

運指情報による推測型日本文入力手法の提案と評価

藤田 晋也[†] 赤池 英夫^{††} 角田 博保^{††}

本研究では、ウェアラブル機器向けの新しい日本文入力手法を提案する。近年、携帯電話やスマートフォンのような携帯情報端末や、ウェアラブルコンピュータが急速に普及してきている。このような環境では、フルキーボードに代わる優れた入力手段が求められている。そこで、曖昧性をもった運指情報と、それを絞り込むための統計的手法を用いた入力手法を提案、実装し、有用性の評価を行う。

A Predictive Japanese Text Input Method Using Fingering

SHINYA FUJITA,[†] HIDEO AKAIKE^{††} and HIROYASU KAKUDA^{††}

In this research, we propose a new Japanese text input method for wearable devices. Recently, a handheld terminal such as cellular phones and smart phones and a wearable computer are rapidly widespread. In such an environment, a good input method to take the place of a full keyboard is requested. Then, we propose and implement the input method using the ambiguous fingerwork information and the statistical technique to narrow it, and evaluate utility.

1. はじめに

近年、携帯電話やスマートフォン等の携帯情報端末の普及、ウェアラブルコンピュータの登場により、フルキーボードの使えない環境で、日本文を入力する機会が増加してきている。そのような環境ではフルキーボードの代替手段として、マルチタップ方式やソフトウェアキーボード等が採用されているが、それらの多くには入力速度が非常に劣る、習熟までに時間がかかるなどの問題がある。

そこで本研究では、両手の各指先の位置にスイッチを取り付けたデバイスを装着し、“目の前にQWERTYキーボードがあると仮定”して日本文のローマ字入力を行ったときの運指を検出し、運指の組み合わせから、かな候補を推測し提示する手法を提案し、その有用性を評価する。これにより、フルキーボードに熟練したユーザの学習コストの低減、フルキーボードに近い入力速度を得ること等が期待できる。

2. 関連研究

ウェアラブル向けの文字入力装置をいくつか挙げる。Twiddler¹⁾ は片手で文字入力ができる装置だが、習熟までに時間がかかる。Virtual Keyboard²⁾ は平面状に仮想キーボードを投影し、位置センサによって打鍵したキーを検出する装置である。しかし、投影のためにある程度の広さの平面が必要であったり、手元を確認する必要があるといった問題点がある。FtoKey³⁾ は、これらの問題点を解決した手法である。両手の指先の位置に各1個、計10個のセンサを備えており、これをユーザが手に装着し、任意の場所で打鍵を行うことにより高速に日本文を入力することができる。しかし、問題点として、実験装置がプロトタイプのみであること（机上に敷いたアルミシート上でしか打鍵できない、クリック感が無く入力感覚が薄い、誤検出によるエラーが非常に多い）、辞書が制限されている（実験用の文章の入力は可能だが、自由な文章の入力が難しい）、文節ごとの入力しかできない等、実用的な問題が挙げられる。

3. 提案手法

提案手法ではFtoKeyと同様に、両手の指先の位置に各1個、計10個のスイッチを取り付けたデバイスで入力を行う。このデバイスを装着することで、机の上

[†] 電気通信大学 電気通信学研究科 情報工学専攻
Department of Computer Science, Graduate school of
Electro-Communications, The University of Electro-
Communications
^{††} 電気通信大学 電気通信学部 情報工学科
Department of Computer Science, The University of
Electro-Communication

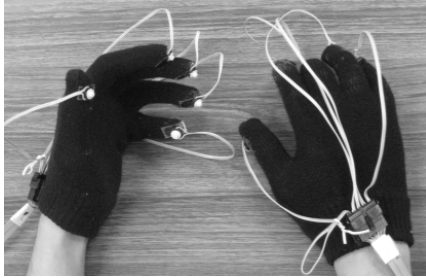


図1 デバイス概観

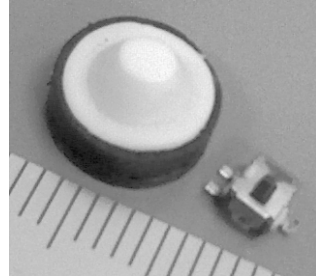


図2 自作スイッチと内蔵した小型検出スイッチ

や大腿部等、任意の場所で打鍵することが可能になる。

親指以外の8個のスイッチが文字キーに対応し、両親指のスイッチは変換や絞り込みといった特殊操作のキーに対応する。文字スイッチが8個のみのため、これらだけでは入力したい文字を一意に決定することができない。このため、1つのスイッチに複数の文字が割り当てられている。スイッチと文字の割り当ては“左小指:QAZ”、“左薬指:WSX”のように、通常のQWERTYキーボードで打鍵を行う指と対応している。

ユーザがキーボードで日本語のローマ字打ちをするのと同様の指使いで打鍵すると、システムはその運指情報により、かな文字列の候補を生成し、後述するアルゴリズムによって、より日本語として成り立ちそうな順に提示する。入力したいかな文字列の順位が低く、提示されない場合は、文頭からかなの絞込み操作を行うことができる。かなを確定した後は、外部のかな漢字変換エンジンを用いて連文節漢字変換を行う。これらの入力操作や、確定後のカーソル操作、修正操作を10個のスイッチの組み合わせのみで行うことができる。

4. 実装

4.1 デバイス

手袋の指先部分へスイッチを取り付けたデバイスを作成した(図1)。

指先に取り付けるスイッチへは、“クリック感”を導入した。キーボードのようなクリック感を導入することで、意図しない入力を減らし、軽快な入力を可能にすることが期待できる。クリック感を与えるために、メンブレンキーボードなどに使われるラバードームを参考に、ゴム部品を作成した。これは、ゴム形状を設計し、樹脂を卓上フライス盤で削り出して型を作成し、その中に模型用のシリコーンゴムを流し込んで成型した。ゴム部品を使用して作成したスイッチと、これを

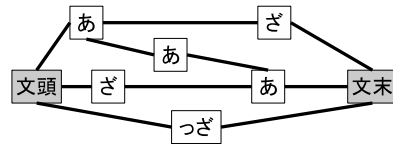


図3 データ構造 (左小指 {Q,A,Z} を三回打鍵)

作成するために内蔵した小型検出スイッチを図2に示す。

4.2 かな候補の順位付け

10個のスイッチを用いて打鍵しただけでは、その曖昧性によりかな候補の数が膨大になってしまう(例:“KAKUDAKENNKYUUSITU”と入力できるような打鍵を行うと、かなとして成立する候補数は869,256個)。自前の辞書で絞り込む方法もあるが、そうすると連文節変換や辞書に無い文字列の入力が難しくなるため、かな候補に順位付けを行うアルゴリズムを採用した。

打鍵を行ったときの内部状態のデータ構造は図3ようになる。各リンクにコストを与え、文頭から前向きにViterbiアルゴリズムを行った後、文末から後向きにA*アルゴリズムを適用することで、N-best解を得ることができる⁴⁾。

コストの値として、隣り合うかな2文字の頻度、かな2-gramを使用する(出現確率の逆数の対数をコストとする)。これにより、より日本語として成立しそうなかな文字列の順位を高くすることができる。コーパスとしては青空文庫⁵⁾から新字新仮名遣いで記述された文献を用いた。これを形態素解析エンジンMeCab⁶⁾を用いてかなに変換した後に、2-gramを計算した。計算には約1億文字のかなを使用した。

一例として、“KAKUDAKENNKYUUSITU”と入力できるような打鍵を行った後、アルゴリズムを適用すると、最適解として“かくだけんきゅうしつ”を得ることができた。

表 1 各モードでの操作

モード	左小指	左薬指	左中指	左人差指	左親指	右親指	右人差指	右中指	右薬指	右小指
未入力	[QAZ]	[WSX]	[EDC]	[RFVTGB]	記号選択モード	■カーソルモード	[YHNUJM]	[IK]	[OL]	[P-]
【カーソル】	HOME	END	↑	↓	BS	□未入力モード	←	→	Enter	Enter
記号選択	確定	確定	次ページ候補	前候補	次候補	■記号解除モード	確定	確定	確定	確定
【記号解除】	x	x	x	x	未入力モード	□記号選択モード	x	x	x	x
かな入力	[QAZ]	[WSX]	[EDC]	[RFVTGB]	かな選択モード	■絞込みモード	[YHNUJM]	[IK]	[OL]	[P-]
かな候補選択	かな無変換確定	カナ無変換確定	次ページ候補	前候補	次候補	■絞込みモード	漢字変換モード	漢字変換モード	漢字変換モード	漢字変換モード
【絞込み・消去】	絞込み1字解除	絞込み全解除	未入力モード	次頁絞込み候補	BS	□かな入力モード	絞込み候補1	絞込み候補2	絞込み候補3	絞込み候補4
漢字変換	x	前ページ候補	次ページ候補	前候補	次候補	■文節調整へ	確定	確定	確定	確定
【文節調整】	先頭文節選択	末尾文節選択	左文節選択	右文節選択	かな選択モード	□漢字変換モード	左文節選択	右文節選択	文節を縮める	文節を伸ばす



図 4 実験アプリケーション

4.3 絞込み

かな候補の順位付けの結果，入力したいかなの候補が表示されない場合，文頭から絞込みを行うことができる。システムは絞込みできるかな候補のリストを提示するが，このリストは，前の文字とのかな 2-gram のコストが低い順に並び替えられたものである。

絞込みは，使用しなくなる各リンクのコストを一時的に無限大にすることで行っている。この状態で，再度 Viterbi アルゴリズムと A* アルゴリズムを適用することで，絞込みをしたかな候補の N-best 解を得ることができる。

4.4 日本文への変換

今回は，Linux でよく使用されるフリーのかな漢字変換エンジン Anthy⁷⁾を使用した。外部のかな漢字変換エンジンを使用することで，連文節変換や学習機能を使用することが可能になった。

5. 入力操作について

入力とその修正に関する操作を，10 個のスイッチのみで行うので，提案手法には多くのモードがある。

表 1 に，各モードにおいて各スイッチに割り当てた機能を示す。ただし操作によりモードが切り替わる場合には，遷移先のモードを示している。右親指のスイッチにだけ，押している間だけ一時的に別のモード

に切り替える機能を割り当てた。残りのスイッチはクリック操作 (押して離すという一連の動作) 後に各機能が実行される。名前に“【】”がついているモードは，右親指を押下することにより遷移できるモードを示す。また，右親指のモード遷移のうち，“■”記号は右親指を押す操作，“□”記号は右親指を離す操作によるモード遷移を表す。

どのモードでも，右親指を押下した状態で左親指を押すことで，バックスペースもしくは 1 段階前のモードに戻るといった操作に対応する。

“絞込み・消去モード”の“絞込み候補 n ”については，絞り込めるかながコストの小さい順にガイドに表示されるので，それに対応するスイッチを押すことで絞込み機能を実行することができる。絞込み候補が 4 つに収まらない場合は，左人差指の“次頁絞込み候補”機能で絞込み候補の表示を次のページに送ることができる。

6. 評価実験

6.1 実験方法

フルキーボードおよび提案手法による各 5 セッションからなる被験者入力実験を行った。各セッションでは漢字かな交じり文の短文 (平均 30 文字) の入力を 20 回行う (合計 600 文字)。各短文の入力では，確定時に例文と異なる入力が行われている場合は確定操作を受け付けず，次の短文の入力を行うことができない。カーソル移動やバックスペース等による修正はいつでも可能である。目標の漢字が変換で出てこない場合などは，その漢字を含む熟語を入力してから不要な部分を消去するというような入力も可能である (例: 「攻」と入力したいときに, 「こうげき」を変換して「攻撃」とし, 「撃」を消去する)。

被験者はタッチタイピングに熟練した学生 4 名である。それぞれ QWERTY フルキーボードの入力実験 (かな漢字変換エンジンに Anthy を使用) を全て完了した後，提案手法 (何も無い机を打鍵) の入力実験を行った。被験者のうちの 2 名はフルキーボードの入力

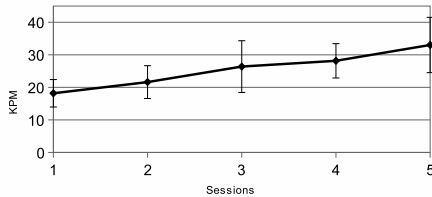


図 5 提案手法の各セッションの平均 KPM

文章として短文セット A, 提案手法の入力文章として短文セット B を用いた。また, もう 2 名の被験者はフルキーボードの入力文章として短文セット B, 提案手法の入力文章として短文セット A を用いた。短文セット A と B の全体に占める漢字の割合, ひらがなの割合, カタカナの割合はほぼ同じになるように調整した。全体に占める漢字の割合は, 約 25% となった。

Anthy の学習機能は切っておいた。実験アプリケーションは Java 言語で実装した。どのモードでも, どのスイッチがどの機能に対応するかを示すガイドは常に表示されている (図 4)。

6.2 結果と考察

提案手法の各セッションの全被験者の平均 KPM (漢字かな混じり文字数/分; 提示された短文の入力開始から, 修正も含めて完了までにかかった時間で計算) を図 5 に示す。横軸はセッション番号, エラーバーは被験者間の標準偏差である。1 回目の確定時にエラーがあった文については解析から除いた (確定前の修正は可能)。5 セッション目で, 平均 33KPM の入力速度を得ることができた。学習による入力速度の上昇が見て取れるので, 今後セッションを続けることで, さらに入力速度の上昇が期待できる。

アンケートでは, 「日常的な文章ならあまり絞込みを必要とせずに入力できた」, 「入力したい指の打鍵が確実に行えた」, 等の肯定的なコメントが多かったが, 「かな候補が多いため, 運指の失敗に気付くまでに時間がかかる」等の否定的なコメントも得られた。

提案手法とフルキーボードの全被験者の平均 KPM の比較を図 6 に示す。横軸はセッション番号, エラーバーは被験者間の標準偏差である。提案手法は, フルキーボードにはまだまだ及ばないが, 5 セッション目ではフルキーボードの約 1/3 の速度を得ることができたことがわかる。

提案手法とフルキーボードの全被験者の平均文字打鍵時間の比較を図 7 に示す。文字打鍵時間とは, 通常の文字打鍵 (変換操作やカーソル操作以外) が連続した場合のそれぞれが行われた時刻の間隔のことである。

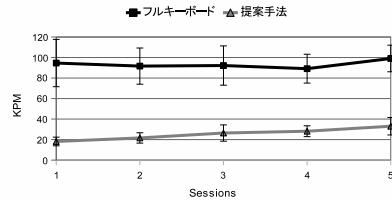


図 6 提案手法とフルキーボードの各セッションの平均 KPM の比較

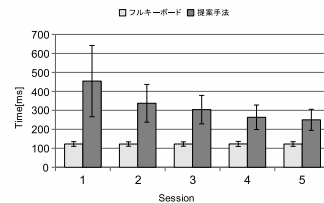


図 7 提案手法とフルキーボードの各セッションの平均文字打鍵時間の比較

横軸はセッション番号, エラーバーは被験者間の標準偏差である。5 セッション目で, 提案手法の平均文字打鍵時間はフルキーボードの約 2 倍になった。このままセッションを続けていき, 装置に慣れることで, フルキーボードの打鍵時間に近づいていくと考えられる。

7. おわりに

本稿では, タッチタイピングに熟練したユーザーのための, ウェアラブル機器向け日本語入力手法を提案し, その実装と評価実験について述べた。

今後は, 引き続き評価実験を進め, 被験者数とセッション数を増やした実験や, 電車内や歩行中などのより実際のウェアラブル環境における実験, 他のウェアラブル入力装置との比較を行い, 提案手法の有用性を検証したい。

参考文献

- 1) HandyKey Corporation:Twiddler, <http://www.handykey.com/>
- 2) i-Tech Dynamic:Virtual Keyboard, <http://www.virtual-laser-keyboard.com/>
- 3) 青木亮磨:運指情報を利用した推測による日本語入力手法 “FtoKey” の提案と評価, 電気通信大学修士論文,2007.
- 4) 永田昌明:前向き DP 後向き A* アルゴリズムを用いた確率的日本語形態素解析システム, 情報処理学会研究報告,94-NL-101-10,pp.73-80,1994.
- 5) 青空文庫,<http://www.aozora.gr.jp/>
- 6) 工藤拓:MeCab,<http://mecab.sourceforge.net/>
- 7) 田畑悠介:Anthy,<http://anthy.sourceforge.jp/>