正線分と呼ぶ (Fig.1). 補正線分の両端の座標をそれぞれ $(i_1(x), 1), (i_N(x), N)$ とし, 両端以外の座標は $(i_1(x), 1), (i_N(x), N)$ の線形補間により求める. この補正線分を $a(i_1(x), i_N(x))$ と表現する. 但し, $i_1(x), i_N(x)$ は整数変数とする.

本手法では $i_1(1), i_N(1), \dots, i_1(M), i_N(M)$ を傾き補正の制御変数としてワープの最適化を図る. $i_1(x), i_N(x)$ を変化させることで補正線分の傾き角は変化する. 傾き角は本質的には連続量であるが, ここでは整数値 $i_1(x), i_N(x)$ を用いて離散化し, 扱いやすくしている.

列 $b(x)$ 上には垂直ストロークに対応する連続した黒画素が存在していると想定しているので, 補正線分上に連続した黒画素が存在すれば高い評価量を与えるものとする. 本手法ではこの考えを元に評価関数を定式化する. 具体的な形式は後述する.

補正線分を選択する際, 制御変数 $i_1(x), i_N(x)$ は次の3つの制約条件を満たすものとする. 第一の制約条件は

$$\left\lfloor \frac{i_1(x) + i_N(x)}{2} \right\rfloor = x \quad (1)$$

であり, 補正線分の候補を表している (Fig.2). 第二の制約条件は単調性条件

$$\left. \begin{aligned} i_1(x) &\geq i_1(x-1) \\ i_N(x) &\geq i_N(x-1) \end{aligned} \right\} \quad (2)$$

である. この単調性により画素の位置順の逆転を回避することができる. 不自然な変形を防ぐには一般に単調性以外にも連続性を満たす必要があるが, これは制約(1), (2)より自然に満たされる. すなわち, たとえば $i_1(x) \geq$

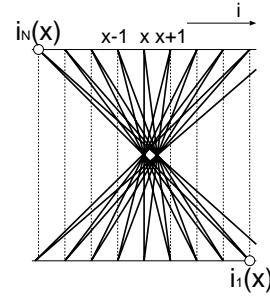


Fig.2 Correcting lines of x th column.

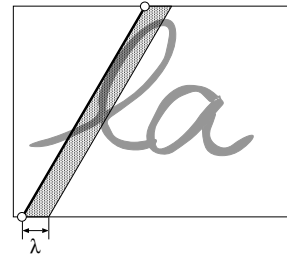


Fig.3 Correcting line with width λ .

$i_1(x-1) + 4$ とすると, 制約(1)により $i_N(x) < i_N(x-1)$ となり, 制約(2)を満足しない. 第三の制約条件は許容する傾斜を規定する条件

$$\left. \begin{aligned} |i_1(x) - x| &\leq W \\ |i_N(x) - x| &\leq W \end{aligned} \right\} \quad (3)$$

である. ここで W は正の整数定数とする.

本手法は, 制約(1)-(3)を満たし, かつ評価関数

$$\sum_{x=1}^M d(i_1(x), i_N(x) | x) \quad (4)$$

を最大にする $i_1(1), i_N(1), \dots, i_1(M), i_N(M)$ を求める. ここで $d(i_1(x), i_N(x) | x)$ は補正線分 $a(i_1(x), i_N(x))$ に与える評価量である. その値は $a(i_1(x), i_N(x))$ に幅 λ を持たせ (Fig.3), その領域内での最大連続黒画素数とする. こうすることで, 曲線ストロークにも対応でき, 曲線で構成される文字についても傾きの補正が可能となる.

ところで, 垂直方向に短いストロークはセグメンテーションにおいて意識する必要はないと考えられる. またそのようなストロークは文字自体の形状が強く反映されており, 評価対象としてしまうことはかえって不自然な補正結果を生じさせる. そこである閾値 ϵ を与えて, 評価量が ϵ 未満の補正線分は全て評価量を0とする.

評価関数(4)の最大化により求められた $(i_1(1), i_N(1)), \dots, (i_1(x), i_N(x)), \dots, (i_1(M), i_N(M))$ より補正画像 B を実際に求める処理は, 補正線分 $a(i_1(x), i_N(x))$ を B の第 x 列 $b(x)$ に写像することで実現される. このとき $a(i_1(x),$

$i_N(x)$ の両端以外の各要素の i 座標は一般に非整数値となるため、各座標における特徴量を何らかの方法で与える必要がある。本論文では最近傍補間を用いる。

以上で定式化された最適化問題をそのまま文字画像に用いても、不自然に文字が変形されてしまう場合がある。この原因として、評価関数(4)が変形量について何も評価されていないことが考えられる。そこで変形量を評価するペナルティ関数を適当な重みで評価関数(4)に加え、それらをまとめて最大化する(付録A参照)ことで不自然な変形を回避する。

2.2 DP アルゴリズム

2.1で定義された制約(1)-(3)の下での評価関数(4)の最大化問題は、動的計画法(dynamic programming: DP)を用いることで効率的に解くことができる。本手法におけるDPアルゴリズムを以下に示す。

(1) 初期状態

制約(1)-(3)を満たす全ての $i_1(1), i_N(1)$ について

$$g(i_1(1), i_N(1)|1) = d(i_1(1), i_N(1)|1)$$

を求める。

(2) DP漸化式

以下の計算を $x = 2, \dots, M$ について順に行なう。制約(1)-(3)を満たす全ての $i_1(x), i_N(x)$ について

$$\begin{aligned} g(i_1(x), i_N(x)|x) \\ = d(i_1(x), i_N(x)|x) \\ + \max_{\substack{i_1(x-1) \\ i_N(x-1)}} [g(i_1(x-1), i_N(x-1)|x-1)] \quad (5) \end{aligned}$$

を求める。ここで $g(i_1(x), i_N(x)|x)$ は $b(x)$ の像となる補正線分として $a(i_1(x), i_N(x))$ を選んだときの第1列から第 x 列までの最大累積評価量である。また第2項の最大値選択は制約(1),(3)を満たし、かつ $i_1(x), i_N(x)$ と制約(2)を満たす $i_1(x-1), i_N(x-1)$ について行なう。

(3) 終了

評価関数の最大値は

$$\max_{i_1(M), i_N(M)} g(i_1(M), i_N(M)|M)$$

で与えられる。最大値選択は $i_1(M), i_N(M)$ について制約(1),(3)を満たす範囲で求める。

終了後、バックトラック処理を行なうことで $(i_1(1), i_N(1)), \dots, (i_1(M), i_N(M))$ 、すなわち補正線分の最適系列を求めることができる。

3. 実験

本節では、本手法を用いた英単語の傾き補正実験について述べ、その結果により本手法の文字傾きの補正についての有効性を示す。本実験において、各列で許容する最大傾きを表すパラメータ W を63、補正線分の幅 λ を4、

評価量の閾値 ϵ を25とした。なお、サイズ 64×256 の画像で傾き補正にかかる計算時間は0.99秒(PentiumIII, 500 MHz)であった。

3.1 実験 1

入力画像としてCEDAR CDROM-1 (city and state words)に含まれる手書き英単語画像を用いて、本手法による傾き補正の効果を確認する実験を行なった。各画像には以下の手順で前処理を行なった。画像の縦サイズが64画素になるように正規化した。その際、画素の輝度値は共1次内挿法を用いて補間した。さらに、2値化処理を施し、左右に W 画素分の空白列を加えた。

Fig.4に本手法による傾き補正の例を示す。上が入力画像、下が傾き補正処理後の画像である。Fig.4(a)では、どの入力画像に対しても、傾いていた文字が形を崩すことなく立体に近い形に補正されているのが分かる。またFig.4(b)を見ると本手法は罫線を含む画像も問題なく補正できることが分かる。しかし、Fig.4(c)のように"X"のようにもともと斜線で構成される文字については逆に傾いてしまう。特に最後の例ではその影響で近傍の文字まで傾いている。これは本手法における問題点である。

3.2 実験 2

文字列全体にわたる文字傾きを推定して補正する手法と本手法の定性的及び定量的な比較を行なった。ここで比較対象とした全体的な傾きの推定法には、従来法¹⁾⁻⁶⁾ではなく本手法と同一の評価関数を用い、傾き角が画像全体で一定であると制限した方法を用いた。以下、この手法を従来法と同じ目的の手法であることから単に従来法と呼ぶ。原画像として5種類の印字英単語画像(Fig.5(i))を用い、それらを一定の傾き角で一様に変形した画像(Fig.5(ii))及び列毎に傾き角を正弦関数に従って変化させながら変形した画像(Fig.5(iii))を作成した。以下、作成した画像をそれぞれTrans1, Trans2と呼ぶ。

2種類の変形画像を従来法、本手法それぞれで傾きを補正した結果を(Fig.5(a),(b))に示す。一定の角度で傾いたTrans1については、どちらの手法でもうまく補正されている。但し、image4,5は斜線で構成される文字("Y", "X")を含んでおり、本手法ではそれらの影響により局部的に過変形を起こしているのが分かる。一方、列毎に異なる角度で傾いたTrans2においては、従来法では補正できていないのに対し、本手法では比較的良好に対処できている。

次にこの結果を定量的に評価した。すなわち求められた各列の傾き角を補正線分の両端座標から求め、原画像を変形する際の傾き角との平均2乗誤差を測定した。測定した2乗誤差をTable 1に示す。Trans1については、どちらの手法とも誤差は小さく、適切に対応できていること

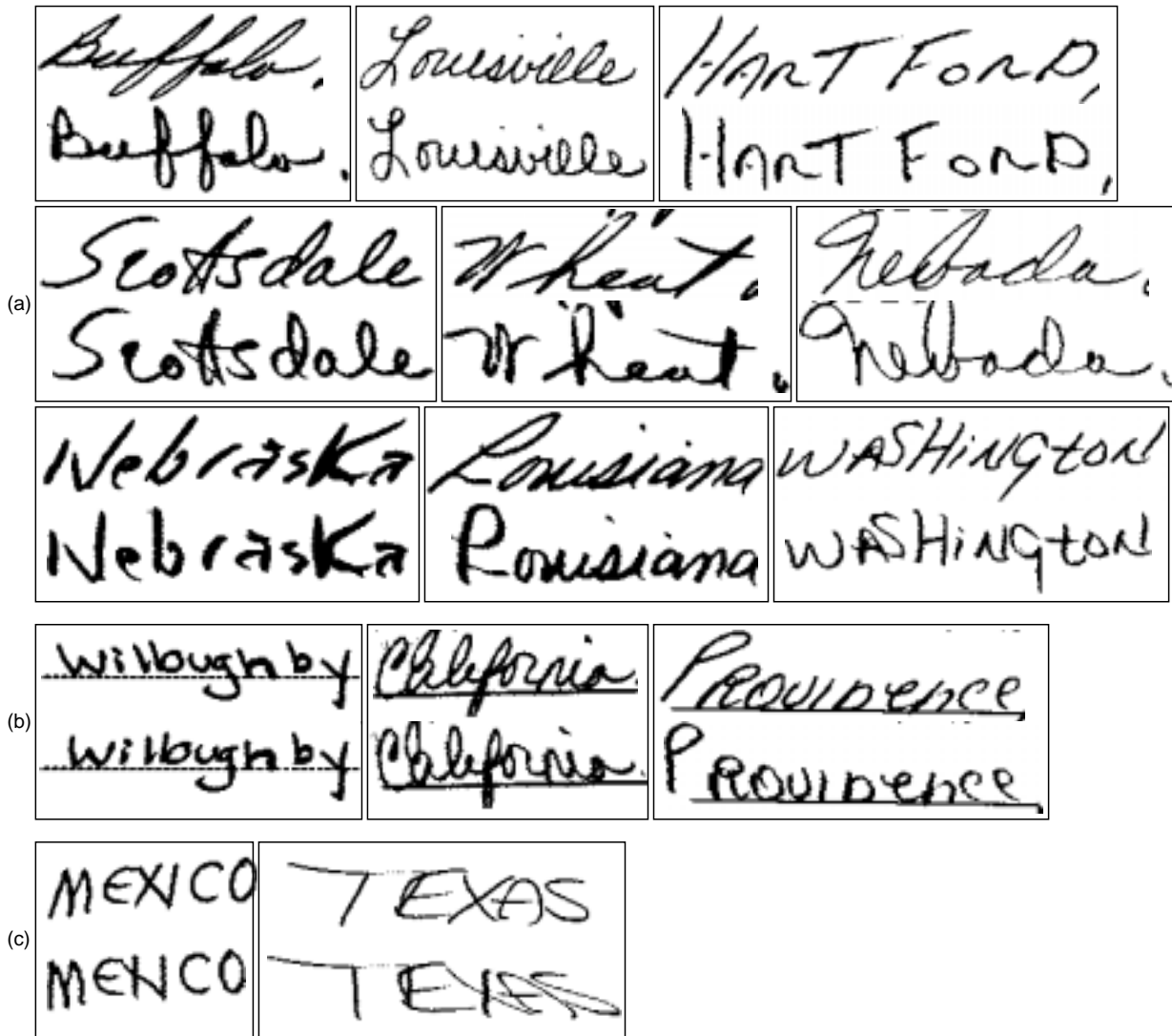


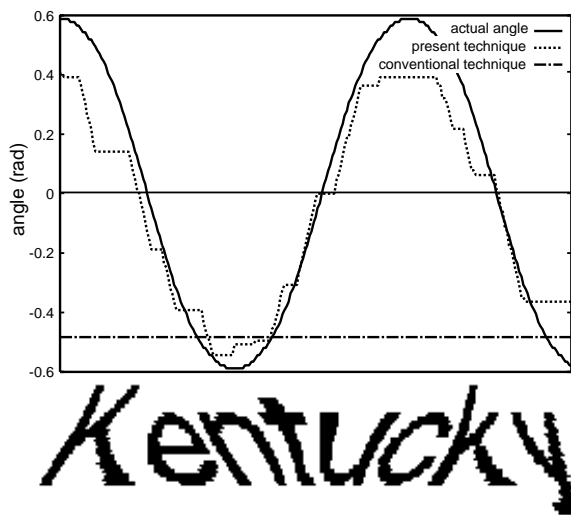
Fig.4 Examples of slant corrected words.



Fig.5 Slant correction for artificially slanted words by the conventional technique or the present technique.:(i)Original word images. (ii),(iii)Artificially slanted word images. (a)Slant corrected results of the conventional technique. (b)Slant corrected results of the present technique.

Table 1 Mean square error between estimated and actual angles.

Original word image	Mean-square-error ($\times 10^{-3}$)			
	Conventional technique		Present technique	
	Trans1	Trans2	Trans1	Trans2
image1	0	467	0.188	29.3
image2	0	578	0.007	188
image3	0	318	0.337	49.0
image4	0	224	8.47	140
image5	0	224	6.66	183

**Fig.6** Actual angle and estimated angles.

が分かる．image4,5で本手法に多少の誤差が出ているのは，前述したように”Y”，”X”の影響による．Trans2における結果を見ると，本手法の傾き補正能力は従来法の傾き補正能力より明らかに優れていることが分かる．

さらに傾き角の理論値と従来法及び本手法における推定値を列毎に比較したグラフをFig.6に示す．ここでは用例としてimage1を用いた．本手法が理論値を追従できているのが分かる．従来法で推定した傾き角度は約-0.5(rad)となっている．なお平均角度(約0(rad))になっていない理由は，従来法では画像全体の平均傾きを推定するのではなく，長い垂直ストロークが多く検出できる傾き角を推定しているためである．

4. ま と め

本論文では，手書き単語認識において問題となる文字の傾きを補正する方法として2次元ワーブを用いた手法を提案した．本手法では，部分的に異なる角度で傾いている文字列でも傾きの補正が可能である．実験では，本手法の文字傾きの補正における有効性を定性的に示した．

また，入力文字列全体にわたる文字傾きを推定して補正する手法との比較実験から本手法の傾き補正能力を定量的に示した．今後の課題として，”X”のようにもともと斜線で構成される文字も適切に傾きを補正できるように改善することが挙げられる．

謝 辞

本研究の一部は文部省科学研究費(基盤研究C(2), No.10680385)によった．

参 考 文 献

- 1) R.M. Bozinovic and S.N. Srihari: Off-Line Cursive Script Word Recognition, IEEE Trans. Patt. Anal. Mach. Intell., Vol. 11, No. 1, pp. 68-83, Jan. 1989.
- 2) D. Guillevic and C.Y. Suen: Cursive Script Recognition: A Sentence Level Recognition Scheme, Proc. 4th IWFHR, pp. 216-223, Dec. 1994.
- 3) E. Kavallieratou, N. Fakotakis and G. Kokkinakis: A slant removal algorithm, Patt. Recog., Vol. 33, No. 7, pp. 1261-1262, Jul. 2000.
- 4) L. Simoncini and Zs.M. Kovács-V: A System for Reading USA Census'90 Hand-Written Fields, Proc. 3th IC-DAR, Vol. II, pp. 86-91, Aug. 1995.
- 5) S. Madhvanath, G. Kim and V. Govindaraju: Chain-code Contour Processing for Handwritten Word Recognition, IEEE Trans. Patt. Anal. Mach. Intell., Vol. 21, No. 9, pp. 928-932, Sep. 1999.
- 6) Y. Ding, F. Kimura, Y. Miyake and M. Shridhar: Accuracy Improvement of Slant Estimation for Handwritten Words, Proc. Int. Conf. Patt. Recog., Vol. 4, pp. 527-530, Sep. 2000.
- 7) 石田敏之，内田誠一，迫江博昭: 手書き文字認識における字形変動処理に関する一検討 - ダッチロールワーブ - ，九州大学大学院システム情報科学研究科報告, Vol. 5, No. 1, pp. 99-104, Mar. 2000.

付 録

A ペナルティ

本実験では以下のように定義される2つのペナルティ関数 P_1, P_2 を用いる．

P_1 は，第 x 列で選択した補正線分の傾きが第 $(x-1)$ 列で選択した補正線分の傾きと異なる時，すなわち $i_1(x) \neq i_1(x-1) + 1$ または $i_N(x) \neq i_N(x-1) + 1$ のときに課せられるペナルティ関数である．そのペナルティ量は第 x 列で選択した補正線分 $a(i_1(x), i_N(x))$ 上の黒画素の全個数とする．文字と重なる補正線分ほどペナルティ量は大きい．すなわち，文字に関わる場所では傾きを維持するように補正線分を選択するため，1文字毎の傾き補正をできるだけ一定の角度で行なうようになる．

また，補正線分の両端以外の座標は最近傍補間により求められるため，第 x 列で選択した補正線分的一端が第 $(x-1)$ 列で選択した補正線分的一端と同じ座標をとると，

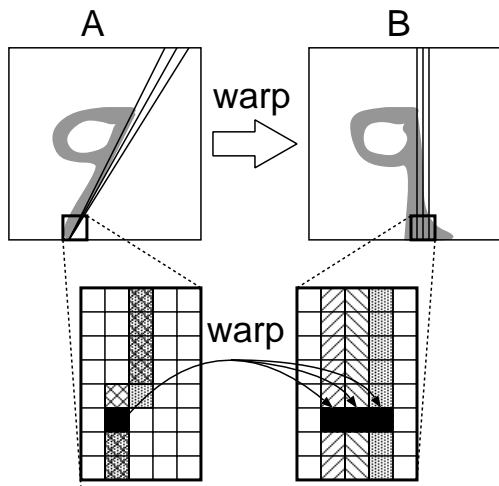


Fig.A Unnatural translation.

その端の近傍もそれぞれ同じ座標をとる場合が多い。そ

こに黒画素があると、ワーブした後の画像では、その黒画素が横に広がる(Fig.A)。これが連続して起こると不自然な変形となる。 P_2 はそれを防ぐためのペナルティ関数で、 $i_1(x) = i_1(x - 1)$ または $i_N(x) = i_N(x - 1)$ のときに課する。そのペナルティ量は第 x 列で選択した補正線分 $a(i_1(x), i_N(x))$ 上の同じ座標をとった一端の近傍にある黒画素の個数とする。本論文では近傍の範囲を $N/4$ としている。

これらのペナルティの導入に応じて、DP漸化式(5)は

$$\begin{aligned}
 &g(i_1(x), i_N(x)|x) \\
 &= d(i_1(x), i_N(x)|x) \\
 &\quad + \max_{\substack{i_1(x-1) \\ i_N(x-1)}} [g(i_1(x-1), i_N(x-1)|x-1) \\
 &\quad - \alpha P_1(i_1(x), i_N(x), i_1(x-1), i_N(x-1)|x) \\
 &\quad - \beta P_2(i_1(x), i_N(x), i_1(x-1), i_N(x-1)|x)]
 \end{aligned}$$

となる。ここで $\alpha, \beta (> 0)$ は重み係数である。3の実験では $\alpha = 1, \beta = 2$ とした。

