

推薦論文

入力文字数制限のない平仮名空中手書き文字の分割法

重本 賢太郎¹ 清水 忠昭^{1,a)} 鈴木 慶¹ 吉村 宏紀¹ 松村 寿枝²

受付日 2015年9月2日, 採録日 2016年2月8日

概要:我々は, 空中手書き文字 (AHC) 入力システムのための自動的な文字分割手法の開発を行ってきた。これまでに提案した手法では, 平仮名の AHC について高い精度で文字分割に成功したが, 連続して入力可能な文字数に制限があった。本稿では, 同一領域に重ねて文字を入力する方法により入力文字数に制限のない文字分割手法を提案する。提案手法では, AHC のストロークを評価する 5つのストローク評価指標を学習したサポートベクタマシン (SVM) によりストローク判別して文字分割を行う。提案手法によるストローク判別は, 学習データでは, 移動ストローク 87.0%, 文字ストローク 96.5%の正解率となった。評価データに対しても, 移動ストローク 84.9%, 文字ストローク 96.1%の正解率を示した。試作した提案手法のデモ・システムについても紹介する。

キーワード: 空中手書き文字, 身振りインタフェース

Hiragana Aerial Handwriting Character Segmentation for an Unlimited of Number of Input Characters

KENTARO SHIGEMOTO¹ TADAAKI SHIMIZU^{1,a)} KEI SUZUKI¹
HIROKI YOSHIMURA¹ TOSHIE MATSUMURA²

Received: September 2, 2015, Accepted: February 8, 2016

Abstract: In this paper, we propose a segmentation method for an aerial handwriting character (AHC) input system. This work is an extension of a previously proposed hiragana AHC segmentation method that achieved high accuracy. However, its number of input characters was limited. In this paper, we propose a character segmentation method without such a limitation by overwriting characters on the same input area. Our method separates an AHC trajectory into characters by stroke distinction using a support vector machine (SVM) trained with five stroke evaluation indexes. The results of the evaluation experiments show that the detection accuracy was 80.9% for transition strokes and 98.1% for character strokes in the closed test, whereas it was 78.9% and 97.7%, respectively, in the open test. We also present a prototype system of the proposed method.

Keywords: aerial handwritten character, gesture interface

1. はじめに

人間同士の対話では, 意識しなくても“声”や“身振り手振り”といった「自然な動作」を交えてスムーズに意図を伝えあう。一方, 情報機器などに対してユーザの意図を

伝える場合, キーボードやマウスといった特別な入力装置を用いなければならない。このような入力装置の利用は初心者ユーザにとって難しく, 多くの学習時間が必要となる。この問題を解決するため, 人間にとって「自然な動作」を用いて入力を行うインタフェース NUI (Natural User Interface) の研究・開発が近年さかんに行われている。

NUIの中でも, 簡便な文字入力技術として“手書き文字認識”が広く研究されてきた。隠れマルコフモデルによる

本稿の内容は 2014 年 10 月の支部連合大会にて報告され, 支部長により情報処理学会論文誌ジャーナルへの掲載が推薦された論文である。

¹ 鳥取大学大学院工学研究科
Graduate School of Engineering, Tottori University, Tottori
680-8552, Japan

² 奈良工業高等専門学校
National Institute of Technology, Nara College, Yamato-
koriyama, Nara 639-1080, Japan

a) tadaaki@ecs.tottori-u.ac.jp

手書き文字認識の研究 [1], [2] やオンライン手書き文字認識の精度向上の研究 [3] など多数の研究事例がある。

文字は一般に紙上などに書かれるが、手元に紙や筆記具がない場合、指で空中をなぞることによって文字を伝えることがある。タッチパネルなどのデバイス上で前者を模したインタフェース技術が一般的な手書き文字認識であるのに対し、後者を模したインタフェース技術を“空中手書き文字認識”と呼ぶ。

センサ装着型の空中手書き文字認識手法として、ペン先についた加速度センサを用いる方法 [4], [5], ペン先についたカメラを用いる方法 [6], ヘッドマウントディスプレイ, IR センサ, Web カメラなどを装着する方法 [7] などが提案されている。センサ装着型的手法には、「いつでもどこでも」利用できるユビキタスを可能にするという利点があるが、センサデバイスを身につける必要があり、利用者にとって負担となる。一方、センサ非装着型的手法として、画像処理を軽減するために手に電球を持ちビデオカメラに向かって文字を書く方法 [8] や、複数の Web カメラによる 3D 処理を用いる方法 [9] などが提案されている。これらのセンサ非装着型には、ハンズフリーによる自然な文字入力の可能性がある。

空中手書き文字認識によるハンズフリーで自然な文字入力のためには、文字分割の問題を解決しなければならない。我々が紙に文字を書く際には、無意識に筆を上げる動作を行い文字と文字を切り離している。しかし、空中に文字を書く場合には、文字はひとつつながりの軌跡として連続して書かれる。文献 [8] の手法では文字の切り離しを指示する“スイッチ”となる電球をユーザに持たせることで、文献 [9] の手法では空中手書き文字の筆記動作としては不自然な“筆上げ”の動作を導入することで文字分割を行っている。文献 [10] の手法は、続けて書かれた数字を分割することなくテンプレートマッチングすることで文字分割を回避しているが、平仮名やアルファベットなど認識対象となる文字種の増加への対処に懸念がある。

この問題に対し、我々は平仮名の空中手書き文字を自動で分割する手法 [11] を提案した。この手法では、USB カメラで撮影された指を粒子フィルタ [12] で追跡して文字軌跡を取得し、空中に書かれたストロークの特徴から文字どうしをつなぐ“移動ストローク”を自動的に発見して文字分割を行う。この手法は平仮名に対して高い文字分割性能を示したが、USB カメラの撮影領域の制限から連続して入力可能な文字数に制限があった。

本稿では、先行研究の入力文字数制限を解消するために新たな文字分割手法を提案する。提案手法では同一領域に文字の筆記を連続して行うことで入力文字数制限の問題を解決した。評価実験の結果、提案手法によるストローク判別は、学習データでは、移動ストローク 87.0%, 文字ストローク 96.5%の正解率となった。また、評価データに対し

ても、移動ストローク 84.9%, 文字ストローク 96.1%の正解率を示した。

2. 本研究の目的

2.1 入力文字数制限の解消と指先マーカの廃止

我々は、文献 [11] において、空中手書き文字 (Aerial Handwritten Characters: AHC) による文字入力システムの実現を目指した平仮名 AHC の分割手法 (先行手法) を提案した。先行手法では、USB カメラで撮影した筆記者の指先を粒子フィルタによって追跡し、図 1 に示す“文字軌跡”を得る。文字軌跡を文字の 1 画とそれらをつなぐ画に分割したものを“ストローク”と呼ぶ。文字を構成するストロークを“文字ストローク”，文字と文字をつなぐストロークを“移動ストローク”と呼び、移動ストロークを発見することで文字軌跡を 1 文字ごとに分割する。

先行手法では、自然な文字入力動作を保証するために、以下の 2 つの条件を掲げた。

- スイッチなどの特別なデバイスを持たずハンズフリーで AHC 入力ができること
- AHC の筆記動作以外に文字を区切るための不自然な動作を導入しないこと

これらの条件を達成するために、AHC のストロークの性質だけから文字ストロークと移動ストロークを判別できるストローク評価指標を開発した。これにより、上記 2 つの条件を満たす文字分割法を開発できたが、USB カメラの撮影領域の制限から連続して入力可能な文字数に制限があった。

先行手法における入力文字数制限を解消するために、本稿では同一領域に連続して文字の筆記を行う手法を提案する。提案手法では、図 2 に示すように指先を追跡するセンサの撮影範囲や文字表示画面による入力文字数の制限なく文字を書き続けることができる。

先行手法では、画像処理を簡便にするために筆記者の指先にマーカとして赤い指サックを装着し、その赤色を粒子

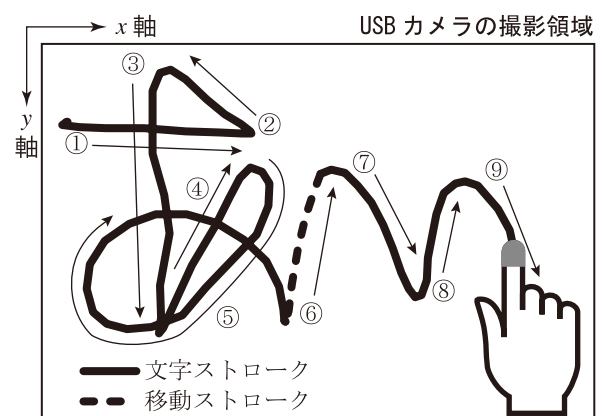


図 1 先行手法における空中手書き文字の例 [11]

Fig. 1 Example of AHC operation (previous system) [11].

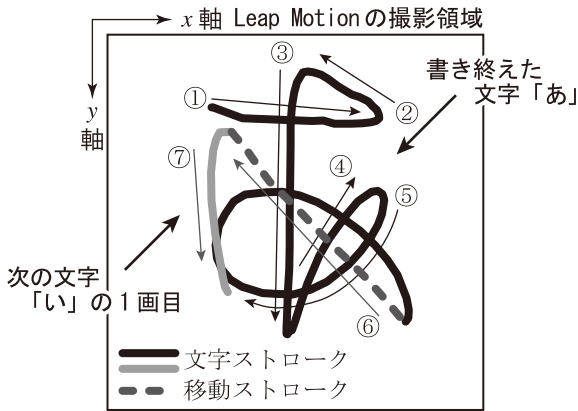


図 2 提案手法における空中手書き文字の例

Fig. 2 Example of AHC operation (proposed method).



図 3 AHC 入力システムの操作の様子

Fig. 3 AHC input system setup.

フィルタにより追跡することで指先の軌跡を取得していた。指先マーカを廃止して、より自然な操作を実現するために、提案手法の実装にあたり USB カメラに代えて赤外線センサの一種である Leap Motion [13] を採用した。図 3 に実装したシステムの操作の様子を示す。ユーザは“文字表示画面”で筆記中の文字軌跡を見ながら、Leap Motion の上に手をかざして AHC を筆記する。USB カメラや Leap Motion のように検出範囲が限られるデバイスを利用する場合、特に不慣れたユーザでは指先の検出範囲外れが発生しがちである。“文字表示画面”は、指先の検出範囲外れを防ぎ安定した筆記を助けるだけでなく、ユーザに筆記中の安心感を与えるために必要であると考え、提案手法の実装仕様に加えた。

2.2 提案手法を実現するための課題

提案手法の実現のために、主として以下に述べる 2 つの課題を解決する必要がある。

第 1 の課題は、入力デバイスの変更により生じたストローク分割の精度悪化の問題であり、3.2 節で詳述する。

第 2 の課題は、ストローク判別のリアルタイム性に関する問題である。先行手法では、2 文字を筆記した時点でストローク判別を行っていた。しかし、提案手法では、図 2 のように文字表示画面上で文字が重なるため、書き終えた

文字を次の文字の入力前に消去しなければならない。また、文字軌跡の画面表示を持たないシステムを想定した場合でも、入力文字数が多い応用では入力中の文字を何らかの方法でユーザにエコーバックする必要がある。そのようなシステムでは文字のエコーバックの遅延はシステムの使いにくさにつながるため、入力の 1 文字ごとに実時間で文字分割する必要がより高くなると考えられる。実時間での文字分割は、ストロークを書き終わると同時にストローク判別して文字分割を行う必要があることを意味しており、先行手法で有効性が示されたストローク評価指標を使うことができない。この問題を解決するための新しいストローク評価指標の提案については 3.3 節で述べる。

また、ストローク判別について先行手法ではベクトル量子化を応用した方法を用いていたが、提案手法では判別精度向上のために Support Vector Machine (SVM) [14] を導入した。これについては 3.4 節および 4 章で述べる。

3. 提案手法

図 4 に提案手法による文字軌跡の取得から文字分割までの処理の流れを示す。流れ図中の処理箱左側の番号は、処理箱の中に記載された処理を詳述した節の番号を示している。提案手法の実装には表 1 に示した一般的な PC を用いた。

3.1 指先の追跡と文字軌跡の取得

AHC の文字軌跡は、筆記中の指先を Leap Motion で追跡し取得する。Leap Motion には 2 つのカメラと赤外線 LED が搭載されており、本体上空の物体の左右 (x 軸)、上下 (y 軸)、前後 (z 軸) の三次元情報を検出する。メーカー公表では 1/100 [mm] の精度と最大約 300 [fps] のトラッキング速度で手の検出が可能である [13]。

提案手法の実装では、システムを構成する PC の処理能力の制限からトラッキング速度を約 $F = 50$ [fps] とした。また、Leap Motion SDK (1.0.8.7999) [15] を用いて、人差し指先が指す方向とディスプレイ画面の交点であるインターセクションポイント (ISP) を指先座標 $p_n = (x_n, y_n)$ として取得した。この際、ディスプレイ上の 1,000 [点] × 1,000 [点] の領域を文字筆記領域として設定した。得られた指先座標の並び $P = \{p_n\}$, $1 \leq n \leq N$ が文字軌跡であり、 N は文字軌跡の長さである。

3.2 文字軌跡をストロークに分割する

3.2.1 文字軌跡の筆記速度と筆記加速度および屈曲角度

文字軌跡をストロークに分割する手がかりとして、指先座標 p_n における筆記速度 v_n と筆記加速度 α_n および屈曲角度 θ_n を以下のように定義する。ここで、 $T = 1/F$ は指先座標のトラッキング間隔 [s] である。筆記加速度は筆記平面上での速度変化 $|v_n - v_{n-1}|$ で定義するのではなく、

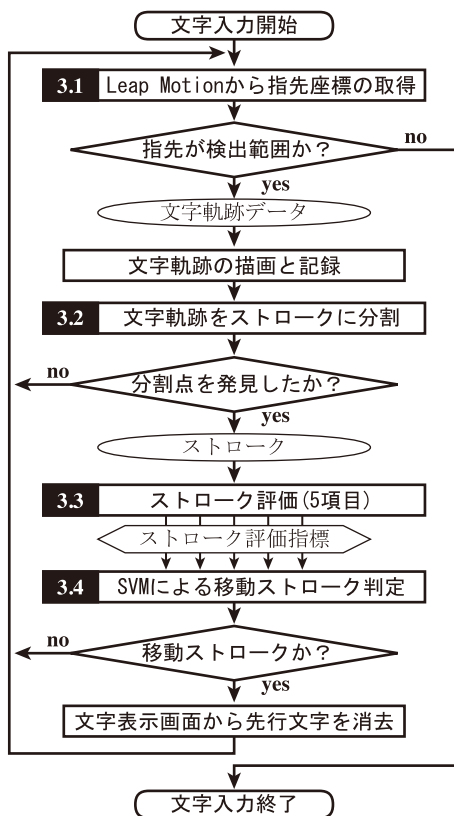


図 4 提案する AHC 分割手法の概要

Fig. 4 Block diagram of the proposed AHC segmentation method.

表 1 PC の仕様

Table 1 PC specifications.

CPU	Intel®Core™ i7 CPU K875 @2.93GHz
メモリ	8GB
OS	Windows7 Pro 64bit
ディスプレイ	FLATRON IPS226 (解像度 1920×1080)

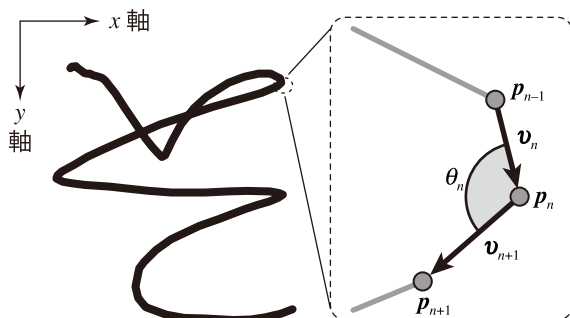


図 5 文字軌跡の筆記速度と屈曲角度

Fig. 5 Writing velocity and bending angle of the character trajectory.

軌跡に沿った指の動きの速さの変化をとらえるため式 (2) のように定義した. 図 5 に筆記速度と屈曲角度の関係を示す.

$$\mathbf{v}_n = (v_{xn}, v_{yn}) = \left(\frac{x_n - x_{n-1}}{T}, \frac{y_n - y_{n-1}}{T} \right) \quad (1)$$

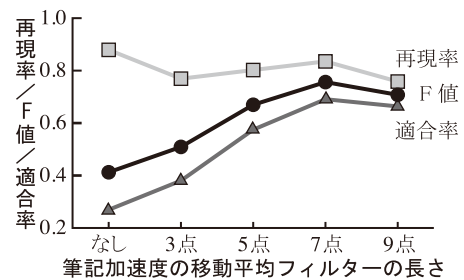


図 6 移動平均フィルターの長さが筆記加速度の変化によるストローク分割の性能に与える影響

Fig. 6 Effect of MA filter length on the performance of acceleration-based stroke separation.

$$\alpha_n = \frac{|\mathbf{v}_n| - |\mathbf{v}_{n-1}|}{T} \quad (2)$$

$$\theta_n = \cos^{-1} \frac{-\mathbf{v}_n \cdot \mathbf{v}_{n+1}}{|\mathbf{v}_n| |\mathbf{v}_{n+1}|} \quad (3)$$

3.2.2 ストローク分割法の評価方法

次項から述べるストローク分割法の予備評価のために, 以下の条件で簡易な評価実験を行った. 評価資料として, 20 歳代男性 1 名が「あ行」から「た行」を行ごとに連続して書いた 20 文字の文字軌跡資料を収録した. 収録した文字軌跡には, 1,342 点の指先座標が含まれ, うち 91 点がストロークの分割点である. ストローク分割法によって分割点検出を行った結果について, 分割点をポジティブデータとし, それ以外の指先座標をネガティブデータとして求めた再現率, 適合率, F 値により評価を行った. 分割点の正解は前後 1 点の揺れを許して評価した.

3.2.3 筆記加速度の変化によるストローク分割

AHC には, ストロークの終端でいったん減速し, 次のストロークの書き始めで加速する性質がある. この性質を利用して, 先行手法では移動平均フィルタで平滑化した筆記加速度が負から正に切り替わる点でストローク分割を行った.

筆記加速度の変化によるストローク分割法を, 筆記加速度に適用する移動平均フィルタの長さを変えて, 3.2.2 項の方法で評価した結果を図 6 に示す.

分割点検出の F 値は最良の場合でも 0.8 に達しなかった. この原因は, 指先追跡のデバイスの変更により指先座標の解像度とトラッキング速度が向上したためである. Leap Motion では指先のぶれまで精密に検出され, 移動平均フィルタがないと分割点以外の多くの点が分割点として検出されて適合率が低くなった. 一方, 指先のぶれの影響を避けるため移動平均フィルタを長くすると, 分割点での加速度変化まで失われ再現率が低下した.

3.2.4 文字軌跡の屈曲によるストローク分割

Leap Motion の採用によって先行手法に比べ指先座標の解像度とトラッキング速度が向上し, 文字軌跡の形状の精度は向上している. このため, 文字軌跡の形状に直接関わる屈曲点に着目したストローク分割法を新たに提案する.

表 2 文字軌跡の屈曲によるストローク分割の基準

Table 2 Criteria for stroke separation based on the bending point of the character trajectory.

C1	v_{xn} または v_{yn} の符号が変わる
C2	$\theta_n < \Theta$
C3	分割点候補を中心とする T_A 点で筆記加速度が負から正に代わる
C4	分割点検出後 T_D 点は分割点としない

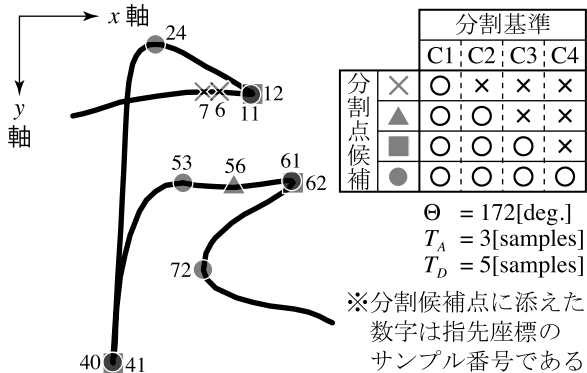


図 7 文字軌跡の屈曲によるストローク分割の例

Fig. 7 Example of stroke separation based on the bending point of the character trajectory.

まず、筆記速度 v_n の x 成分 v_{xn} または y 成分 v_{yn} の符号が変化するような屈曲点をストロークの分割点候補とする (基準 C1)。分割点候補には、多くの偽の候補が含まれる。特に、弧を描くストロークや直線のストロークを湾曲して描いてしまった場合には基準 C1 を満たす偽の候補が現れやすい。これらの偽の候補は屈曲角度 θ_n が大きい傾向があるため、角度制限 Θ を設け $\theta_n < \Theta$ となる分割点候補だけを残す (基準 C2)。さらに、指先のぶれなどによって生じる偽の候補を排除するため分割点候補で筆記加速度が負から正に変化している点だけを残す (基準 C3)。ただし、筆記加速度の変化は分割点から数点ずれる場合があるため、加速度範囲 T_A を設け、加速度候補点を中心とした前後 $\text{floor}(T_A/2)$ 点の範囲で加速度変化を調べるものとする。最後に、狭い範囲に分割点が続いて現れることを防ぐために不感期間 T_D を設け、分割点検出後 T_D 点の間の分割点候補を取り除く (基準 C4)。以上の4つのストローク分割基準を表 2 にまとめるとともに、図 7 に提案手法によるストローク分割の成功例を示す。図 7 中の \times \blacktriangle \blacksquare \bullet の記号は分割点候補であり、それぞれに添えられた数字は文字の書き始めから数えた指先座標番号である。図中右上の表に示すとおり、 \bullet 記号の分割点候補は C1 から C4 のすべての条件を満たしており正しい分割点である。 \times \blacktriangle \blacksquare 記号の分割点候補は C2 から C4 の条件で取り除かれた偽の候補である。

3.2.5 ストローク分割法のチューニング

文字軌跡の屈曲によるストローク分割法の調整パラメー

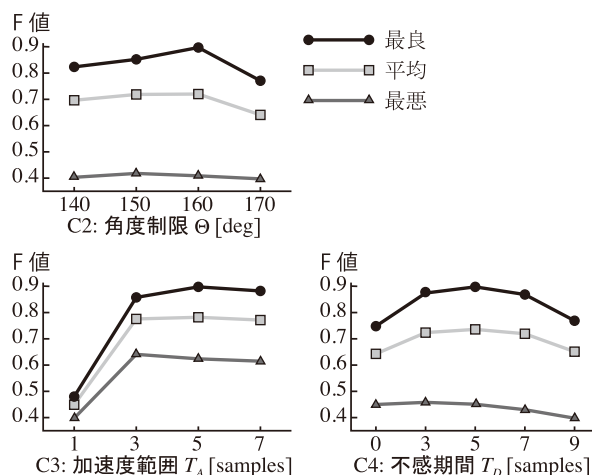


図 8 Θ , T_A , T_D が文字軌跡の屈曲によるストローク分割の性能に与える影響

Fig. 8 Effect of Θ , T_A and T_D on the performance of bending point-based stroke separation.

タである角度制限 Θ と加速度範囲 T_A および不感期間 T_D を実験的に決定した。 $\Theta = 140, 150, 160, 170$ [deg.] の 4 通り、 $T_A = 1, 3, 5, 7$ [点] の 4 通り、 $T_D = 0, 3, 5, 7, 9$ [点] の 5 通りの組合せ全 80 通りのパラメータ設定で、3.2.2 項の方法でストローク分割の評価実験を行った。

図 8 は、角度制限 Θ 、加速度範囲 T_A 、不感期間 T_D をそれぞれ 1 つの値に固定して他のパラメータを変えた場合の分割点検出の F 値の最良値、平均値、最悪値をまとめたグラフである。角度制限 Θ について、最良値では $\Theta = 160$ [deg.] で F 値が最大となった。加速度範囲 T_A については、 $T_A = 1$ [点] とすると、他のパラメータをどのように調整しても非常に低い F 値となった。最良値で見ると、 $T_A = 3$ [点] 以上で F 値の差がない。ストローク分割の結果を詳細に観察すると、C3 の基準でふるい落とされる分割点候補は少なく、C3 の効果は限定的であった。不感期間 T_D については 3 [点] から 7 [点] が適切である。

上記に従ってパラメータを設定した場合、文字軌跡の屈曲に着目したストローク分割法のストローク判定の F 値は 0.9 程度となり、筆記加速度の変化によるストローク分割法の最良の場合の F 値 0.8 程度と比べて非常に良い結果となった。

3.3 ストローク評価指標

先行手法では、5 つのストローク評価指標を設定し高い精度でストローク判別を行うことができた。また、5 つのストローク評価指標の中で、ストロークどうしの集まり方や位置関係を反映した指標が有効であることを明らかにした。しかし、本稿の提案手法では、文字表示画面上で複数の文字が重なることを防ぐため、ストロークを書き終わると同時にストローク判別を行って文字を分割する必要がある。このため、ストロークどうしの集まり方や位置関係を

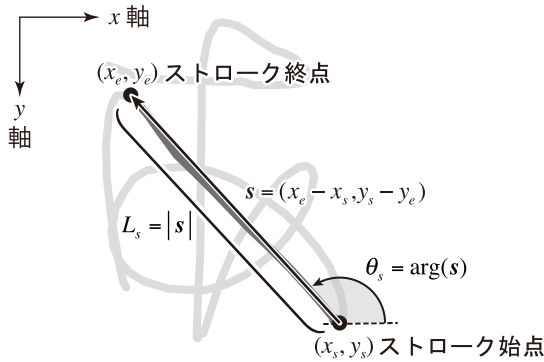


図 9 ストローク方向と長さ
Fig. 9 Direction and length of a stroke.

反映した指標を用いることができず、判別対象となるストロークの性質だけから計算できるストローク評価指標を新たに設定する必要がある。

提案手法では、上記のようなストローク判別に不利な点がある。しかし一方で、図 1 と図 2 を比較して気付くように提案手法では移動ストロークが筆記領域を右下から左上に横切る特徴的な動きをする場合が多い。このようなストロークの特徴をとらえることで、先行手法より高い精度で文字分割を行える可能性がある。以下の項では、本稿で新たに提案する 5 つのストローク評価指標について説明する。

3.3.1 ストロークの方向と長さ

移動ストロークは、右下から左上に長く横切る特徴的な動きをする場合が多い。このため、ストロークの方向と長さをストローク評価指標とする。これらを簡易に評価するため、図 9 に示すようにストロークの始点と終点をつなぐベクトルを s として、 s の偏角をストローク方向 θ_s (指標 1)、絶対値をストローク長 L_s (指標 2) と定義する。

$$\theta_s = \arg(s) \tag{4}$$

$$L_s = |s| \tag{5}$$

3.3.2 ストロークの始点方向と終点方向

移動ストロークの始点は、筆記領域の中心を基準として右下に、終点は左上になることが多い。このため、筆記領域の中心を基準とした始点と終点の方向をストローク評価指標とする。図 10 に示すように、筆記画面の中心と始点を結ぶベクトルを d_s 、終点を結ぶベクトルを d_e として、 d_s の偏角を始点方向 φ_s (指標 3)、 d_e の偏角を終点方向 φ_e (指標 4) と定義する。

$$\varphi_s = \arg(d_s) \tag{6}$$

$$\varphi_e = \arg(d_e) \tag{7}$$

3.3.3 次ストロークの方向

ストローク判別の精度を上げるために、ストロークどうし の関係を反映した指標を 1 つだけ加える。多くの文字の組合せを調べると、移動ストロークに似た長いストロークを持つ文字では、次のストロークは右方向に書かれやす

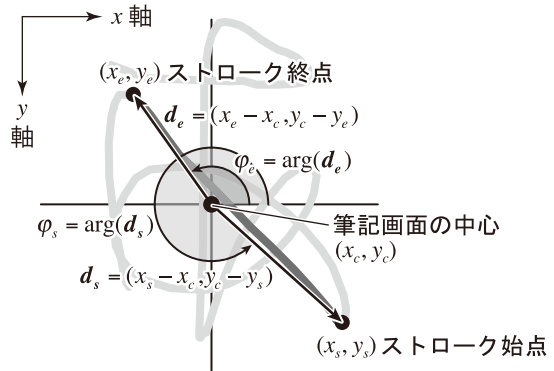


図 10 ストロークの始点方向と終点方向
Fig. 10 Initial direction and final direction of a stroke.

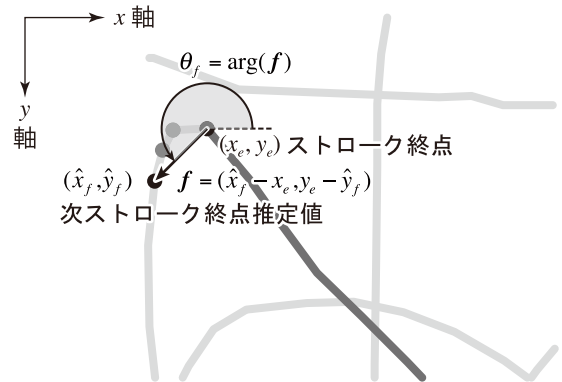


図 11 次ストローク方向
Fig. 11 Direction of the following stroke.

く、移動ストロークの次のストロークは左方向に書かれやすい。この性質を反映するために、判定中のストロークの次のストロークの方向をストローク評価指標とする。

先に述べたように、提案手法ではストロークを書き終わった時点でリアルタイムにストローク判別を行う必要があるため、次ストロークの終点の推定値として、次ストロークの 3 点目のサンプル点を用いる。図 11 に示すように、ストローク終点と次ストロークの 3 点目を結ぶベクトルを f とし、 f の偏角を次ストローク方向 θ_f (指標 5) と定義する。

$$\theta_f = \arg(f) \tag{8}$$

3.4 SVM によるストローク判別

提案手法ではストローク判別のために SVM を用いた。提案手法のストローク判別は、ストロークを文字ストロークか移動ストロークに分類する 2 分類問題なので、SVM の最も簡単な応用といえる。SVM の実装には OpenCV (2.4.4) [16] を用いた。SVM の学習の詳細については 4 章で述べる。

4. 提案手法の評価実験と結果

4.1 評価実験に用いる文字軌跡資料の収録

提案手法の評価実験のために、図 3 に示す実験環境で文

字軌跡資料を収録した。被験者はディスプレイに相対して椅子に座り、Leap Motion の上に手を伸ばし、ディスプレイに表示される文字軌跡を確認しながら筆記を行う。実験者は筆記される文字軌跡を確認して、文字の切れ目でキーボード操作を行い文字の分割および先行文字の消去を行う。文字分割を提案手法によらず実験者の手動で行った理由は、1) 提案手法の文字分割失敗が被験者の筆記動作に影響を与えないため、2) 提案手法の文字分割動作に筆記者が慣れて提案手法に合わせた筆記動作を学習しないようにするためである。

文字軌跡資料の収録では、① 実験者が被験者に対して文字入力方法の説明を行い、② 5分程度の練習筆記をさせた後、③ 文字軌跡資料の収録を行った。筆記する文字については実験説明の際に口頭と書面で指示し、筆記開始のタイミングは被験者に任せた。

被験者は、初めてシステムを使用する 20 代男子学生 8 名である。収録した文字軌跡は五十音順の平仮名計 46 文字であり、「あ」～「そ」、「た」～「ほ」、「ま」～「ん」の 3 回に分けて、休憩を挟みながら収録した。収録した文字軌跡資料に含まれる理想的なストローク数は、移動ストロークが $43[\text{本}] \times 8[\text{人}] = 344[\text{本}]$ 、文字ストロークが $179[\text{本}] \times 8[\text{人}] = 1,432[\text{本}]$ である。

4.2 ストローク分割の評価

提案した文字軌跡の屈曲によるストローク分割法を用い、3.2.4 項の結果に基づいて角度制限 $\Theta = 160[\text{deg.}]$ 、加速度範囲 $T_A = 7[\text{点}]$ 、不感期間 $T_D = 5[\text{点}]$ に設定して、4.1 節で説明した文字軌跡資料をストロークに分割した。

分割の失敗は、① 書き淀み、② 書き迷い、③ ストロークの分断、④ ストロークの連結の 4 種類に分類できる。① 書き淀みは、AHC の書き始めや書き終わりで起こりやすく、ごく短い余剰ストロークが生じる。② 書き迷いは、AHC の筆記中に筆記者が次の画を書く前に筆運びを迷ったために余剰ストロークが生じる。③ ストローク分断は、本来 1 本であるべきストロークが複数のストロークに分断される現象であり、ストロークが曲線で筆記途中で減速加速する場合に起こりやすい。④ ストローク連結は、ストロークの切れ目が発見できず、2 本のストロークがつながる現象である。先行手法では、連続して書かれた文字どうしが移動ストロークを介さずに接続してしまう“移動ストロークの消失”が発生していたが、提案手法では生じなかった。

①②③ではストローク数が増加し、④はストローク数が減少する。ストローク分割結果の詳細を表 3 に示す。

ストローク分割により得られたストローク数（“総数”）は、文字ストローク 1,792 本、移動ストローク 309 本、計 2,101 本である。表 3 中の“成功”は理想の位置で分割されたストロークの数である。“成功”の数に“④”により減少した数を足すと“理想本数”と一致する。“成功”の数に

表 3 AHC 軌跡から分割されたストロークの内訳

Table 3 Breakdown of stroke separation from AHC trajectories.

ストローク種別		文字(本)	移動(本)	全体
理想本数		1432	344	1776
分割	総数	1792	309	2101
	成功	1285(89.7%)	309(89.8%)	1594(89.8%)
結果	失敗	507	0	507
	④	147	35	182

“①②③”により増加したストローク数を加えると“総数”となる。

移動ストロークが①②③により余剰に細分された場合は、細分された中で最も長さの長いストロークを移動ストロークとし、他の余剰ストロークは文字ストロークとして数えている。このため表 3 で移動ストロークの“①②③”の欄が 0 となっている。このような取扱いの理由は以下に示すとおりである。

- 移動ストロークが複数に細分されても、そのうち 1 本でも移動ストロークと判定されれば文字分割が成功する。このため細分された移動ストロークを移動ストロークとして扱ってよい。
- 一方、短く細分された移動ストロークは移動ストロークの特徴が失われるため、次節で述べる SVM の学習に悪影響を与える可能性がある。このため、細分された移動ストロークのうち最長の 1 本のみを移動ストロークとして扱った。

表 3 によると、文字ストローク、移動ストロークともほぼ 90% の正解率となった。しかし、先行手法では 93～98% の正解率であったので、提案手法のストローク分割には改良の余地があると考えられる。特に、先行手法に比べて④のストローク連結が増加していることが性能の低下の原因である。

4.3 SVM によるストローク判別

4.3.1 SVM によるストローク判別の評価

4.2 節で得られたストロークを用いて、3.3 節で提案したストローク評価指標を用いた SVM によるストローク判別の性能評価実験を行った。SVM の実装には OpenCV を用い、カーネルをガウシアン動径基底関数 (RBF) カーネルに設定し、学習パラメータ γ はデフォルトの 1.0 とした。

SVM の学習データとして被験者 4 名分のストロークを、学習後の SVM の評価データとして残りの 4 名分のストロークを用いた。SVM には、ポジティブ・データ (POS データ) として移動ストロークのストローク評価指標を、ネガティブ・データ (NEG データ) として文字ストロークのストローク評価指標を与えた。

学習データと評価データとなる被験者の組合せをランダムに変えて学習と評価を 20 回試行した。学習と評価に用

表 4 SVM によるストローク判別精度

Table 4 Stroke classification discrimination precision of the SVM.

	学習データ			評価データ		
	適合率	再現率	F 値	適合率	再現率	F 値
平均	0.893	0.903	0.897	0.869	0.877	0.871
標準偏差	0.026	0.035	0.028	0.043	0.058	0.030

表 5 SVM によるストローク判別精度 (POS データ × 3)

Table 5 Stroke classification discrimination precision of the SVM (POS data × 3).

	学習データ			評価データ		
	適合率	再現率	F 値	適合率	再現率	F 値
平均	0.934	0.970	0.952	0.807	0.943	0.869
標準偏差	0.012	0.008	0.009	0.035	0.036	0.021

いるデータ数は被験者の組合せにより試行ごとにばらつくが、その平均は学習データの POS データが 154.2[本]、NEG データが 890.8[本]であり、評価データの POS データが 154.9[本]、NEG データが 901.2[本]である。

評価実験の結果として、表 4 に 20 回の試行における適合率、再現率、F 値の平均と標準偏差を示す。

評価実験の結果、学習データ、評価データともに 0.9 に近い F 値でストローク判別が行われている。また、F 値の標準偏差は 0.03 以下で SVM の学習が安定して行われていることを示している。

4.3.2 POS データ複製によるストローク判別性能の改善

AHC において文字ストロークに比べ移動ストロークが少ないことから、SVM の学習に用いる POS データと NEG データの数が不均衡になる。NEG データの数が多いため、特に学習後の SVM において再現率が低下しやすい。

この点を改善するために、4.3.1 項と同じ条件で学習データの POS データを複製して数を 3 倍に増やし、SVM の学習および評価を行う実験を試みた。

評価実験の結果として、表 5 に 20 回の試行における適合率、再現率、F 値の平均と標準偏差を示す。

学習データの POS データを増やしたことで、学習データに対しては適合率、再現率ともに向上し、F 値は 0.952 と非常に良い結果となった。評価データに対しては、適合率は低下したものの再現率は大幅に向上し、F 値はほぼ変わらない結果となった。F 値が同様であれば、再現率が高い方が文字分割に有利であると考えられる。また、表 5 に比べて表 6 の F 値の標準偏差が小さくなっており、SVM の学習がより安定して行っているといえる。

4.4 提案手法のシステム全体の評価

4.2 節でストローク分割の評価を行い、4.3 節で SVM によるストローク判別の評価を行った。この節では、両者をあわせた提案手法のシステム全体の評価を行う。

移動ストロークの正解率は、4.3.2 項の実験で得られた移

表 6 システム全体としての移動ストロークと文字ストロークの判別正解率

Table 6 Discrimination accuracy between transition and character strokes in the overall system.

	学習データ			評価データ		
	ストローク数	正解率	母数 (%)	ストローク数	正解率	母数 (%)
	正解			正解		
移動	448.7	516	87.0	146.1	172	84.9
文字	859.25	890.8	96.5	886.2	901.2	96.1

動ストロークの平均正解数を移動ストロークの理想本数を母数として除して求める。これによって、ストローク分割の失敗で失われた移動ストロークも考慮したシステム全体としての正解率が得られる。

一方、文字ストロークはストローク分割の失敗で失われたとしても文字分割に影響を与えない。反対に余剰に分割された文字ストロークが 1 つでも移動ストロークと判定されると間違った文字分割が行われる。このため、文字ストロークの正解率は、4.3.2 項の実験で得られた文字ストロークの平均正解数を文字ストロークの平均総数を母数として除して求める。

学習データと評価データに対する提案手法全体を通じての移動ストロークと文字ストロークの正解率を表 6 に示す。

表 6 によると、移動ストロークについては、学習データでは 87%、評価データでは 85% 程度の正解率となり、文字ストロークについては、いずれも 96% 程度の正解率を得た。評価データに対しては、移動ストローク、文字ストロークとも先行手法より高い正解率を得ることができた。また、学習データと評価データで正解率に大きく差が出なかったことは、SVM が未知データに対しても有効に働いていることを示している。

本研究では、2.2 節で述べた理由により文字分割の実時間性を重要視してストローク評価指標を定めた。このため各評価指標は次ストローク方向 θ_f を除いてストローク間の関係を反映していない。ストローク判別は逐次的なものとなっている。したがって、提案手法は文字列中のストロークどうしの関係を反映した全体的なストローク判別に比べて原理的に判別精度が低い。今後、最低限の実時間性を担保しながら文字列中のストロークどうしの関係を考慮した文字分割手法の可能性についても考えていきたい。

4.5 試作デモ・システムの動作の様子

提案手法の試作デモ・システムの動作の様子を、文字表示画面のキャプチャ画像によって図 12 に示す。文字表示画面は筆記中の文字が表示される筆記表示部と、それ以前に書かれた文字を表示する先行文字表示部からなる。① は平仮名「く」の筆記中の様子であり、② で次の文字「け」に移るための移動ストロークが書かれている。③ で「け」

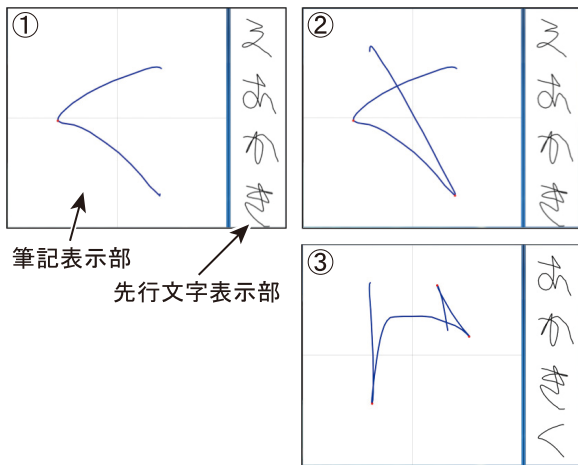


図 12 デモ・システムの動作例

Fig. 12 Example operation of the prototype.

が書かれ始めると、先行文字である「く」が消去され、先行文字表示部に転記される。②は「く」が消去される直前の状態であるが、移動ストロークの先にわずかに次の文字「け」の1画目が表示されている。これは、3.3.3項で述べた次ストローク方向 θ_f を求めるために次の文字「け」の1画目の最初の3点が必要なためである。

試作デモ・システムは平均 49.6 fps でリアルタイム動作することを確認している。文字の分割失敗時の手動補正などの機能を追加することで実用レベルの動作が可能である。本システムは、2014年9月26日、27日に開催された「とっとり産業フェスティバル 2014」[17]にデモ展示し、好評を得た。

5. おわりに

本稿では、先行研究における入力文字数制限を解消するために、同一領域に重ねて文字を入力する方法を提案した。その実現のために、新たなストローク評価指標を定義し、SVMによってストローク判別を行って文字分割を行う手法を提案した。また、指先マーカを廃止するためにシステムの実装に Leap Motion を採用した。評価実験の結果、提案したストローク判別法は未学習の評価データについて、移動ストロークで 84.9%、文字ストロークで 96.1%の正解率を示した。被験者が5分程度しか本システムを試用していないことを勘案すると、システムに対する学習をほとんど必要としない“自然な動作”での文字分割法として一定の有効性を示すことができたと考える。

参考文献

[1] 高橋賢一郎, 安田英史, 松本 隆: Hidden Markov Model を用いたオンライン手書き文字認識, 電気情報通信学会技術研究報告, PRMU96-211, pp.143–150 (1997).
 [2] 片山喜規, 内田誠一, 迫江博昭: 座標特徴と方向特徴の選択的利用に基づくオンライン文字認識 HMM, 電子情報通信学会論文誌, Vol.J91-D, No.8, pp.2112–2120 (2008).

[3] 小林 充, 真崎晋哉ほか: オンライン手書き文字認識アルゴリズム RAV, 情報処理学会論文誌, Vol.41, No.9, pp.2536–2544 (2000).
 [4] 中井 満, 米沢久光: 加速度センサを用いた空中手書き文字認識, FIT 講演論文集, Vol.8, No.3, pp.133–134 (2009).
 [5] 中井 満, 草島広志: 筆記の向きや筆記具の持ち方が自由な空中手書き文字認識, 第 11 回情報科学技術フォーラム講演論文集, H-031, 第 3 分冊, pp.185–186 (2012).
 [6] 西村好宏, 苗村昌秀: ビデオカメラを利用した空中非目視手書き文字入力方式, 電子情報通信学会技術研究報告, ITS2005-67, pp.119–124 (2006).
 [7] 園田智也, 村岡洋一: 空中での手書き文字入力システム, 電子情報通信学会論文誌, Vol.J86-D-II, No.7, pp.1015–1025 (2003).
 [8] 大倉 充, 森田展行ほか: 空中に描かれた数式を計算するシステムの試作, 映像情報メディア学会誌, Vol.57, No.10, pp.1367–1372 (2003).
 [9] 保呂 毅, 稲葉雅幸: 複数カメラを用いた手書き文字認識システム, WISS2006, pp.121–122 (2006).
 [10] 大倉 充, 真鍋亮介, 島田英之, 島田恭宏: 空中に指先で描かれた続け書き数字の認識アルゴリズム, 情報処理学会論文誌, Vol.52, No.2, pp.910–916 (2011).
 [11] 鈴木 慶, 清水忠昭, 吉村宏紀, 松村寿枝: 2 文字連続筆記した平仮名空中手書き文字の分割法, 情報処理学会論文誌, Vol.55, No.9, pp.2235–2244 (2014).
 [12] 樋口知之: 粒子フィルタ, 電子情報通信学会誌, Vol.88, No.12, pp.989–994 (2005).
 [13] Leap Motion: Leap Motion for Mac and PC, available from (<https://www.leapmotion.com/product/desktop>) (accessed 2015-08-28).
 [14] Vapnik, V.N.: *The Nature of Statistical Learning Theory*, Springer-Verlag (1995).
 [15] Leap Motion: Leap Motion Release Notes, available from (<https://developer.leapmotion.com/documentation/Leap.SDK.Release.Notes.html#version-1-0-8>) (accessed 2015-08-28).
 [16] OpenCV.jp: OpenCV.jp, available from (<http://opencv.jp/>) (accessed 2015-08-28).
 [17] 鳥取県: とっとり産業フェスティバル 2014 パンフレット, 入手先 (http://www.pref.tottori.lg.jp/secure/902611/sanfes2014_PRRpanf_HP.pdf) (参照 2015-08-28).

推薦文

情報処理学会中国支部表彰規定に則り、平成 26 年度 (第 65 回) 電気・情報関連学会中国支部連合大会で発表された中から、特に優秀であることが認められた優秀論文発表賞を受賞した論文である。

(中国支部支部長 谷口秀夫)



重本 賢太郎 (正会員)

2014 年鳥取大学工学部知能情報工学科卒業。2016 年鳥取大学大学院工学研究科博士前期課程を修了。手書き文字認識をはじめとするジェスチャ・インタフェースの研究に従事。



清水 忠昭

1987年大阪大学基礎工学部生物工学科卒業。同年鳥取大学工学部助手。2002年博士（工学）。現在、鳥取大学大学院工学研究科准教授。音声信号処理とインタフェース関連の研究に従事。



鈴木 慶

2012年鳥取大学工学部知能情報工学科卒業。2014年鳥取大学大学院工学研究科博士前期課程を修了。手書き文字認識をはじめとするジェスチャ・インタフェースの研究に従事。



吉村 宏紀（正会員）

1993年鳥取大学工学部知能情報工学科卒業。1997年鳥取大学大学院工学研究科博士後期課程工情報生産工学専攻修了。博士（工学）。現在、鳥取大学大学院工学研究科情報エレクトロニクス専攻助教。画像、音声のデジタル信号処理の研究に従事。電子情報通信学会会員。



松村 寿枝

1998年鳥取大学大学院工学研究科博士前期課程修了。2000年まで大日本スクリーン製造株式会社勤務。2003年鳥取大学大学院工学研究科博士後期課程修了。博士（工学）。財団法人鳥取県産業振興機構勤務。2004年より奈良工業高等専門学校情報工学科助手を経て、現在、同准教授。音声情報処理の研究に従事。