オンライン手書き文字認識アルゴリズム RAV（Reparameterized Angle Variations）

小林 充† 真崎晋哉† 宮本 修†
中川洋一† 小宮義光† 松本 隆†

入力がペンで行われるコンピュータ，ワードプロセッサ，電子手帳等では，オンライン手書き文字認識が重要な役割を演じつつある．本論文は，1文字内でのストロークの連結による続け字に対して非常に強力なオンライン手書き文字認識の新しいアルゴリズム Reparameterized Angle Variations（RAV）を提案し，初期的実験結果を報告する．提案アルゴリズムは，次のような特徴を持つ：（i）タブレットから得られるペントラジェクトリーは角度変動トラジェクトリーに変換され，単純かつ有効な resample を行っている，（ii）2つの文字間の距離はペン角度変動とペン up/down 情報を考慮した特別なものが用いられている，（iii）筆順変動に対応するための自動辞書作成アルゴリズムが用いられている．東京農工大中川研究室の主導により収集，作成されたオンライン文字パターンデータベース kuchibue_d-96-02（mdb0001～mdb0010）中の教育漢字881文字に対して平均認識率91.2%，3位までの累積認識率96.2%である．

An On-line Character Recognition Algorithm RAV (Reparameterized Angle Variations)

Mitsuru Kobayashi,† Shinya Masaki,† Osamu Miyamoto,†
Youitch Nakagawa,† Yoshimitsu Komiya†
and Takashi Matsumoto†

With the advent of popularity of pen input devices including PDA, PC, word processor, accurate on-line character recognition starts playing a vitally important role. This paper proposes a new algorithm for pen input on-line character recognition which is extremely robust against stroke connections (stroke number variations) while maintaining a reasonable degree of robustness against stroke order variations. The proposed algorithm RAV has several important features; (i) Raw data consisting of pen position trajectory is transformed into angle variation and resampled in a simple but very effective manner, (ii) A special distance function is proposed to evaluate distance between two characters taking into account the angle variations as well as pen up/down variations, and (iii) An automatic dictionary generation scheme is proposed in order to copy with stroke order variations. Against the kuchibue-database, recognition rate for 881 Kyoiku Kanji was 91.2% while cumulative top three recognition rate was 96.2%.

1. はじめに

1.1 目　　的

オンライン文字認識では筆記過程情報（たとえば各時刻における座標値，筆圧等）が得られる一方，それを用いることにより問題も発生する.「筆画（ストローク）数変動」と「筆順変動」である．これら2つの変動から同時に自由なアルゴリズムの開発はオンライン文字認識における大きなチャレンジである．本論文の目的は，筆画数変動に強力かつ，ある程度筆順変動にも対応可能なオンライン文字認識アルゴリズム RAV（Reparameterized Angle Variations）を提案し，東京農工大中川研究室データベース kuchibue_d-96-02 を用いた初期認識実験を報告することである．本論文で提案するアルゴリズムは3つの特徴を持つ：

(a) 時・空間データからの特徴抽出手法

タブレットから得られる時・空間情報から複雑な部分はサンプル点を多く，単純な直線的な部分はサンプル点を少なくして特徴抽出する．位置情報，角度情報そしてペンの up/down 情報を用いる．

† 早稲田大学理工学部電気電子情報工学科
Department of Electrical, Electronics and Computer Engineering, Waseda University

(b) 認識フェーズにおける距離関数

他の多くの手法と同様，本論文の提案手法では，データベースから辞書を作成し，辞書に格納されているパターンと入力パターンを工夫された距離関数を用いて計算して，その最小値を与えるパターンに対応する文字を候補とする．

認識に用いる距離関数は，位置に関する距離以外に，各筆画内の局所的長さ，およびペンの up/down 情報を「重み」とする角度差を用いている．

(c) 自動辞書生成機能

一般に，辞書をどのように作成するかはもう 1 つの大きな問題である．提案手法では，辞書作成フェーズにおいてもある評価関数を用意して新しい学習データを評価し，その値が閾値を超えたときそのパターンを辞書に加える．もちろん，この部分も自動的に行われる．

アルゴリズムの効果を評価する初期的な認識実験では教育漢字 881 文字を対象に，文脈処理をいっさい行わずに実施した．使用したデータは最近，東京農工大中川研究室が中心となり，我々も作成に参加した手書き文字データベース kuchibue_d-96-02 中の mdb0001 〜mdb0010 である．データベースは非常に自由に筆記されており，多くの筆画数および筆順の変動を含んでいるばかりでなく，変形も激しいものも多い．この中の教育漢字 881 文字に対して認識率 91.2%，3 位までの累積認識率 96.2%を得た．

2 章で認識アルゴリズムの詳細について述べ，3 章で辞書の作成手法および計算量の削減に関する工夫，4 章で実験結果を報告する．5 章は考察とこれからの展望で締めくくる．

1.2 関連する研究

オンライン文字認識に関する研究の流れは文献 1) に要領よくまとめられており，全体を見通すことができる．

大雑把に分類すると，オンライン文字認識アルゴリズムは「構造解析的手法」と「パターンマッチング的手法」に分けられ，後者はさらに，「特徴点集合利用型」と「トラジェクトリー利用型」[2),3)]に分類される．次節で提案する本論文のアルゴリズムには「トラジェクトリー利用型」に属する．当該分野に限ってもすでに膨大な研究成果があるので，手法として本論文に直接関連するもののみを概観する．まず文献 4) はペンの位置情報のみでなく筆圧情報も用いて DP マッチングを遂行している点で次章に詳述する本論文の手法と関連するが，本論文で提案している角度に注目した圧縮とストローク長に関するペナルティは，少なくとも陽には入っていないと思われる．次に文献 5) はペンの位置と角度情報を用いて DP マッチングを用いているが，シャローバックトラックと呼ばれる対応点抽出がポイントであると思われる．また，文献 6) は，ペンの方向と方向変化を特徴量とし，前者が筆順，筆画数の変動に強いこと，一方，後者が字形変動に強いことに留意し，オフライン，オンライン両特徴融合型手法を提案している．

本論文は，これまで行ってきた口答発表等[7)〜10)]を正式論文としてまとめたものである．本論文で提案する手法はもちろん，オンライン情報のみを用いている．

2. アルゴリズム

2.1 デ ー タ

タブレットから得られるデータは 2 次元の位置情報 $x^1(t_i)$，$x^2(t_i)$ とストロークの終点か否かを判別する情報 $p(t_i)$ を含んでおり，ストロークの終点のときを $p(t_i)=1$，そうでないときを $p(t_i)=0$ とする．これを

$$(x^1(t_i), x^2(t_i), p(t_i)) \in R^2 \times \{0,1\} \quad i=0,1,\ldots,M \tag{1}$$

と書く．時間 t で順序付けられているのでこれをトラジェクトリー（軌跡）と呼ぶことは自然であろう．

以下に述べる我々の特徴抽出アルゴリズムは単純であるがきわめて有効である．今，

$$\bar{x} := ((\bar{x}(t_0), p(t_0)), \ldots, (\bar{x}(t_M), p(t_M)))$$

とする．

2.2 データ圧縮（resampling）

タブレットから得られるデータから後の認識アルゴリズムに相応しい特徴を抽出し，なおかつできる限りデータの圧縮を行いたい．ペンが “up” であるか “down” あるかに応じて異なる操作を施す．

【I. ペンが “down” のとき】

つまり，

- $p(t_0)=p(t_1)=\cdots=p(t_{D-1})=0$
- $p(t_D)=1$

である場合を考える．もし $D<2$ であればデータ圧縮は行わない．$D\geq 2$ とし，θ^* を以下に述べる閾値としておく．

Step1（**図 1** (a) 参照） データ $\bar{x}(t_0), \bar{x}(t_1), \bar{x}(t_2)$ に対して

$$\Delta\theta_i := \angle(\bar{x}(t_2)-\bar{x}(t_0), \bar{x}(t_i)-\bar{x}(t_{i-1})), \quad i=1,2 \tag{2}$$

とおく．ここに右辺は $\bar{x}(t_2)-\bar{x}(t_0)$ を基準としたときの $\bar{x}(t_2)-\bar{x}(t_0)$ と $\bar{x}(t_i)-\bar{x}(t_{i-1})$ のなす角度でもある．もし

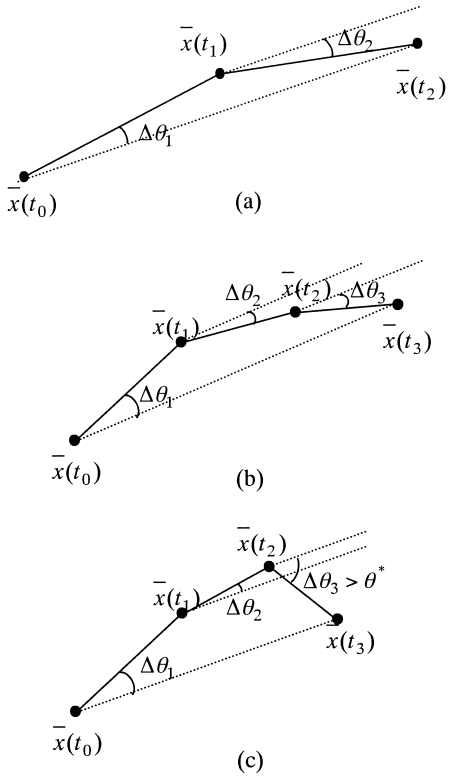


図 1 データ圧縮
Fig. 1 Data compression.

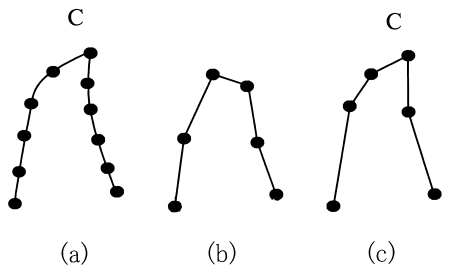


図 2 鋭い角度情報の保存
Fig. 2 The compression algorithm preserves sharp.

$$|\Delta\theta_i| < \theta^*,\ i = 1, 2 \tag{3}$$

であれば Step2 へ行く．

そうでなければ $\bar{x}(t_0), \bar{x}(t_1), \bar{x}(t_2)$ にデータ圧縮は施さず，$t_0 := t_2$ として Step1 へ．

Step2（**図 2** (b) 参照） 4 つのデータ $\bar{x}(t_0), \ldots, \bar{x}(t_3)$ に対して

$$\Delta\theta_i := \angle(\bar{x}(t_3) - \bar{x}(t_0), \bar{x}(t_i) - \bar{x}(t_{i-1})),$$
$$i = 1, 2, 3 \tag{4}$$

とおく．角度は $\bar{x}(t_3) - \bar{x}(t_0)$ を基準とする．もし

$$|\Delta\theta_i| < \theta^*,\ i = 1, 2, 3 \tag{5}$$

なら Step3 へ，そうでなければ圧縮データを

$$(\bar{x}(t_0), \bar{x}(t_2)) \tag{6}$$

で定義し，$\bar{x}(t_1)$ は捨てる（図 2 (c)）．$t_0 := t_3$ として Step1 へ．

Step *k* $k + 2$ 個のデータ $\bar{x}(t_0), \ldots, \bar{x}(t_{k+1})$ に対して

$$\Delta\theta_i := \angle(\bar{x}(t_{k+1}) - \bar{x}(t_0), \bar{x}(t_i) - \bar{x}(t_{i-1})),$$
$$i = 1, \ldots, k + 1 \tag{7}$$

を定義する．角度は $\bar{x}(t_{k+1}) - \bar{x}(t_0)$ を基準とする．もし

$$|\Delta\theta_i| < \theta^*,\ i = 1, \ldots, k + 1 \tag{8}$$

であれば Step $k+1$ へ，そうでなければ圧縮データを

$$(\bar{x}(t_0), \bar{x}(t_k)) \tag{9}$$

で定義し．$\bar{x}(t_1), \ldots, \bar{x}(t_{k-1})$ はすべて捨てる．$t_0 := t_k$ として Step1 へ．

【II. ペンが “up” のとき】

タブレットから得られる $P(t)$ の情報は，たとえば

- ペンが “up” のとき，$t_{i-1} \le t \le t_i$
- ペンが “down” のとき，$t < t_{i-1}$ and $t_i < t$

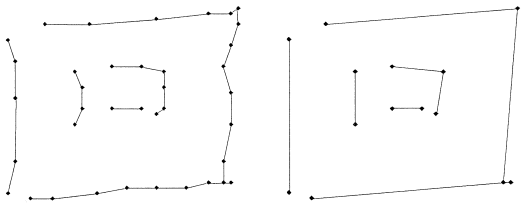
という形をしており，ペンの位置情報 $(x^1(t), x^2(t))$ は $t_{i-1} \le t \le t_i$ では得られず，$(x^1(t_{i-1}), x^2(t_{i-1}))$ と $(x^1(t_i), x^2(t_i))$ のみが得られる．したがって，この場合はデータ圧縮を行わない．つまり，ペンが “up” の部分を直線で結び，1 本の “up” ストロークで表現する．

上に詳述した圧縮（resampling）手法は一様 resampling でなく，トラジェクトリーの複雑さに適応したものである点が本質的である．

$(x^1(t_i), x^2(t_i))$ に対して，ペンが “down”，$i = 1, 2, \ldots, D$ のとき，「一様に “間引き” を行ってデータを圧縮してもよいのではないか？」という疑問は自然である．図 2 でそれがふさわしくないことがしばしばあることを示す．図 2 (a) は軌跡が鋭い角度で折れ曲がっている典型的な場合である．タブレットから得られるデータを後述の手法と同様に固定点数で間引きを行い圧縮すると，たとえば図 2 (b) のようになり，角点 C が失われてしまう．一方，上で述べた我々の圧縮を行うと，図 2 (c) のように角点 C が保存され鋭い角度情報が完全に保存される．

圧縮率は θ^* の関数であり，$\theta^* = 10$ 度で 55%程度，$\theta^* = 45$ 度にすると 40%程度となり後の認識アルゴリズム遂行の際，計算負荷を軽減する．漢字の場合は 45 度位まで可能であるばかりでなく，タブレットから得られるデータより美しい字が得られることもしばしばである．**図 3** (a) はタブレットから得られるデータ，図 3 (b) は $\theta^* = 45$ 度の圧縮データである．

多くの圧縮・特徴抽出アルゴリズムでは，ストローク中点や OCR 的な文字の形状，重心等を圧縮に利用



(a) (b)

図 3 生データと圧縮データ
Fig. 3 Raw data and the compressed data.

するためストロークの入力が終わらなければ圧縮の処理が始められないものが多かった．しかし，提案アルゴリズムでは，入力と同時に情報を取り込み自動的に角度変化や長さ等の特徴をとらえ圧縮を進めていく．

2.3 特徴抽出

圧縮されたデータも同じ記号

$$\bar{x} := ((\bar{x}(t_0), p(t_o)), \ldots, (\bar{x}(t_M), p(t_M))) \quad (10)$$

を用いる．特に混乱はない．$i = 1, 2, \ldots, M$ に対して，

$$\bar{\theta}_i := \tan^{-1} \frac{x^2(t_i) - x^2(t_{i-1})}{x^1(t_i) - x^1(t_{i-1})}, -\frac{\pi}{2} \le \bar{\theta}_i \le \frac{\pi}{2} \quad (11)$$

とし，

- $x^1(t_i) - x^1(t_{i-1}) \ge 0$ のとき
  $\theta_i := \bar{\theta}_i$
- $x^1(t_i) - x^1(t_{i-1}) < 0$ のとき

$$\theta_i := \begin{cases} \pi - \bar{\theta}_i (x^2(t_i) - x^2(t_{i-1}) \ge 0) \\ \bar{\theta}_i - \pi (x^2(t_i) - x^2(t_{i-1}) < 0) \end{cases} \quad (12)$$

とする．次に，

$$\Delta \bar{f}_i = \sqrt{(x^1(t_i) - (x^1(t_{i-1}))^2 + (x^2(t_i) - (x^2(t_{i-1}))^2} \quad (13)$$

$$\Delta f_i := \Delta \bar{f}_i \Big/ \sum_{j=1}^{M} \Delta \bar{f}_j \quad (14)$$

とおく．これは scale invariance を得るためである．そして

$$(\theta_i, \Delta f_i, p_i) \in R^2 \times \{0, 1\} \quad (15)$$

を特徴量とする．ただし，

$$p_i := p(t_{i-1})$$

である．θ_i は時刻 t_i における角度変動（Angle Variation）であり，Δf_i は正規化された局所弧長（Local Arc Length）である．

2.4 角度距離基準

入力のトラジェクトリーを

$$(x^1(t_i), x^2(t_i), p(t_i)),\ i = 1, \ldots, M$$

とする．他の多くのアルゴリズムと同様，我々のアルゴリズムも登録された標準パターン

$$(y^1(t_j), y^2(t_j), q(t_j)),\ j = 1, \ldots, N$$

と入力トラジェクトリーから抽出された特徴との距離を計算する．このようなアルゴリズムでは 2 つの特徴ベクトル間の距離関数の性能が決定的役割を演じる．今，

$$\bar{x} := ((\bar{x}(t_0), p(t_0)), \ldots, (\bar{x}(t_M), p(t_M))) \quad (16)$$

$$\bar{y} := ((\bar{y}(t_0), q(t_0), \ldots, (\bar{y}(t_N), q(t_N))) \quad (17)$$

とし，$(\theta_i, \Delta f_i, p_i)$，$(\eta_j, \Delta g_j, q_j)$ を前述の特徴ベクトルとするとき，これら 2 つのトラジェクトリー間の各時刻での距離を

$$|\theta_i - \eta_j| d(p_i, q_j) \rho(\Delta f_i, \Delta g_j) \quad (18)$$

で定義する．ここに $d(p_i, q_j)$ はペンの up/down 情報を考慮するための重み，そして $\rho(\Delta f_i, \Delta g_j)$ はトラジェクトリーの局所的長さであり，これらについては後述する．各時刻における距離関数ができたので，式 (16) と式 (17) 全体の間の距離を定義するには i，j に関して総和をとればよい．しかし，一般に $M \ne N$ なので subindex i_k，j_k を見つけ，$k = 1, \ldots, K$ までの和をとる必要がある．K は和のとり方に依存する．また，subindex i_k，j_k は

$$i_k \le i_{k+1},\ j_k \le j_{k+1} \quad (19)$$

を満たす必要がある．これは，トラジェクトリー $\bar{x}(t_i)$，$\bar{y}(t_i)$ には因果関係があるので，その順序を変えないためである．また，圧縮されたデータ (θ_i, p_i)，(η_j, q_j) はとばされないのが望ましいので

$$i_{k+1} \le i_k + 1,\ j_{k+1} \le j_k + 1 \quad (20)$$

の拘束を付ける．これらをまとめ，我々の距離は次式で与えられる．

$$D_1(\bar{x}, \bar{y}) := \min_{\substack{i_k \le i_{k+1} \le i_k + 1 \\ j_k \le j_{k+1} \le j_k + 1}} \sum_{k=1}^{K} |\theta_{i_k} - \eta_{j_k}| d(p_{i_k}, q_{j_k}) \rho(\Delta f_{i_k}, \Delta g_{j_k}) \quad (21)$$

ただし

$$i_1 = j_1 = 1,\ i_K = M,\ j_K = N \quad (22)$$

は固定する．

式 (21) で与えられる距離基準を言葉でいえば角度差 $|\theta_i - \eta_i|$ の局所的重み $d(p_i, q_j)\rho(\Delta f_i, \Delta g_i)$ を付けた warping であり，この重みが重要な役割を演じる．また，「トラジェクトリー $\bar{x}(t_i)$，$\bar{y}(t_j)$ を適当に正規化し，その差をユークリッド距離 $\|\bar{x}(t_i) - \bar{y}(t_j)\|$ ではかってはなぜいけないのか？」という疑問は自然に起きる．これには**図 4** によって簡単に答えることができ

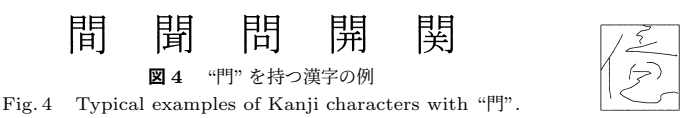


図 4 “門” を持つ漢字の例
Fig. 4 Typical examples of Kanji characters with “門”.

る．この図はすべて “門” を持つ漢字であるが，手書きの場合，“内側” の部分文字は小さく書かれ，しかもあまり丁寧に書かれないことが多く，$\|\bar{x}(t_i) - \bar{y}(t_j)\|$ は “内側部分” に関しては貧しい誤差情報しか与えないことが多い．このような状況は随所に生じる．一方，式 (18) は各 “内側部分” に対しても角度差によって必要とされる情報を鋭くとらえることができる．

2.5 ストロークの対応づけ

式 (21) の最小化問題の性質から，Dynamic Programming[11]はふさわしい手法の 1 つであろう：

$$D_1(0,0) = 0$$

$$D_1(i_k, j_k) = \min \begin{cases} D_1(i_k-1, j_k-1) + \\ |\theta_{i_k} - \eta_{j_k}| d(p_{i_k}, q_{j_k}) \rho(\Delta f_{i_k}, \Delta g_{j_k}) \\ D_1(i_k-1, j_k) + \\ |\theta_{i_k} - \eta_{j_k}| d(p_{i_k}, q_{j_k}) \rho(\Delta f_{i_k}, 0) \\ D_1(i_k, j_k-1) + \\ |\theta_{i_k} - \eta_{j_k}| d(p_{i_k}, q_{j_k}) \rho(0, \Delta g_{j_k}) \end{cases}$$

$$D_1(i_K, j_K) = D_1(\bar{x}, \bar{y}) \qquad (23)$$

ここで $d(p_{i_k}, q_{j_k})$ はペンの up/down 情報を考慮するための重みであるが，続け字を構成するストロークを対応付けするためには非常に重要な要素となる．

実際にペンを握って書かれる文字の pen up/down は同一文字に対して大幅に異なる場合が多いし（続け書き），同一人物が同一の文字を書いても異なりうる．登録標準パターンが比較的丁寧に書かれた文字である場合，実際に使用されるとき入力される文字は，はるかに少ないストローク数（pen up の回数が少ない）であることが多い．**図 5** は 15 画の漢字が 7〜14 画で書かれた例であるが，珍しい例ではない．

このようなとき，$p_{i_k} = 1$（pen up）に対して $q_{j_k} = 0$（pen down）は続け字を考慮してそれほどのペナルティを課すべきではないが，逆の場合は大きなペナルティを課すべきであろう．その他についても各々考察が必要である．

$\rho(\cdot,\cdot)$ の目的は単一の特徴ベクトル，たとえば

$(\bar{x}(t_i), \bar{x}(t_{i+1}))$

が，複数のベクトル，

$(\bar{y}(t_j), \bar{y}(t_{j+1}), \ldots, \bar{y}(t_{j+m}))$

に “対応” することにペナルティを課すものである．距離関数 (2.18) の第 1，第 2 factor は各々，角度と pen up/pen down 情報に関するものであり，第 3 factor に初めて，長さに関する情報が間接的に含まれる点に注意したい．

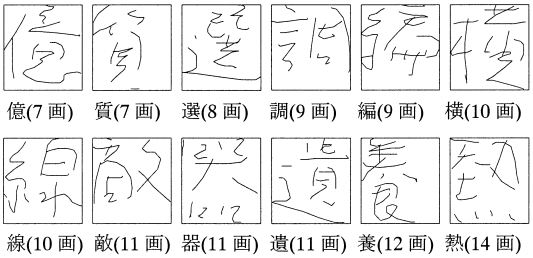


図 5 続け書きされた文字の例
Fig. 5 Examples of Kanji characters which has stroke connection.

2.6 位置距離基準

2.6.1 特徴ベクトル間の対応

角度情報中心の距離 $D_1(\bar{x}, \bar{y})$ の結果出力される上位数カテゴリの標準パターンに対しのみ入力文字と再度特徴ベクトルの角度距離基準による距離計算を行う．この計算は文字を数候補絞ることで，負荷も小さく時間的にも大きな問題にならない．この計算結果の局所的な最小化を与えるクリティカルパスを利用し，特徴ベクトル間の局所的な対応関係を調べる．

2.6.2 位置距離基準

ストローク特徴点間の相対距離を用いて各文字間の距離を算出する．入力パターンの i 番目の点を $\bar{x}(i) = (x_i^1, x_i^2)$，その点に対応する標準パターンの点を $\bar{y}(i) = (y_i^1, y_i^2)$ とする．また，文字の書き始めから書き終わりまでの全長をそれぞれ L_x と L_y とするとき，文字間の位置距離を

$$D_2(\bar{x}, \bar{y}) := \sum_{i=1}^{K^*} \sqrt{\left(\frac{x_i^1}{L_x} - \frac{y_i^1}{L_y}\right)^2 + \left(\frac{x_i^2}{L_x} - \frac{y_i^2}{L_y}\right)^2} \qquad (24)$$

で定義する．ここに K^* はストローク数である．

2.7 評価関数

最終認識結果は，ストロークの角度情報による距離と，点の位置情報による距離を考慮した

$$D_1(\bar{x}, \bar{y}) + \lambda D_2(\bar{x}, \bar{y}) \qquad (25)$$

の値が最も小さな文字とする．ただし，λ は経験値であり，4 章の実験では $D_1(\bar{x}, \bar{y})$ と $D_2(\bar{x}, \bar{y})$ の値をほぼ 1 対 1 の割合で考慮するような値としている．

3. 筆順変動と文字変形への対応

データベース kuchibue_d-96-02 をはじめ，人間の書く文字 1 つ 1 つにいくつかの筆順や変形が存在する．筆順違いや変形への対応策はまちまちであるが，

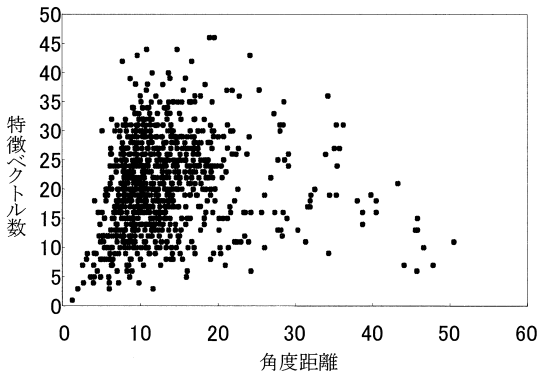


図 6 D_1 のプロット（教育漢字 881 文字）
Fig. 6 Plot of D_1 (881 characters of Kyoiku Kanji).

文字データをトラジェクトリーとして計算に取り入れるようなアルゴリズムでは標準パターンと異なる筆順の文字は異なる文字と見なされることが多い．そのため，筆順変動に関しては辞書への登録が多く用いられてきた．ここで問題になるのが，登録手法であって，たとえばある膨大な学習用データに対し，マニュアルで 1 文字ずつ登録するのは非効率的であり，やみくもに登録を行うと辞書容量を無駄に増やしてしまう．したがって，どのように登録文字を生成するかは大きな研究課題であるが，ここでは，1 つの文字に対し複数の標準パターンを辞書に登録すべきか否かを学習データから自動的に判断するアルゴリズムを提案する．

学習用の文字データの中から，現在の辞書に登録してある標準パターンと異なる筆順を持つ文字や大きく変形した文字は新たに辞書に登録したいが，新たに登録すべき文字であるか否かをいかにして自動的に判断するかが重要になる．以下に述べる $D_1(\bar{x},\bar{y})$ と特徴ベクトル数の分布による手法は，筆順違いのすべてをカバーしているとはいえないものの，単純で比較的有効である．

登録済みの標準パターンに対して，学習データ中の対応する文字パターンとの角度距離を計算する．この値を入力パターンの特徴ベクトル数を縦軸にとって教育漢字 881 文字に対してプロットしたときの典型的な場合（ここでは kuchibue_d-96-02 の mdb0005 から教育漢字 881 文字を抜粋した）を示したのが**図 6** である．

全体的に右上がりの傾向を示し，筆順違いや大幅な変形に対応する点は右側に大きくはずれる傾向が強い．そのような点に対し適当な登録の閾値を設定し，これを超える点を持つ文字パターンを既存の辞書ファイルに追加登録していく．

表 1 実験の平均認識率
Table 1 Average recognition rates for experiment.

	認識率	3 位までの累積認識率	7 位までの累積認識率
実験 1	87.9%	94.0%	95.0%
実験 2	83.0%	92.9%	93.9%
実験 3	87.3%	93.6%	96.2%
実験 4	88.8%	93.4%	94.8%
実験 5	91.2%	94.7%	96.2%

4. 認識実験

4.1 実験用データベース kuchibue_d-96-02

学習，評価ともに本論文で用いている文字データはすべて，オンライン手書き文字パターンデータベース HANDS-NK-kuchibue_d-96-02[12]から用いた．したがって，本論文の対象とする文字は，枠あり，字体制限なしの文章形式で採集された文字である．当然この中には，画数変動，筆順変動，変形等を多く含んでいる．

4.2 アルゴリズムの評価

kuchibue_d-96-02 の mdb0001〜mdb0010 から教育漢字 881 文字に含まれるものだけ抜き出し，これを入力パターンデータ（881 × 10 文字）とする．

まず，2 章で述べた角度距離基準から計算される累積誤差の小さい標準パターンを順に 7 位まで選び出した．これは，認識率と計算量の兼ね合いから経験的に決めた値である．前述の入力パターンデータに対し，ここでの認識率は約 87%であり，第 7 位累積認識率は約 96%程度である．

初期辞書は独自に作成したものである．また，この実験では kuchibue_d-96-02 の mdb0011〜mdb0020 から教育漢字（881 種）を抜き出し，それぞれ最初の登場文字を取り上げ 881 文字のファイルを作り，この 10 データ 8810 文字を順に学習に用いた．

ここでは，角度距離 D_1 が閾値に対し著しく大きな値を持つ文字は，他の文字との予期せぬ誤認識が生じる可能性を考慮して登録しないように上限を設けた帯状の閾値を用いている．

アルゴリズムの効果を定量的に評価を行うため，次の 5 つの実験を行った．

［実験 1］up/down パラメータ $d(p,q)$ を考慮しない．
［実験 2］Local Arc Length $\rho(\Delta f,\Delta g)$ を考慮しない．
［実験 3］位置距離基準を考慮しない．
［実験 4］複数登録された辞書を用いない．
［実験 5］提案手法

これらのアルゴリズム，データを用いて認識実験を行った結果が**表 1** である．値はすべて 10 データの平

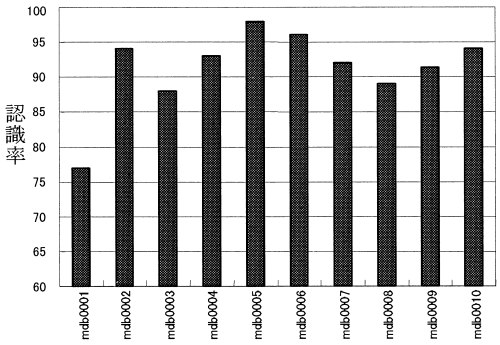


図 7 実験 5 における各データごとの認識率（教育漢字）
Fig. 7 Recognition rates of individual data on experiment 5.

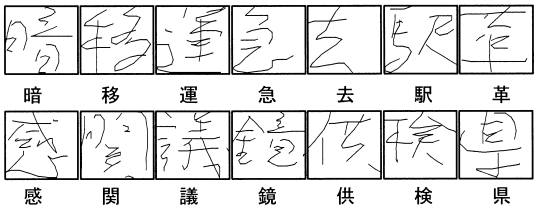


図 8 認識できた教育漢字の例
Fig. 8 Example of the successfully recongnized Kanji characters.

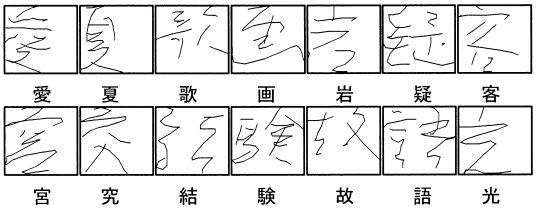


図 9 認識できなかった教育漢字の例
Fig. 9 Example of the Kanji characters which were not correctly recognized.

均である．また，実験 5 の提案手法による各データごとの認識率を**図 7** に示す．

また，本提案手法（実験 5）において認識できた文字およびできなかった文字の例を各々**図 8** および**図 9** に示す．

5. 考　　察

(1) 実験 1，実験 2 は本提案距離基準の続け字に対する有効性を示していると考えられる．ペンの up/down（実験 1）や Local Arc Length（実験 2）を考慮しない距離基準よりも，提案手法（実験 4）では 3 〜8 ポイントの認識率の向上が得られている．7 位までの累積認識率に大きな差がない点から，ストロークの連結による続け字に対し，ストロークを時間に関するトラジェクトリーとしてとらえその角度に注目して距離を与えたことが効果的であったと考えられる．また，Local Arc Length とペンの up/down に関する重みを付けながらマッチングをとる本提案手法の方が続け字に対し有効であることを，この実験結果から読み取ることができる．

(2) 実験 3 は本提案手法における角度距離基準から得られた文字候補の詳細識別の有効性を示すものである．この詳細識別は，文字間のストローク特徴点の相対位置距離に注目したアルゴリズム（2.6 節）であるが，この位置距離基準を用いることで，4 ポイントほどの認識率の向上がみられる．

(3) 実験 4 は提案手法の認識アルゴリズムにおいて，教育漢字 881 文字種に対し各 1 パターンのみ登録した 881 文字から成る辞書を用いて認識実験を行った結果である．一方，実験 5 で用いている辞書は，筆順違い文字に対していくつかのバリエーションを辞書に自動登録して作成したものである．しかし，すべての筆順違いを列挙して登録したものではない，という意味で組合せ的爆発は起きていないと考えられる．実験 4，5 の結果から，筆順違い等の変形に対応するために辞書への登録を用いることと提案アルゴリズムによる自動作成辞書の有効性が示される．辞書容量は，実験 4 で 219 KB であり，実験 5 で 345 KB である．登録による辞書容量増大は免れないが，本提案アルゴリズムでは，文字データのみを記述した簡素な辞書と，マッチング計算量削減のためのアルゴリズムを駆使し対処している．字種拡大にともなう筆順違いへの対応は引き続き重要検討課題である．

(4) 実験 5 は本提案手法による認識率を示している．使用した kuchibue データベースに含まれる文字は，筆記者に特に制限を設けていないので，程度としては普通から相当雑に書かれた文字が多い．図 8，9 には，人間が見ても分かりにくい文字も含まれるが，あえてこれらの文字を除かずに実験を行った．このような筆順変動や続け字，崩し字を多く含む文字データについても第 3 位累積認識率で 94.7%を得ているので，文脈処理等を併用すればさらに能力は向上すると思われる．

(5) 実験 5 の各データごとの認識率が図 7 に示されている．最も良い認識率は 98.3%（mdb0005）であり，最も悪かった認識率は 78.4%（mdb0001）であり，各データごとに認識率の変動は大きい．mdb0001 には，非常に多くの続け，崩し等の変形が含まれている．どの程度の変形にまで対応すべきかは，別の観点からの検討が必要と思われるが，軽〜中程度の変形を含む mdb0002〜mdb0010 に対して平均で 92.7%を得てい

ることも記しておきたい．

まとめると，提案アルゴリズム RAV の特徴は構造解析的手法と本質的に異なり，各文字全体を R^2 上のトラジェクトリーとしてとらえ，
(i) 効率の良いデータ圧縮と特徴抽出
(ii) 式 (18) による角度距離基準
(iii) 筆順違いや変形を自動登録した辞書
(iv) 計算量を削減した伸縮マッチング
を用いて対応しているのがポイントである．

6. 結　　言

本論文の目的は「筆画数変動」に頑健，かつ「筆順変動」にもある程度対応可能なオンライン手書き文字認識アルゴリズムを提案することであった．「筆画数変動」に対する頑健性は基本的には Dynamic Programming による time warping で解決できることはほぼ明らかであるが，問題は DP による最小化を行う評価関数（距離関数）がいかに問題をよくとらえているかである．前章の考察 (1)，(2)，(4)，は提案距離関数が比較的健全であることを示すものと考えられる．「筆順変動」への対応は自動辞書生成アルゴリズムで行っている．前章考察 (3) は，提案手法が機能していることを示していると考えられる．

これからの課題は
(A) 3 章で報告した実験は汎用性の検討といってよいが，個人性の検討も考慮に値する．すなわち，ある特定の筆者のみが用いることを前提とし，その個人の文字を標準パターンとする場合，提案アルゴリズムはきわめて強力である．これをどのような形で客観的に示すかを検討すること．
(B) 登録標準パターンに対する exhaustive search を，階層化や辞書の大分類，stack DP 等で改善すること．
(C) 認識対象文字をひらがな，カタカナ，漢字（JIS 第 1 水準），記号等に広げること．等であり，現在鋭意検討中である．

謝辞　多くの貴重なアドバイスをいただいた吉村ミツ先生（中部大）に感謝します．コメントをいただいた石垣一司氏，田中宏氏（富士通研究所），登内洋次郎氏（東芝），岡本正義氏（サンヨー），郡司圭子氏（日立製作所），吉井裕人氏（キャノン），高橋賢一朗氏（早稲田大），Roderick Koehle 氏（早稲田大），ディスカッションとデータベースに関して多大の便宜をおはかりいただいた中川正樹先生（東京農工大）に感謝します．

参 考 文 献

1) 中川正樹：ペンによる符号入力，ヒューマンインターフェースハンドブック，計測自動制御学会ヒューマンインターフェース部会編，オーム社（近日刊）（原稿は著者からのドラフト配布による）．
2) Tonoutchi, Y. and Kawamura, A.: An On-Line Japanese Character Recognition Method Using Lehngth-Based Stroke Correspondence Algorithm, *Proc. 4th ICDAR*, Vol.2, pp.633–636 (1997).
3) 趙鵬，佐藤幸男，吉村ミツ：オンライン走り書き文字認識における汎用辞書の作成，情報処理，Vol.34, No.3, pp.418–425 (1993).
4) 佐藤幸男，足立秀鋼：走り書き文字のオンライン認識，信学論（D），J68-D-12, pp.2116–2122 (1985).
5) 秋山勝彦，中川正樹：ストロークのつながりに寛容なオンライン手書き文字認識，画像の認識・理解シンポジウム (1994).
6) 岡本正義，山本和彦：方向特徴と方向変化特徴を用いたオンライン手書き文字認識，電学論，Vol.119-C, No.3, pp.358–366 (1999).
7) 小林　充，宮本　修，森　哲也，中川洋一，松本隆：Reparameterized Angle Variations を用いる on-line 手書き文字認識，信学技報，PRU94-121, pp.23–30 (1995).
8) 宮本　修，中川洋一，松本　隆：On-line 文字認識アルゴリズム Reparameterized Angle Variations を高速に実行するハードウェアボードについて，信学技報，PRU94-136, pp.49–56 (1995).
9) 真崎晋哉，小林　充，宮本　修，中川洋一，松本隆：オンライン手書き文字認識アルゴリズム RAV の筆順違い文字自動登録，信学技報，PRMU96-210, pp.135–142 (1997).
10) Masaki, S., Kobayashi, M., Miyamoto, O., Nakagawa, Y. and Matsumoto, T.: An On-Line Handwriting Character Recognition Algorithm RAV (Reparameterized Angle Variations), *Proc. 4th ICDAR*, Vol.2, pp.919–925 (1997).
11) Bellman, R.: *Dynamic Programming*, Princeton University Press, Princeton, New Jersey (1957).
12) 中川正樹，東山孝生，山中由紀子，澤田信一，レイ・バン・トゥー，秋山勝彦：文章形式字体制限なしオンライン手書き文字パターン収集と利用，信学技報，PRU95-110, pp43–48 (1995).

(平成 11 年 7 月 16 日受付)
(平成 12 年 7 月 5 日採録)

小林　充

1991 年早稲田大学理工学部電気工学科卒業．1993 年早稲田大学大学院理工学研究科修士（電気工学専攻）．在学中，文字認識等の研究に従事．

真崎　晋哉

1995 年早稲田大学理工学部電気工学科卒業．1997 年早稲田大学大学院理工学研究科修士（電気工学専攻）．（株）フジテレビジョン技術本部 技術局放送技術センター勤務．在学中，文字認識等の研究に従事．

宮本　修

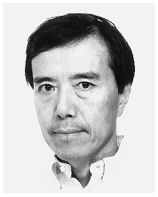
1993 年早稲田大学理工学部電気工学科卒業．1995 年早稲田大学大学院理工学研究科修士（電気工学専攻），同年川崎重工（株）勤務．1997 年よりイネーブルソフト（株）勤務．在学中，文字認識等の研究に従事．

中川　洋一

1994 年早稲田大学理工学部電気工学科卒業．1996 年早稲田大学大学院理工学研究科修士（電気工学専攻），同年（株）松下電器産業勤務．在学中，文字認識等の研究に従事．

小宮　義光

1998 年早稲田大学理工学部電気電子情報工学科卒業．現在，同大学院 2 年在籍．文字認識・署名照合等の研究に従事．



松本　隆（正会員）

1966 年早稲田大学理工学部電気工学科卒業．1970 年ハーバード大学大学院応用数学修士．1973 年工学博士（早稲田大学）．1977〜79 年カリフォルニア大学バークレー校電気工学・計算機科学研究員．1980 年早稲田大学理工学部電気工学科教授．現在早稲田大学理工学部電気電子情報工学科教授．分岐とカオス，非線形時系列予測，信号処理用集積回路設計等に興味を持つ．Circuit，Systems and Signal Processing 編集委員．テレビジョン学会次世代画像入力研究委員会（1995 年より）．電子情報通信学会非線形問題調査専門委員会委員長（1990，91 年度）．電気学会「カオス/数理と新技術」調査専門委員会委員長（1993 年より）．1993 年度日本神経回路学会論文賞．著書「Bifurcations」（Springer-Verlag），「カオス」（サイエンス社）等．IEEE（Fellow），APS，電気学会，テレビジョン学会，日本神経回路学会，計測自動制御学会等各委員．