

1 文字単位の区切り操作を必要としない 二次元移動方向による手書き文字入力

西田 好宏^{*1} 野口 祥子^{*1}

A Study for Aerial Handwritten Character Recognition with Auto Period Detection

Yoshihiro Nishida^{*1} and Shoko Noguchi^{*1}

Abstract - This paper describes a method to recognize a character handwritten in the air. This recognition method uses the motion direction instead of positions of the device. It also doesn't use the information of pen-up and pen-down. It selects and orders character candidates with DP (dynamic programming) matching. We prototyped a handwritten character recognition system that detect handwriting period by hand-open and hand-close. This system worked well, but it was very tired that we try input character speedy. In this time, we study a method of auto handwriting period detection instead of hand-open and hand-close. We achieved correct recognition rate about 80% for categories of Hiragana characters by using auto handwriting period detection.

Keywords: Character Recognition, Handwriting in the Air, DP Matching and Input Method Editor

1. はじめに

インターネットの利用目的の中で常に上位を占めているのは、友人とのコミュニケーションや色々な情報を得るための検索である。そのためにはテキストデータの入力が必要不可欠で、片手しか使えない状況でも簡単に文字を入力できたらより便利になると考えられる。そのような背景の中、空中に描いた文字を認識する空中手書き文字入力を検討してきた。空中手書き文字入力の実現には、空中での手指の移動方向を検出するセンサ処理、移動方向から文字を認識する文字認識エンジン処理、認識した文字を基に変換や確定等で文章にする文章編集処理の3つの課題が存在する。

最初に検討したのは文字認識エンジン処理で、絶対位置情報を利用せず、二次元平面上の相対的な移動方向情報から1文字単位で一筆続け書き文字として認識する方法を提案した。具体的には、筆跡検出をマウス入力で代用し、一筆続け書きで入力した文字をリアルタイムで認識するアプリケーションを作成、ひらがな、数字、アルファベットの1文字単位での認識を実現した[1][2]。

その次に検討したのは文章編集処理、即ち文字の変換や確定を行うためのインプットメソッドである。このインプットメソッドを文字入力と同様に空中操作で実現するためには、操作が簡単であるだけでなく、操作方法が動作イメージと近い必要がある。そこで、最小限必要な疑似キーを選びその空中操作を定義し、マウスで手書きした文字の認識に加えて疑似キー入力を認識してIME経由で文章を作成できることを確認した[3]。

最後に、実際のシステム構成に大きく依存するセンサ処理について検討した。システム構成としては、各個人が何らかのセンシングデバイス等を身につけるパーソナルユースの場合と、公共の場にセンサを設置して不特定多数の人に対してセンシングする場合が考えられる。後者のセンサ処理の例として、Microsoft製のKinectセンサを用い、文字区切り検出に「Kinect Interaction」を利用したシステム等を検討した[4]。実際にシステム動作させた結果、1文字単位で一筆続け書き文字として認識できる事が確認できたが、1文字単位の文字区切りの指定が煩雑になり、操作が疲れるだけでなく、文字入力のスピードアップに対しても大きな弊害になると考えられた。

そこで今回は、文字認識エンジン処理に戻って1文字単位の区切り操作を必要としない認識方法について検討した。具体的には、文字の開始はDPマッチングの端点に自由度を与える端点フリーDPマッチングの採用、文字の終了は所定時間の移動停止検出を採用した。その結果、文字単位の区切り操作をなくしても約80%も認識率が得られたので、その実機デモを行う。

2. これまでの研究経緯

2.1 マウスを用いた評価アプリケーション

紙に文字を書く場合、文字を書いている時にはペンは紙に接し、文字を書かずにペンの位置を移動する時にはペンは紙から離れている。しかし、空中で文字を書く場合は紙に接している/いないというペンのON/OFFの判定が非常に困難である。そこで、1文字は一筆続け書きで書くこととし、1文字単位のON/OFF操作を設けることとした。作成した評価アプリケーションでは、この1文字単位の入力操作をマウスクリックのON/OFFで実現していた。この筆記イメージを図1に示す。

^{*1}: 福井工業大学 電気電子工学科

^{*1}: Department of Electrical and Electronics Engineering, Fukui University of Technology

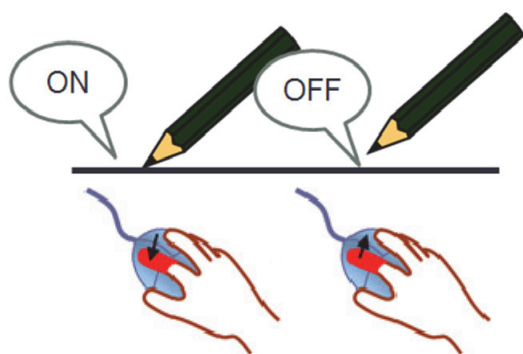


図 1. マウスクリックによる筆記イメージ
Fig. 1 ON/OFF image of pen and mouse

文字認識エンジンと入力メソッドの機能を確認するための評価アプリケーションを図 2 に示す。GUI 画面には第 1 候補から第 3 候補までの候補文字とその DP マッチングでのエラー値が表示される。候補に表示される文字は「あ」「い」と言った文字以外に「EN」「SP」といった特殊キー（「Enter」「Space」）の擬似操作を示すものが表示される。IME には Google IME を使用し、出力先はメモ帳とした。

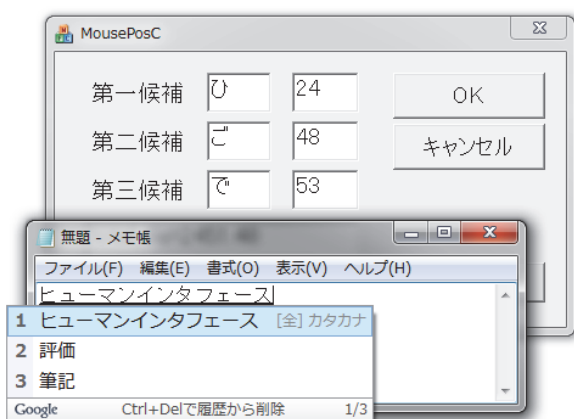


図 2. 評価アプリケーション
Fig. 2 Application for handwritten character

2.2 センサを使用したシステム

センサを使用した空中手書き文字入力には以下の 2 つのパターンが考えられる。まず 1 つ目は、文字を書く手にセンサを内蔵したデバイスを装着あるいは保持し、文字を書く手の座標の変化を利用するものである。この場合は加速度センサなどをデバイスに内蔵し座標の変化を取得する方法が考えられる。このデバイスの例としては、富士通研究所が 2015 年 1 月に発表した指輪型ウェアラブルデバイス等がある[5]。この指輪型ウェアラブルデバイスは、充電式の内蔵バッテリーを電源とし、モーションセンサ（加速度／ジャイロ／磁気）、センサ処理マイコン、状態表示 LED、入力操作ボタンなどを備える。

2 つ目は、図 3 に示すように設置されたセンサに対して手を動かし座標の変化を取得するものである。この場

合は距離センサを内蔵したモーションセンサを使用することとなり、例としては Kinect や Leap Motion があげられる。Kinect センサはゲーム機用に開発されたセンサで、骨格情報から手等の座標を取得することができる。

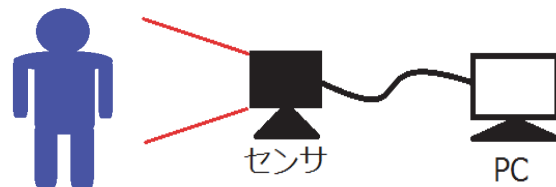


図 3. 設置型センサを利用するシステム
Fig. 3 Motion sensor setting type systems

デバイスを利用する場合にはボタン操作により 1 文字単位の ON/OFF 操作が行えるが、Kinect のような設置型センサを用いる場合にはボタン操作以外の 1 文字単位の ON/OFF 操作手段が必要になる。そこで、この 1 文字の区切りに Kinect Interaction による手の開閉状態検出を利用した。具体的には手を握っている状態が文字を書いている状態、開いている状態が文字を書いていない状態として、手を開いている状態から握っている状態へ移行した瞬間から筆跡の検出を開始し、そのまま一筆書きで文字を書き、手を握っている状態から手を開いた状態にすることで筆跡の検出を終了し、それまでの筆跡から文字認識エンジンによる文字の判定を行う構成にした。この操作イメージを図 4 に示す。

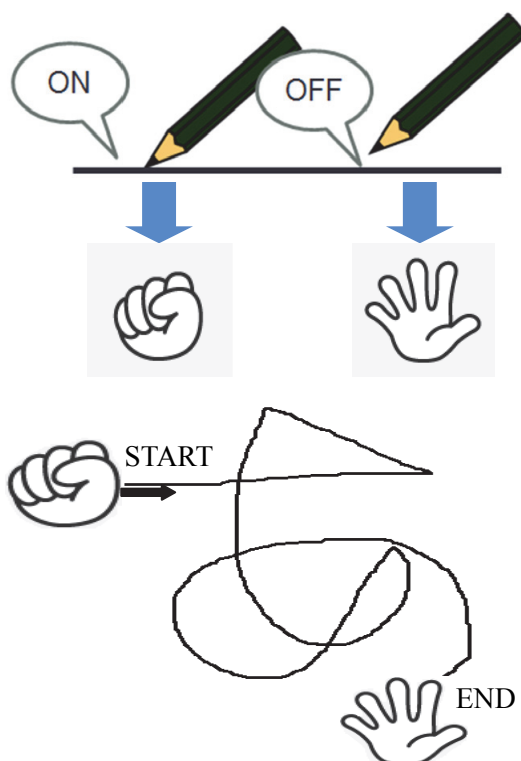


図 4. 手の開閉を利用した操作例
Fig. 4 Operation method of Kinect Interaction

文字区切りに Kinect Interaction を利用した評価アプリケーションを作成した結果、空中に書いた文字が正常に認識できることが確認できた。しかし、文章を入力し編集するためには何度も手を開閉操作することになり、非常に疲れることがわかった。

デバイスを用いてボタン操作する場合も含め、文字区切りのための操作は疲れるだけでなく、文字入力のスピードアップに対しても大きな弊害になると考えられるため、文字区切り検出の自動化を検討した。

3. 文字区切りの自動化

3.1 文字区切り時に発生するノイズ

文字の書き終わりの検出方法として、動きの停止（ホバリング状態）が考えられる。ユーザーとしても、1文字の入力が終了、その認識結果を確認、次のアクションのようなインタラクティブなマン・マシン・インタフェースを実現する上で自然な操作と考えられる。

問題は文字の書き始めの検出である。前回の文字が書き終わった地点、すなわち停止していた地点から次の文字を書き始める事となる。一般的に文字の書き始めは左上、書き終わりは右下の場合が多く、例えば図5のように、「あ」の終了地点から次の文字の「か」の開始地点である左上付近に移動し、そこから文字を書き始めることになる。この図5の赤線で示す「か」の開始地点へ移動するまでの方向コード情報がノイズとして入り込んでしまう問題が発生する。そのため、このノイズを除去、あるいは無視する方法が必要になり、文字認識エンジンの DP マッチングを見直すことにした。

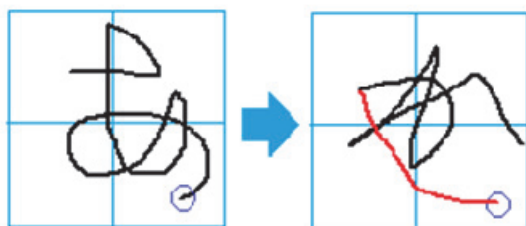


図5. 文字の終了から開始地点までのノイズ
Fig. 6 Noises of between end-point and start-point

3.2 端点フリーDP マッチング

空中に描いた動きの方向をコード化した筆跡データと辞書データとの照合には動的計画法（Dynamic Programming）による手法の DP マッチングを用いている。DP マッチングは長さの異なる2組の連続するデータ間の隔たり「ペナルティ」として、各データ間の隔たりを示す「局所ペナルティ」と、伸縮（ずらし）に応じた「経路ペナルティ」を演算、累積加算していき、「累積ペナルティ」が最小となるパターン（経路）を探索する手法である。この DP マッチングを筆跡データとすべての辞書

データに対して行い、最も累積ペナルティが小さかった辞書コードの文字を入力文字と判定する。

一般的な従来の DP マッチングは開始点から終了点までのペナルティを累積加算しているため、筆跡データにノイズが含まれている場合は正しくマッチングがとれない問題が発生する。そこで、まずプログラム上で辞書データと筆跡データの文字列を反転させ、ノイズのない筆跡データの書き終わり側からマッチングを開始して、マッチングの最後の方に筆跡データの書き始め側、すなわちノイズが含まれる部分が来るようにした。

次に、ノイズの含まれる終端側についてのフリー化をする。フリー化されていない DP マッチングの場合、各要素での累積ペナルティ値がすべて算出された後、図6の上側にあるように一番右下の累積ペナルティ値のみを参照している。これに対し、終端をフリー化した DP マッチングの場合は、図6の下側にあるように、右下のものだけでなく、同じ列の複数の位置に対して、最も小さい累積ペナルティ値を返すようになっている。ただし、同じ列の全ての位置を対象にして比較をしてしまうとハイフンなどの単調な文字として認識してしまうことがあるため制限を設けている。具体的には、プログラム上で筆跡データの60%の位置以降において累積ペナルティが最小となる位置の探索を行っている。

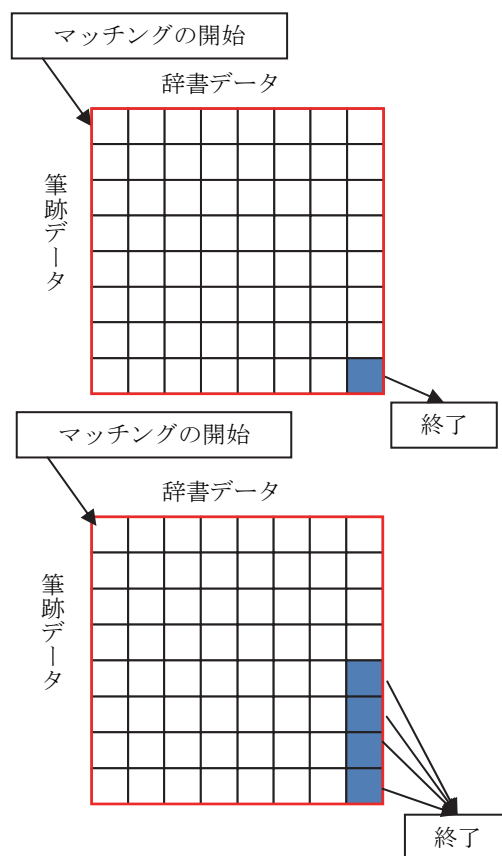


図6. DP マッチングの端点フリー化の比較
Fig. 6 Previous DP matching method and new DP matching method

4. 端点フリーの効果確認実験

4.1 実験方法

端点フリーDP マッチングの比較に使用する筆跡データは、センサの性能や仕様による誤差を無くするため、マウスを使用して生成した。ただし、旧評価アプリケーションのように筆記中に筆跡を表示するのではなく、実際の空中手書きの時のように筆跡が見えないようにした筆跡データ収集用アプリケーションを作成した。

収集する筆跡はひらがなの「あ」から「ん」までの46文字に加え、濁点半濁点を合わせた合計71文字である。また、次の文字の開始位置への移動ノイズは、直前に書いた文字の終了位置に依存する。そのため、直前に書く文字は「か」「く」「し」「め」「や」の5種類として、その後にそれぞれ71文字を書くため、計355文字パターンについて評価した。

4.2 実験結果

6名の被験者に筆跡データ収集用アプリケーションを使用して、上記の355文字、直前に書く文字を合わせた710文字を筆記してもらった。こうして収集したノイズを含む筆跡データを従来のDPマッチングと新方式の端点フリーDPマッチングにかけ、どれだけ正しく認識できているか（第一候補に挙がっているか）を調査した。

その結果を表1に示す。端点フリーDPマッチングを導入することによって、文字の書き始め位置への移動ノイズが含まれていても認識率を約80%まで改善できることが確認できた。

表1. DPマッチング方式による認識率の比較

Table.1 Recognition rates by both DP matching method











被験者	従来のDPマッチング	端点フリーDPマッチング
A	15.5	73.5
B	16.9	82.5
C	15.8	72.1
D	16.9	80.8
E	22.3	73.0
F	14.9	78.6
合計	17.0	76.8

5. その他の改良点

今回、使い勝手を向上させるために第一候補にならなくても、第二候補もしくは第三候補として認識された場合には、その文字を直接指定できる擬似操作を追加した。さらに、促音や拗音などの入力やIMEの矢印キー操作について検討して擬似操作を追加した。主な機能（IMEのキー入力）とその擬似操作を表2に示す。

表2. 機能（IMEのキー入力）とその擬似操作

Table.2 Special function keys and Operations.

機能（キー）	擬似操作	機能（キー）	擬似操作
確定 (Enter)		削除 (Back Space)	
予測 (Tab)		変換 (Space)	
第二候補		第三候補	
促音,拗音		句読点	
左矢印		右矢印	

6. まとめ

これまでに検討して来た、文字認識エンジン、インプットメソッド及びセンサを接続して空中手書き文字入力システムを構築できたが、文字区切りのための操作が煩雑で、非常に疲れる問題が発生した。そこで、使い勝手良く楽にスピーディに入力するために、文字区切りの自動検出について検討した。

DPマッチングの方向を逆方向にして端点フリーDPマッチングを導入することにより、筆記の動きの停止により1文字の書き終わりとする事で、文字区切りの操作を無くしても約80%の認識率が得られた。

さらに、特殊キーに対応した擬似操作を増やして使い勝手向上を図ったので、実際に操作してもらいたい。

参考文献

- [1] 西田好宏,小倉一孝,三浦浩一,松田憲幸,瀧寛和,安部憲広: 移動方向情報のみを利用した空中手書き文字認識; ヒューマンインタフェース学会誌, Vol.12, No.3,pp.289-296 (2010).
- [2] 青池勇樹,西田好宏: 二次元移動方向に基づく空中手書き文字認識のアルファベット対応の検討; ヒューマンインタフェースシンポジウム 2011, pp.701-702 (2011).
- [3] 西田好宏,入江臣知,吉田大志: 空中手書き文字入力におけるインプットメソッドの検討; ヒューマンインタフェースシンポジウム 2012, pp.317-318, (2012)
- [4] 西田好宏, 吉田大志, “「Kinect Interaction」を利用した空中手書き文字入力の検討”, ヒューマンインタフェースシンポジウム 2013, pp.275-276,(2013)
- [5] 株式会社富士通研究所プレスリリース
<http://pr.fujitsu.com/jp/news/2015/01/13.html>