

論文

## 動的計画法によるパターン・マッチング\*

磯道義典\*\*・小川辰男\*\*\*

### Abstract

In order to recognize handwritten characters, a new type of pattern matching technique is introduced. This technique relies on the hypothesis that patterns are a kind of elastic bodies. From this hypothesis a new definition of distance between two patterns is derived.

This distance is defined as the minimum of sums of the internal energy of the elastic bodies and the mismatching quantities. The minimization is accomplished by using the Dynamic Programming Method. The calculation of the distance between two patterns, which are presented on  $20 \times 20$  lattice points, requires 1.2 second on Tosbac 5,600.

### 1. まえがき

最近文字図形等の処理において、動的計画法が利用されるようになってきた。その代表的なものとして Levi-Montanari による Grey-Weighted Skeleton<sup>①</sup>、鹿野-鳥脇-福村による Wave Propagation Method<sup>②</sup> および森-森らによる場の効果法<sup>③</sup> をあげることが出来る。これらの手法は文字図形などの特徴抽出のために考案されたものである。しかし、動的計画法そのものは特徴抽出以外の場面においても利用可能なものである。本論文においては、迫江-千葉<sup>④</sup> の考え方を参考にして動的計画法による文字図形のマッチングの問題を考察することにする。

本論文において導入する手法は、中野-中田-中島の導入した周辺分布におけるマッチング<sup>⑤</sup> よりも柔軟性を持つ。それゆえ手書き文字のようにバリエーションの多いパターンは本手法によって始めて取り扱いが可能となった。こうしたマッチング能力の拡大にもかかわらず、本手法の必要とする計算量は周辺分布によるマッチングの場合の約4倍となるにすぎないことが証明された。

本論文の手法を簡単な模擬データに適用した結果所

期の性能を持つことも確かめられた。

### 2. 文字図形間の距離

入力文字図形と、システム中の標準パターン(複数)との間の距離を計算し、その最小距離に対応する標準パターン名を答えれば、文字パターンの識別が実現出来る。普通この距離は、メッシュ化パターンの対応メッシュにおける明るさの差の総和として与えられている。この場合には単純類似度による識別法と同一の結果がえられることとなる。

本論文においては、パターン間の距離を従来のものとは異なる形式で定義し、これによって新しい識別法を導入する。

文字図形は2次元面上に表現されるものであるが、動的計画法適用のために一次元図形の集合であると考えることにする(これは図形の走査を行なうことによって簡単に実現される)。それゆえ以下においてまずこの一次元図形間の距離の定義から着手することとする。図形全体としての距離は各一次元図形間の距離の和として定義されることとなるわけである。

#### 2.1 図形間の距離

一次元図形  $S_1(x), S_2(x)$  間の距離として

$$\int_0^1 |S_1(x) - S_2(x)| dx \quad (1)$$

とすれば、普通に知られているパターンマッチングが構成されるわけである。しかし、本論文では媒介変数  $t$  と2つの関数  $u(t), v(t)$  を使って異なった点にお

\* Pattern Matching by Using Dynamic Programming, by Yoshinori ISOMICHI (Pattern-Information Division Electro-technical Laboratory) and Tatsuo OGAWA (Chiba Institute of Technology)

\*\* 電子技術総合研究所パターン情報部

\*\*\* 千葉工業大学電子工学科

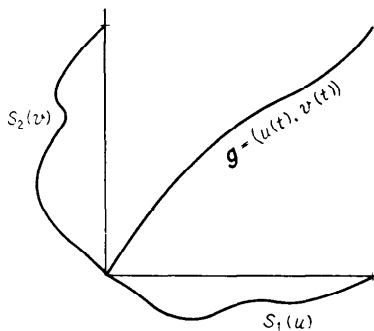


Fig. 1 Correspondence with elasticity

ける  $S_1$  と  $S_2$  の比較を行なうこととする。すなわち

$$S_1(u(t)), S_2(v(t)) \quad (0 \leq t \leq 1) \quad (2)$$

の比較を行なおうというのである。この状況は Fig. 1 に示されており、図中右上がりの曲線は  $(u(t), v(t))$  の表現となっている。ここで  $S_1$  軸と  $S_2$  軸とは全単射、双連続に対応させられていなければならないから、グラフは連続かつ強単調でなければならない。ここで  $S_1$  と  $S_2$  の反転した対応は除外してよいから、結局グラフは連続かつ強単調増加と言うことになる。以上の結果を多少一般化して数式表現すれば以下のようなる。

(1)  $t_1 \leq t_2$  ならば

$$(u(t_1), v(t_1)) \leq (u(t_2), v(t_2)) \quad (3)$$

(2)  $t \rightarrow t_0$  ならば

$$(u(t), v(t)) \rightarrow (u(t_0), v(t_0)) \quad (4)$$

ここでベクトルの順序および収束はコンポーネントワイズの順序と収束のことである。すなわち、それぞれ

$$u(t_1) \leq u(t_2) \text{ かつ } v(t_1) \leq v(t_2) \quad (5)$$

$$u(t) \rightarrow u(t_0) \text{ かつ } v(t) \rightarrow v(t_0) \quad (6)$$

を意味している。式 (3), (4) のような記述法は便利であるので以後時々使うこととする。

式 (3), (4) をみたとす  $(u(t), v(t))$  はある 1 つのグラフに対応しているが、逆にあるグラフのこの形式の表現は無数に存在する。そこでグラフとパラメータ表現の双一意性を保証するため、パラメータ表現には次の条件を課すものとする。

$$t = \frac{u(t) + v(t)}{2} \quad (7)$$

これはグラフを  $45^\circ$  線上に直交射影した時の  $45^\circ$  線上の位置をパラメータとして採用することを意味している。この条件下では  $t$  と  $u(t)$  とを与えさえすれば  $v(t)$  は自動的に決定されることとなる。これを具体的に与えらる

$$v(t) = 2t - u(t) \quad (8)$$

となる。以上のいくつかの条件を満足する点集合  $\{(u(t), v(t))\}$  を以後グラフと呼び、記号  $g$  で表示することとする。

ところで  $S_1(u(t))$  と  $S_2(v(t))$  との比較を行なうとしても、 $S_1$  軸上のある近傍に  $S_2$  軸上の広い領域が対応するというようなことはパターンと言うものの性質上おこりえない。そこでこのような対応が排除されるように 2 つのパターンをおのおの一種の弾性体であるとみなし、極端な歪みをもつ対応に対しては大きなペナルティーが付加されるようなマッチング尺度を導入することとしよう。この要請をみたす尺度として本論文では次のものを採用する。

$$d(S_1, S_2; g) \triangleq \int_0^1 \omega(S_1, S_2; g) dt \quad (9)$$

ただし

$$\begin{aligned} \omega(S_1, S_2; g) &\triangleq |S_1(u(t)) - S_2(v(t))| \\ &+ \alpha \frac{\left( \left| \frac{du}{dt} \right| - \left| \frac{dv}{dt} \right| \right)^2}{\left| \frac{du}{dt} \right| \cdot \left| \frac{dv}{dt} \right|} \end{aligned} \quad (10)$$

ここで式 (10) の右辺第 2 項は  $\theta = \arctan \left( \frac{du/dt}{dv/dt} \right)$  に対して Fig. 2 のような値を与える。この形は弾性体の変形に伴って発生する内部エネルギーを近似的に表現したものと考えられる。

式 (9) においては、グラフ  $g$  が定まって始めて値が決定されるのであるが、この  $g$  を種々変えた中で最小となる値をもって  $S_1$  と  $S_2$  の距離とすることとしよう。すなわち

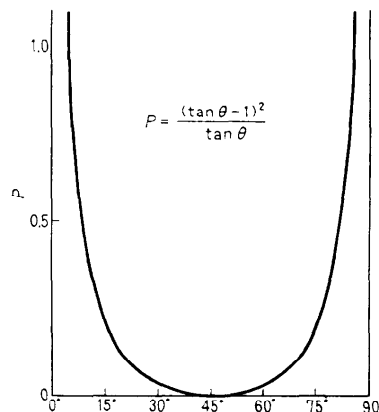


Fig. 2 Penalty for elasticity

$$d^*(S_1, S_2) \triangleq \min_{\mathbf{g}} d(S_1, S_2; \mathbf{g})$$

$$= \min_{\mathbf{g}} \int_0^1 \omega(S_1, S_2; \mathbf{g}) dt \quad (11)$$

とおくこととする。この距離に対して

- (1)  $d^*(S_1, S_1) = 0$
- (2)  $d^*(S_1, S_2) > 0 \quad S_1(x) \equiv S_2(x) \quad (12)$
- (3)  $d^*(S_1, S_2) = d^*(S_2, S_1)$

が成立する。ただしいわゆる三角不等式  $d^*(S_1, S_2) \leq d^*(S_1, S_3) + d^*(S_3, S_2)$  の成立は未だ確認出来ない。

以上で次元図形の距離が定義できたわけであるが、これをもとにすれば平面上の図形  $\{S_1^{(i)}(x)\}_{i=1}^L, \{S_2^{(i)}(x)\}_{i=1}^L$  に対しては

$$\sum_{i=1}^L d^*(S_1^{(i)}, S_2^{(i)}) \quad (13)$$

をもって距離とすれば良いわけである。ただしここで伸縮性を考慮してあるのは  $x$  方向（走査線方向）だけであるので、これに直交する方向への伸縮も考慮するためには、後に述べるような変更が必要である。

2.2 距離の計算法

式(11)で定義される距離を計算するには、動的計画法が有効である。そこで以下においてその基礎となる関数方程式を導びきだすこととする。まず準備として新しい記号を導入することとする。

$(u_1, v_1) \leq (u_2, v_2)$  のものにおいて

$$d_{(u_1, v_1)}^{(u_2, v_2)}(S_1, S_2; \mathbf{g}) \triangleq \int_{(u_1+v_1)/2}^{(u_2+v_2)/2} \omega(S_1, S_2; \mathbf{g}) dt \quad (14)$$

ただし

$$u \left( \frac{u_i + v_i}{2} \right) = u_i \quad i=1 \text{ または } 2 \quad (15)$$

$$v \left( \frac{u_i + v_i}{2} \right) = v_i$$

なる尺度を導入する。ここで式(15)は点  $(u_1, v_1)$  および  $(u_2, v_2)$  がグラフ  $\mathbf{g}$  上に乗っていることを意味している。これを今後  $(u_i, v_i) \in \mathbf{g} (i=1 \text{ または } 2)$  のように記述することとする。式(9), (14) から次の2つの関係式をうる事ができる。

$$d_{(0,0)}^{(1,1)}(S_1, S_2; \mathbf{g}) = d(S_1, S_2; \mathbf{g}) \quad (16)$$

$$d_{(u_1, v_1)}^{(u_1, v_1)}(S_1, S_2; \mathbf{g}) = 0 \quad (17)$$

また  $(u_1, v_1) \leq (u_3, v_3) \leq (u_2, v_2)$  および、 $(u_3, v_3) \in \mathbf{g}$  のもとにおいて、

$$d_{(u_1, v_1)}^{(u_2, v_2)}(S_1, S_2; \mathbf{g})$$

$$= d_{(u_1, v_1)}^{(u_3, v_3)}(S_1, S_2; \mathbf{g}) + d_{(u_3, v_3)}^{(u_2, v_2)}(S_1, S_2; \mathbf{g}) \quad (18)$$

が成立することとなる。

第2の新しい記号を次式で定義する。

$$d_{(u_1, v_1)}^{*(u_2, v_2)}(S_1, S_2) \triangleq \min_{\mathbf{g}} d_{(u_1, v_1)}^{(u_2, v_2)}(S_1, S_2; \mathbf{g}) \quad (19)$$

ただし minimum をとる範囲は、点  $(u_1, v_1)$  および  $(u_2, v_2)$  を通るすべてのグラフとする。

式(11), (19) より次の2つの式がえられる。

$$d_{(0,0)}^{*(1,1)}(S_1, S_2) = d^*(S_1, S_2) \quad (20)$$

$$d_{(u_1, v_1)}^{*(u_1, v_1)}(S_1, S_2) = 0 \quad (21)$$

記号  $d^*$  に関する関数方程式を導びくために新しい概念を導入する。区間  $\{(x, y); (u_1, v_1) \leq (x, y) \leq (u_2, v_2)\}$  の世襲境界 (hereditary boundary) とは、点  $(u_1, v_1)$  および  $(u_2, v_2)$  を通過する任意の連続・強単調増加関数とただ一度、しかもただ一点でのみ交わるような境界（単純弧）のことである\*。Fig. 3 には点  $A_1$  と  $A_2$  とを通る世襲境界の例がいくつか示してある。こうした世襲境界上の点によって点  $(u_1, v_1)$  と  $(u_2, v_2)$  とを通過する連続・強単調増加関数の全体は類別されることとなる。この事実と式(18), (19) とによって次の関係式をうる事ができる。

$$d_{(u_1, v_1)}^{*(u_2, v_2)}(S_1, S_2) = \min_{(u_3, v_3) \in \mathbf{h}} \{d_{(u_1, v_1)}^{*(u_3, v_3)}(S_1, S_2) + d_{(u_3, v_3)}^{*(u_2, v_2)}(S_1, S_2)\} \quad (22)$$

ただし  $\mathbf{h}$  は区間  $\{(x, y); (u_1, v_1) \leq (x, y) \leq (u_2, v_2)\}$  の世襲境界である。

今ここで世襲境界  $\mathbf{h}$  全体が点  $(u_2, v_2)$  のごく近くにあるものとすれば、式(22)は近似的に次のようにならわされることとなる。

$$d_{(u_1, v_1)}^{*(u_2, v_2)}(S_1, S_2)$$

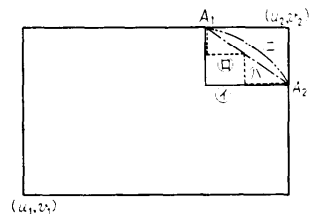


Fig. 3 Examples of hereditary boundary

\* 世襲境界は考えている区間を2つの部分にわけ、逆に区間を2つの部分 (A と B とする) にわけ、境界が任意の  $a \in A$  と任意の  $b \in B$  とに対し常に  $a \leq b (a \neq b)$  を成立させるならばこの境界は世襲境界となる。世襲境界に関連した部分は五十嵐彰氏におう。

$$= \min_{(u_3, v_3) \in \mathcal{H}} [d^{*(u_3, v_3)}(S_1, S_2) + z(u_3, v_3)\Delta t(u_3, v_3)] \quad (23)$$

ただし

$$\Delta t(u_3, v_3) = \frac{u_2 + v_2}{2} - \frac{u_3 + v_3}{2} \quad (24)$$

$$z(u_3, v_3) = \frac{|S_1(u_2) - S_2(v_2)| + |S_1(u_3) - S_2(v_3)|}{2} + \alpha \frac{((u_2 - u_3) - (v_2 - v_3))^2}{(u_2 - u_3)(v_2 - v_3)} \quad (25)$$

適当な格子点と適当な世襲境界  $\mathcal{H}$  とを定めた上でこの近似形を使えば

$d^{*(0,0)}(S_1, S_2) = 0$  から  $d^{*(u_2, v_2)}(S_1, S_2)$  をへて  $d^{*(1,1)}(S_1, S_2) = d^*(S_1, S_2)$  を求めることができる。

本論文においては  $N \times N$  の正方格子を採用し、しかも Fig. 3 ㊦のような世襲境界を採用した。ただし正方格子の境界上では、左または下からの接続を許した。ただしこの形の接続では式(25)の右辺第2項は  $\infty$  となるためこの項を取り払ったものを使った。この取り払いによって副次的に平行移動したパターン間の距離を0と出来ることとなり、良い効果を生ずることとなった。

式(23), (24), (25)に対応する格子点上での公式を与えることとしよう。まず以下の記号を導入しておく。

$$d^*(i, j) \triangleq d^{*((i-1)/(N-1), (j-1)/(N-1))}(S_1, S_2) \quad (26)$$

$$T(i, j) \triangleq \left| S_1\left(\frac{i-1}{N-1}\right) - S_2\left(\frac{j-1}{N-1}\right) \right| \quad (27)$$

また新しく最適グラフの局所記述のため  $R(i, j)$  なるものを導入することとする。この  $R(i, j)$  に入る記号は Fig. 4 に示すような当該格子点の計算を行なった時の最小ルート方向を示すためのものである。

式(23)等の具体形は、定数倍  $(8(N-1))$  等の非本質的変更の後以下に示すような形で与えられる。ただし Fig. 5 に示す7つの領域で公式が少しづつ異なって

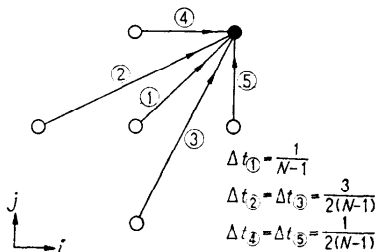


Fig. 4 Symbols for local route

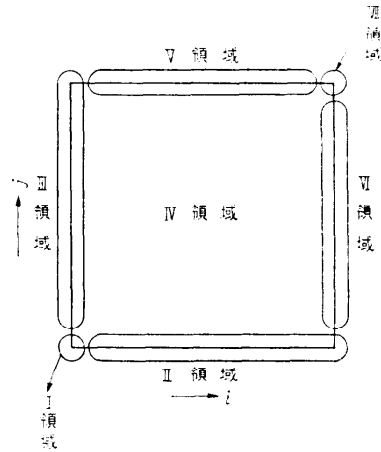


Fig. 5 Calculation domains

いる。それぞれ以下のようなになる。

領域 I (点 (1, 1))

$$d^*(1, 1) = 0 \quad (28)$$

$$R(1, 1) = 0 \quad (29)$$

領域 II

$$d^*(i, 1) = d^*(i-1, 1) + T(i-1, 1) + T(i, 1) \quad (30)$$

$$R(i, 1) = 4 \quad (31)$$

領域 III

$$d^*(1, j) = d^*(1, j-1) + T(1, j-1) + T(1, j) \quad (32)$$

$$R(1, j) = 5 \quad (33)$$

領域 IV

$$d^*(i, j) = \min \begin{cases} d^*(i-1, j-1) + 2\{T(i-1, j-1) + T(i, j)\} & \textcircled{1} \\ d^*(i-2, j-1) + 3\{T(i-2, j-1) + T(i, j) + \alpha\} & \textcircled{2} \\ d^*(i-1, j-2) + 3\{T(i-1, j-2) + T(i, j) + \alpha\} & \textcircled{3} \end{cases} \quad (34)$$

$$R(i, j) = 1, \textcircled{1}; 2, \textcircled{2}; 3, \textcircled{3} \quad (35)$$

領域 V

$$d^*(i, N) = \min \begin{cases} d^*(i-1, N-1) + 2\{T(i-1, N-1) + T(i, N)\} & \textcircled{1} \\ d^*(i-2, N-1) + 3\{T(i-2, N-1) + T(i, N) + \alpha\} & \textcircled{2} \\ d^*(i-1, N-2) + 3\{T(i-1, N-2) + T(i, N) + \alpha\} & \textcircled{3} \\ d^*(i-1, N) + T(i-1, N) + T(i, N) & \textcircled{4} \end{cases} \quad (36)$$

$$R(i, N)=1, \textcircled{1}; 2, \textcircled{2}; 3, \textcircled{3}; 4, \textcircled{4} \quad (37)$$

領域 VI

$$d^*(N, j) = \min \begin{cases} d^*(N-1, j-1) + 2\{T(N-1, j-1) + T(N, j)\} & \textcircled{1} \\ d^*(N-2, j-1) + 3\{T(N-2, j-1) + T(N, j) + \alpha\} & \textcircled{2} \\ d^*(N-1, j-2) + 3\{T(N-1, j-2) + T(N, j) + \alpha\} & \textcircled{3} \\ d^*(N, j-1) + T(N, j-1) + T(N, j) & \textcircled{5} \end{cases} \quad (38)$$

$$R(N, j)=1, \textcircled{1}; 2, \textcircled{2}; 3, \textcircled{3}; 5, \textcircled{5} \quad (39)$$

領域 VII (点 (N, N))

$$d^*(N, N) = \min \begin{cases} d^*(N-1, N-1) + 2\{T(N-1, N-1) + T(N, N)\} & \textcircled{1} \\ d^*(N-2, N-1) + 3\{T(N-2, N-1) + T(N, N) + \alpha\} & \textcircled{2} \\ d^*(N-1, N-2) + 3\{T(N-1, N-2) + T(N, N) + \alpha\} & \textcircled{3} \\ d^*(N-1, N) + T(N-1, N) + T(N, N) & \textcircled{4} \\ d^*(N, N-1) + T(N, N-1) + T(N, N) & \textcircled{5} \end{cases} \quad (40)$$

$$R(N, N)=1, \textcircled{1}; 2, \textcircled{2}; 3, \textcircled{3}; 4, \textcircled{4}; 5, \textcircled{5} \quad (41)$$

以上の公式の意味を明確に理解出来るように 8×8 格子点の場合における計算の流れの接続関係を Fig. 6

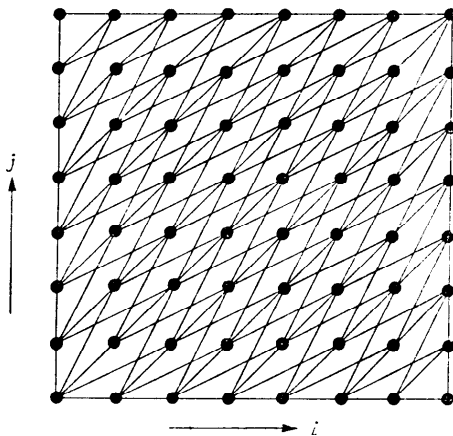


Fig. 6 Calculation flow in case of 8×8 lattice points

に示しておいた。

最適グラフの局所記述から最適グラフを求めるには、点 (N, N) から局所記述を逆にたどりながら以下のようになされる。

- ⑥  $i=N, v^*(N)=N$  とおく。
- ①  $i=1$  となるまで以下の操作を繰り返す。
  - (1)  $R(i, v^*(i))=1$  ならば  $v^*(i-1)=v^*(i)-1, i=i-1$  とおく。
  - (2)  $R(i, v^*(i))=2$  ならば  $v^*(i-1)=v^*(i)-1, v^*(i-2)=v^*(i-1), i=i-2$  とおく。
  - (3)  $R(i, v^*(i))=3$  ならば  $v^*(i-1)=v^*(i)-2, i=i-1$  とおく。
  - (4)  $R(i, v^*(i))=4$  ならば  $v^*(i-1)=v^*(i), i=i-1$  とおく。
  - (5)  $R(i, v^*(i))=5$  ならば  $v^*(i)=v^*(i)-1$  とおく。 (42)

この表現においては、最適グラフは  $\{(i, v^*(i))\}_{i=1}^N$  として表わされることとなる。

以上で距離の計算法と最適グラフの求め方が一応明らかになった。上述した計算法では式(26)のような min 計算が  $N \times N$  回実行されることとなっている。それゆえこのままの計算では平面上の図形間の距離計算には  $N \times N \times L$  回 min 計算が必要と言うことになっている。

しかし近接した二次元図形間には大きな相関があるので、これを利用して計算量を大幅に削減することが可能である。その方法とトータル計算量とを以下において調べることにしよう。

2.3 距離計算における計算量の削減法

上述した計算量の大幅削減可能な理由について述べておこう。Fig. 7 に描いてあるのは標準文字と入力文字とである。今仮りに  $L_1$  と  $K_1$  点とが対応点とみなされたものとする。この時当然  $L_2$  と  $K_2$  点とが対応点となるはずである。この場合  $L_1, L_2$  および  $K_1, K_2$  はともにあるストローク上の点であるから、 $L_1$  と

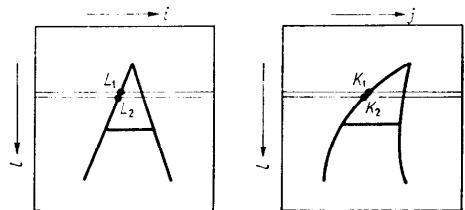


Fig. 7 Continuity of strokes

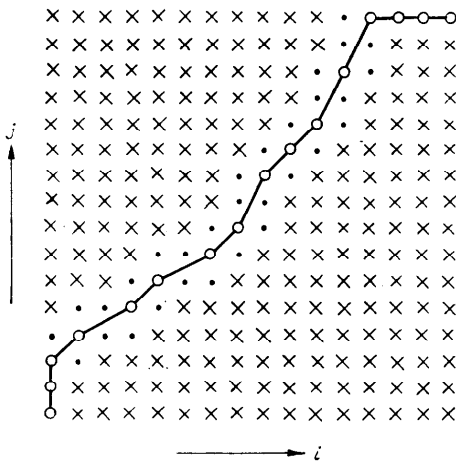


Fig. 8 Optimal graph and restricted calculation-domain of the next line

$K_1$  とを対応させたグラフと、 $L_2$  と  $K_2$  とを対応させるグラフとは、基本的に一行以内のずれしか生じないはずである。それゆえ  $L_1$  と  $K_1$  とを対応させた最適グラフが求められておれば、次の行の計算はこのグラフの左右上下  $\pm 1$  以内の範囲だけに限定してもなんら誤差を生じないはずである。Fig. 8 にある行の最適グラフ  $g^*$  とそれによって決定される次の行の限定計算領域とが示されている。この限定領域の具体的計算においては図のように各  $i$  に対する  $j$  の限定範囲  $J_1(i) \sim J_2(i)$  を決定する方式を採用した。これを求めるアルゴリズムは以下の形式であって、最適グラフの決定 (式(42)) と同時に行なうことができる。

- ①  $i=N, J_2(N)=N$  とおく。
- ①  $i=1$  となるまで以下の操作を繰り返す。
  - (1)  $R(i, v^*(i))=1$  ならば  
 $J_1(i)=v^*(i)-1$   
 $J_2(i-1)=v^*(i), i=i-1$  とおく。
  - (2)  $R(i, v^*(i))=2$  ならば  
 $J_1(i)=v^*(i)-1, J_1(i-1)=v^*(i)-1$   
 $J_2(i-1)=v^*(i), J_2(i-2)=v^*(i)$   
 $i=i-2$  とおく。
  - (3)  $R(i, v^*(i))=3$  ならば  
 $J_1(i)=v^*(i)-2$   
 $J_2(i-1)=v^*(i), i=i-1$  とおく。
  - (4)  $R(i, v^*(i))=4$  ならば  
 $J_1(i)=v^*(i),$   
 $J_2(i-1)=v^*(i), i=i-1$  とおく。

- (5)  $R(i, v^*(i))=5$  ならば  
 $J_1(i)=v^*(i)-1$  とおく。 (43)

以上のような計算領域の制限によって各一次元図形の計算量はほぼ  $3 \times N$  回の min 計算で十分であることとなる。ただし、しかるべき一次元図形だけは  $N \times N$  回の完全計算が必要である\*。それゆえトータルの計算量は

$$N \times N + 3N \times (L-1) = N \times (N + 3 \times (L-1)) \quad (44)$$

となる。これは格子点総数  $N \times L$  の約 4 倍と概算出来る。一方、中野-中田-中島の周辺分布によるマッチングにおいては、横向きだけで  $N \times N$  回の計算が必要である\*\*。よって本論文の手法との計算量の比は 4:1 と言うことになる。各行の計算をすべて行なう本論文の手法が、周辺分布一個のマッチングと同一オーダー計算量となっていることは注目し値する。このような結果を生じさせた最大の理由は、各行間の従属性をうまく活用したことにあるわけである。

2.4 パターン整形と完全な距離

上述してきた方法では、横方向の伸縮に対してしか対処することが出来ない。例をあげれば Fig. 9 の ㊦㊧なる各文字間の距離はほぼ 0 となるのであるが、㊦と㊧のように横向きのストロークの高さが異なっている場合、この間の距離は 0 とはならない。こうした場合にも有効なパターンマッチング法を構成するには、上述の方法を横向きと縦向きに 2 度適用することとすればよい。具体的には最初横向きのマッチングを行なって最適グラフを求め、このグラフによって入力パターンを整形した後、今度は縦向きのマッチングを行なう。この縦向きのマッチングでえられた距離をもってパターン間の距離とすれば Fig. 9 の ㊦と㊧間の距離もほぼ 0 となるのである。



Fig. 9 Variation of letter A

\* しかるべき一行だけは  $N \times N$  のフル計算が必要なのであるが一般に文字の端は空白であるから最初の一行をこれにあてることが出来ない。実際には両パターンの黒部分が最初にあらわれた行をこれにあてる必要がある。

\*\* 中野らの方法では  $45^\circ$  線からある幅  $d$  だけの範囲において最適グラフを求めているが、これでは本論文のように任意の位置づれに対処することが出来ない。そこで両手法の比較を行なうのにこの点に關してだけは同一性能を課すこととした。

まず最初に入力パターンの整形方法について述べる。標準パターンを  $\{S_1^{(l)}(i)\}_{i=1, l=1}^{N, L}$  入力パターンを  $\{S_2^{(l)}(j)\}_{j=1, l=1}^{N, L}$  として横向き距離計算を行なう。この時えられた最適グラフを  $\{v^{*(l)}(i)\}_{i=1, l=1}^{N, L}$  とする。このグラフを使って  $S_2$  の整形パターン  $S_2'$  を次のように定める。

$$S_2'^{(l)}(i) = S_2^{(l)}(v^{*(l)}(i)) \quad \begin{matrix} i=1 \sim N \\ l=1 \sim L \end{matrix} \quad (45)$$

この整形パターン  $S_2'$  と標準パターン  $S_1$  とを縦向きにマッチングをとることによって  $S_2', S_1$  間の距離が計算される。この距離をもって入力パターン  $S_2$  と

標準パターン  $S_1$  の距離としようと言うのである。

### 3. コンピューターシミュレーション

計算機を使って簡単なシミュレーションを行なったので報告しておく。標準パターン、入力パターンともに  $20 \times 20$  の格子点表示とし、大きさ、太さ、位置(上端および左端を所定位置におく)の正規化は完了しているものと仮定した。このうち問題となるのは、太さの正規化であろうが、これは Wave Propagation method<sup>4)</sup> を使うことによって実行出来ると考えている。標準パターンは A, B, C の3個で Fig. 10 の最

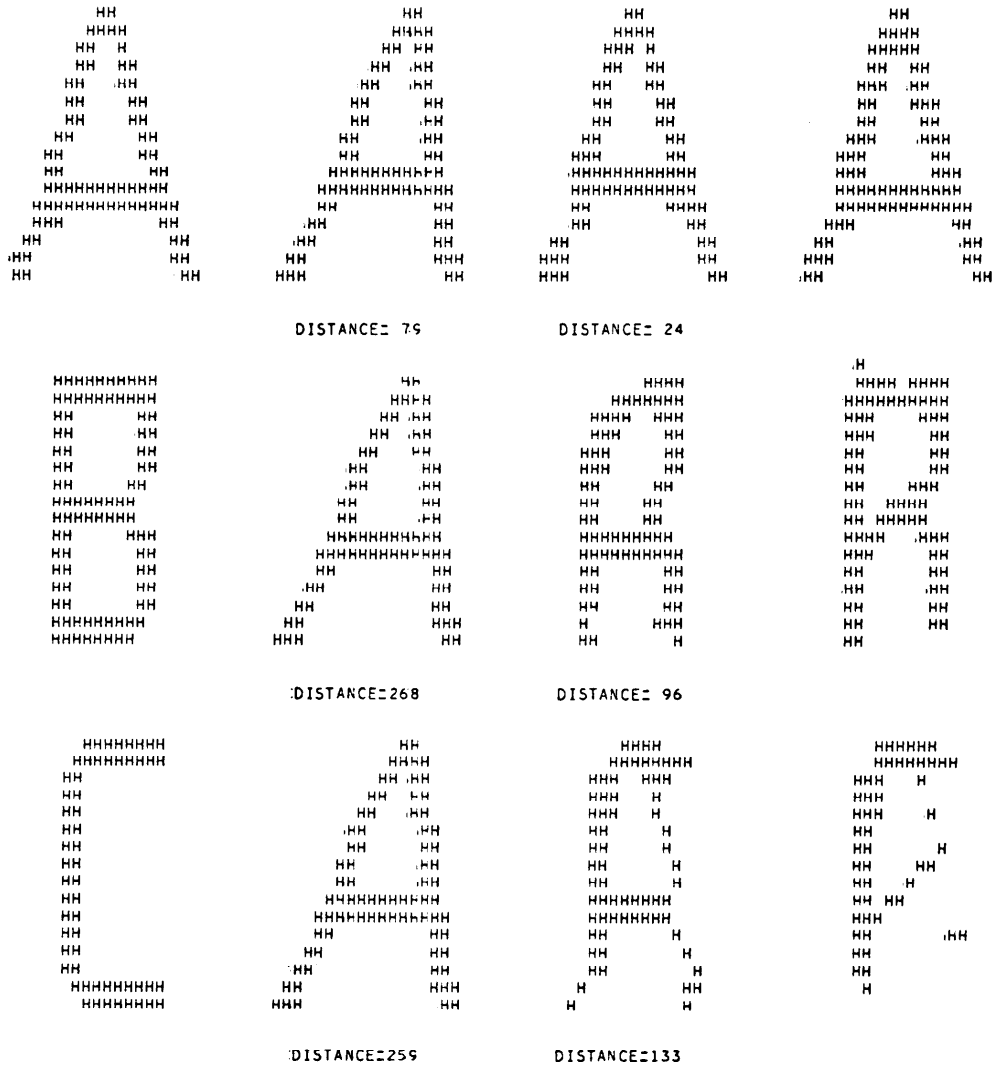


Fig. 10 Results of computer simulation

左端に縦に並べてある。入力パターンは左から2列目に並べてあるもので3つとも同一パターンである。第3列目のものは入力パターンの第1整形パターンである。第4列目のものは入力パターンの第2整形パターンである。第2列と第3列のパターンの下に印刷されている数値は、それぞれその上のパターンと最左端の標準パターンとの距離  $d^*$  である。この数値の大小比較から入力パターンは文字  $A$  であることが判定出来るのである。

本シミュレーションは全体で TOSBAC-5600 を使用して3.6秒かかった。それゆえ文字対あたり約1.2秒の計算時間が必要であることとなる。

#### 4. むすび

迫江-千葉の音声認識手法を検討してこれに2つのアイデアを導入することによって文字図形のマッチング法を構成した。第一のアイデアは2次元図形の各近接走査線間の相関の利用である。この結果パターンマッチングのための計算量を総格子点数に比例するオーダーにまで削減することが出来た。第二のアイデアはパターンを一種の弾性体であるとする弾性体モデルの導入である。この考え方にしたがって、歪みが最小となるマッチングを決定するための距離尺度を構成した。

本手法は、手書き文字のための最初のパターンマッチング法である。これまでパターンマッチング法は印刷文字に対してしか有効でないと考えられてきたが、この考え方は正しくなかったのである。本論文では、シミュレーションは英字を念頭において実行されたが、本手法は手書きの漢字ひらがなまじり文の認識にも適用出来るものと考えている。ただしこの場合には最低  $40 \times 40$  の格子点を必要とするので、本論文での

シミュレーションの約4倍の時間(文字あたり約4.8秒)を必要とするはずである。

本論文においては、横向きおよび縦向きのマッチングを各一回適用した。しかしこのマッチングを交互に何回か繰り返すと、位相幾何学的に同一の文字、例えば  $A$  と  $R$ 、 $B$  と  $8$  を同一視させることが可能である。これらの文字を同一視出来る識別法としては、ストローク・アナリシス法が存在するが、本手法のような一様かつ解析的方法によってこのようなことが実行出来るということは面白いことである。

謝辞 オートマトン研究室の諸氏には種々有益な議論をいただいた記して感謝の印としたい。

#### 参考文献

- 1) 迫江, 千葉: 動的計画法を利用した音声の時間正規化に基づく連続単語認識, 日本音響学会誌 **29**, **9**, pp. 483~490 (1971)
- 2) 中野, 中田, 中島: 周辺分布とそのスペクトルによる漢字認識の改良, 電子通信学会論文誌, **57-D**, **1**, pp. 15~22 (1974)
- 3) 奥田, 田中, 笠井: 距離概念の拡張とその応用 電子通信学会・パターン認識と学習研究会資料 PRL 72-123 (1973-01)
- 4) 鹿野, 鳥脇, 福村: 濃淡図形を線図形に変換する一方法—Wave propagation method について—電子通信学会論文誌 **55-D**, **10**, pp. 668~675 (1972)
- 5) G. Levi and U. Montanari: A grey weighted skeleton, Information and Control **17**, pp. 62 (1970)
- 6) 森, 森, 山本: 場の効果法による特徴抽出—閉じ状態の抽出—電通学会論文誌 **57-D**, pp. 308 (1974)

(昭和49年5月17日受付)

(昭和49年9月17日再受付)