

## 片手用キーボードのための 打鍵間隔を用いた入力単語推定手法

片山 拓也<sup>†1</sup> 村尾 和哉<sup>†1</sup>  
寺田 努<sup>†1,†2</sup> 塚本 昌彦<sup>†1</sup>

近年のコンピュータ小型化に伴い, ウェアラブルコンピューティングに関する注目が高まっている. ウェアラブルコンピューティング環境では, 携帯性や装着性の観点から小型の入出力デバイスが望まれる. 一般に, コンピュータへの入力デバイスとしてはキーボードが広く普及しており, 多くのユーザがキーボードの入力に慣れ親しんでいる. しかし, ウェアラブルコンピューティングのためのキーボードの単純な小型化は, キータッチのしやすさなどのユーザビリティに影響をあたえるので限度がある. そこで本研究では, ユーザが既に体得しているキーボード入力の能力を活かすために, 既存のキー配列をそのままにしながらキーボードを左右に分割し, どちらか一方のみを装着して用いる手法を提案する. 提案手法では, 単語の切れ目毎にキーボード半分の打鍵情報から入力単語を推測する. 提案手法を用いることで, 従来のキーサイズと入力動作を最大限に保ったままキーボードの大きさを半減できる.

### An Input-Word Estimation Method for One-Handed Keyboard using Keying Interval

TAKUYA KATAYAMA,<sup>†1</sup> KAZUYA MURAO,<sup>†1</sup>  
TSUTOMU TERADA<sup>†1,†2</sup> and MASAHIKO TSUKAMOTO<sup>†1</sup>

In recent years, the development of technology has stimulated interest in wearable computing. In wearable computing environments, compact I/O devices are desirable from the aspect of comfortability and portability. Now, many skilled users are accustomed to input with a keyboard, however, there is a limitation of the miniaturization because of the usability of keying. Therefore, in this paper, we propose a method to miniaturize by excluding the half of the keyboard. Our system estimates the input word using keying information of the word from one-hand. Our system makes it possible to reduce the size of the keyboard to half without decreasing the usability.

#### 1. はじめに

近年のマイクロエレクトロニクス技術の発達に伴い, コンピュータの小型化が進んでおり, 携帯電話等の情報機器を携帯する利用者が増えた. さらに現在は, コンピュータを着るように常時装着して生活するウェアラブルコンピューティングに関する注目が高まっている<sup>1),2)</sup>. これらの環境では, 各種デバイスは携帯性や装着性の観点から小型であることが求められる. 現在, デスクトップコンピューティング環境の文字入力にはキーボードが主に用いられているが, フルサイズのキーボードは持ち歩きに適さず, 衣服への装着も難しいため, ウェアラブルコンピューティング環境で利用されている例は少ない. 一方, キーボードの単純な小型化は, キータッチのしやすさなどのユーザビリティを悪化させる原因となるので限度がある.

これまでに小型の文字入力デバイスに関する様々な研究が行われ, 多種多様なデバイスが提案されているが, キーボードは世界中に広く普及しており, コンピュータの使用頻度が高いユーザのほとんどがキーボードの入力に慣れ親しんでいるため, 新たな形状の文字入力デバイスを受け入れてもらうには敷居が高く, 習熟してもキーボードほどの入力速度が得られないものがほとんどである.

そこで本研究では, ユーザが既に体得しているキーボード入力の能力を活かすために, 既存のキーボードのキー配列をそのままにしながらキーボードを左右に分割し, どちらか一方のみを用いることでキーボードの小型化を図る手法を提案する. 提案手法では, ユーザは片方の手で従来と同様にキーボードを打鍵して, もう一方の手は机や腿などを空打ちする. その際, 空打ち動作で発生するキー入力のタイムラグを利用して, 空打ちしている手の打鍵数を推測する. そして, 単語の切れ目ごとに入力された単語を推定し, 優先度順に候補を提示する. 提案手法を用いることで, 従来のキーボードのキーサイズと入力動作を最大限に保ったままキーボードの大きさを半減できる. 本研究ではシステムのプロトタイプを実装し, 文章の入力速度を調査した.

以下, 2章で関連研究を紹介し, 3章で提案手法を説明し, 4章で評価について述べる. そ

<sup>†1</sup> 神戸大学大学院工学研究科

Graduate School of Engineering, Kobe University

<sup>†2</sup> 科学技術振興機構さがけ

PRESTO, Japan Science and Technology Agency

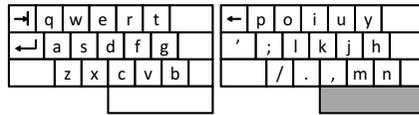


図 1 Wearable Half Keyboard のキー配置  
Fig. 1 The key layout of Wearable Half Keyboard<sup>6)</sup>

して、5章で考察を行い、最後に6章で本研究のまとめを行う。

## 2. 関連研究

これまでに小型文字入力デバイスは数多く提案されている。ウェアラブルコンピューティングに特化したものには装着型センサを用いた入力デバイス<sup>3),4)</sup>が提案されている。文献4)では腕につけた装着型カメラを用いて、空中での手の軌跡を認識して文字入力を行う。これらは、装着型センサを用いることで小型化という要件は満たすが、特殊な文字入力方法を用いるという点で利用者への敷居が高い。

文字入力デバイスの小型化にはキー数を減らすというアプローチが最も一般的であり、携帯電話で採用されているめくり方式や、CUT方式<sup>5)</sup>などがこれにあたる。めくり方式では1キーに複数の文字が割り当てられて、その打鍵回数で入力文字を決定する。CUT方式では、めくり方式からさらに文字のキーへの割り当てを工夫し、使用頻度が高い文字を少ない打鍵回数で入力可能にしている。ウェアラブルコンピューティング用に市販されているものでは、Matias社のWearable Half Keyboard<sup>6)</sup>やHandkey社のTwiddler<sup>7)</sup>などがある。Wearable Half Keyboardは図1(左)に示すようにキーボードの左半分のみを切り取ったようなデザインになっており、キーボードの右半分にある文字を打つにはスペースキーを押しながら対応する左半分のキーを押す。キーの対応は通常のキーボードの右半分のままではなく、図1(右)のような写像型を採用している<sup>8)</sup>。Twiddlerは手に固定したデバイスの前面に4キーとジョイスティック、背面に12キーが配置されているデザインになっており、文字入力の際には背面の12キーの同時入力の組み合わせで全てのアルファベットが入力できる。しかし、これらはいずれも片手入力を想定しており、普及しているキーボードとは異なる特殊なキー配列であり、少ないキー数で全ての入力をカバーするために同時入力などの特殊な入力を要求する。そのため習熟に時間を要する上、両手で入力を行うキーボードに比べて入力速度が大きく劣る。

それに対して提案手法は、従来のキーボードの配列をそのまま用いる点と、従来のキー

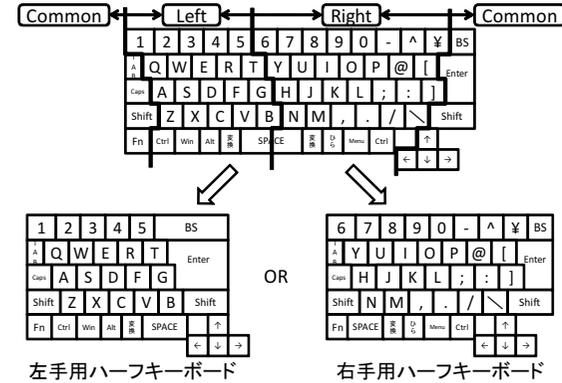


図 2 キーボードの分割位置  
Fig. 2 The division of keyboard

ボードの入力動作を最大限維持している点によって、ユーザが既に体得しているキーボード入力の能力を活かした入力を実現できる。提案手法では打鍵間隔を用いて入力単語を推定するが、似た技術に打鍵認証<sup>9)</sup>というものがある。打鍵認証ではキー打鍵タイミングから個人の認証を行う技術で、導入コストを抑えられるが、他のバイOMETRICS認証と比べて認証精度が低いという問題がある。

## 3. システム設計

### 3.1 システム概要

提案手法では図2のようにキーボードを左右半分に分割し、どちらか一方のみを用いて片手で入力を行う。以降、この半分に分割されたキーボードをハーフキーボードと呼び、入力に用いる手を打鍵手、用いない手を非打鍵手と呼ぶ。入力の際、ユーザは非打鍵手は自分の腿や机などを空打ちすることで通常のキーボードに近い入力動作を維持できる。それと同時に、この非打鍵手の空打ちによって打鍵手のキー入力間に通常のキーボードを使用している時と同様の打鍵間隔が発生する。提案手法では打鍵手のキー入力情報のみから、その打鍵間隔を利用して非打鍵手の入力を推測する。以上によって、従来のキーボードのキーサイズと入力動作を最大限に維持したまま、デバイスの大きさを半減する。ここで、[Space]キー、[Enter]キーなどの使用頻度の高いキーは左右どちらのハーフキーボードも配置する。

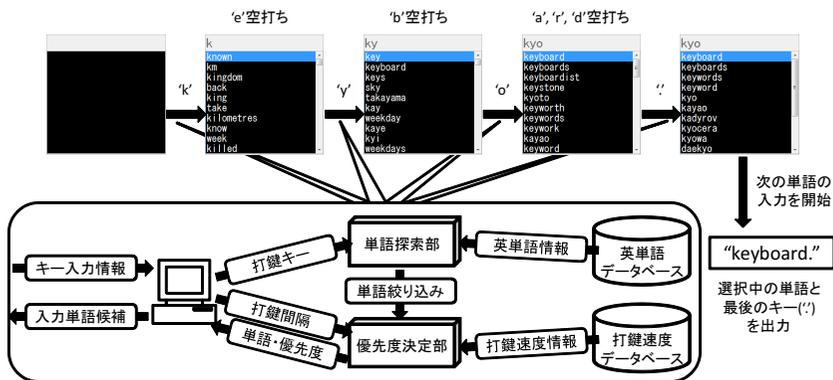


図 3 システムの動作フロー（右手での入力）  
Fig. 3 The system flowchart (inputting with right hand)

### 3.2 動作フロー

システムの動作例を図 3 に示す。図 3 は打鍵手が右手の時の“keyboard”の入力例を示している。ここで、提案手法はどの言語にも対応可能であるが、本稿では英文の入力について述べる。システムは英単語データとユーザの打鍵速度データを保持している。提案手法では、単語探索部において打鍵手のキー入力の度にそのキー入力で成立する英単語の候補を絞り込むと同時に、優先度決定部において打鍵間隔からその間に含まれる非打鍵手の打鍵数を推測し、推測された打鍵数に応じて候補に優先度を付けて優先度順に提示する。そして、アルファベット以外のキーが入力される単語の切れ目で単語選択の状態に入り、ユーザは実際に入力した単語を候補の中から選択する。その後、次の単語の入力が開始されると選択中の単語と最後に押されたアルファベット以外のキーを出力する。ここで、実際に入力した単語が候補の一番先頭に提示された場合は選択動作が必要ないため、従来のキーボードの入力動作がほぼ維持されている。

このように、提案システムでは「候補の絞り込み」、「優先度計算」の後に候補を提示する。以下でそれぞれについて説明する。

### 3.3 候補の絞り込み

提案システムでは入力された単語はあらかじめ作成された辞書ファイルの中から検索される。ところが、通常の辞書に収録されている見出し語はそれぞれの単語の基本形のみであるのに対して、英文ではそれらの単語は複数形や受動形、現在進行形など様々な形に変化して

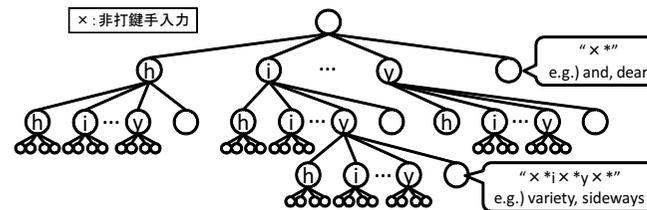


図 4 打鍵手が右手の場合のトライ木  
Fig. 4 The trie tree of right-half keyboard

使用される。そこで、一般の人が実際にどのような形で単語を使用しているかを学習させるために、提案システムでは Wikipedia の Web ページをクローリングして辞書を作成する。

辞書ファイルからの単語の検索にはトライ木の概念を用いる。トライ木とは辞書の実装構造としてよく用いられるもので、その特徴として、各ノードに固有のキーが格納されるわけではなく、根から自身までのルートがキーに対応している。根には空の文字列が対応し、あるノードの配下にある全ノードは、自身に対応する文字列を接頭部としてもつ。右手を打鍵手にした時の提案システムのトライ木の構造を図 4 に示す。根は‘h’, ‘i’, ‘j’, ‘k’, ‘l’, ‘m’, ‘n’, ‘o’, ‘p’, ‘u’, ‘y’の右手入力文字をもつ 10 ノードと文字をもたない 1 ノードを子としてもつ。同様に、右手入力文字をもつノードは、右手入力文字をもつ 10 ノードと文字をもたない 1 ノードを子としてもつ。一方、自身が文字を持たないノードは子をもたない葉ノードであり、葉ノードにはそのノードのキーと非打鍵手入力の文字の組合せから成る単語が格納されている。例えば、先祖に ‘i’ と ‘y’ をもつ葉ノードには正規表現で “[a-gq-tv-xz]\*i[a-gq-tv-xz]\*y[a-gq-tv-xz]\*” と表される単語 (“variety” や “sideways” など) が格納される。

提案システムでは根を始点として、打鍵手の文字入力があった時に、その文字をキーとする子ノードに移動する。その時点では、そのノードを根とする部分木に含まれる全ての葉ノードに格納されている単語が入力された候補となる。その後、アルファベット以外のキーが入力された時のノードと直接結ばれている子の葉ノードに格納されている単語が入力単語の候補となる。つまり、打鍵手での入力 ‘k<sub>0</sub>’, ‘k<sub>1</sub>’ がさらた時点では “[a-gq-tv-xz]\*k<sub>0</sub>[a-gq-tv-xz]\*k<sub>1</sub>[a-z]\*” を満たす単語が候補となり、その後にアルファベット以外のキーが入力されると、“[a-gq-tv-xz]\*k<sub>0</sub>[a-gq-tv-xz]\*k<sub>1</sub>[a-gq-tv-xz]\*” を満たす単語が候補として提示される。

しかし、打鍵手の入力のみでは人名や商品名などの固有名詞や、新語といった辞書に載って

いない単語が入力出来ないという問題があるため、提案システムでは2章で述べた Wearable Half Keyboard と同様の反転入力機構を備えている。具体的には、あらかじめ設定した反転キーを押しながらのキー入力は、図1に示すようにキーボードを左右反転した位置の非打鍵手入力に置き換えられる。例えば反転キーを押しながらの‘j’や‘x’の入力は‘f’や‘.’の入力に置き換えられる。また、この入力機構は辞書に載っていない単語を入力する以外に、候補の絞り込みとしても使用できる。提案手法では、非打鍵手に入力が偏った単語を入力すると、候補の絞り込みが十分に出来ずに大量の単語の候補を提示することがある。そこで、反転入力機構を用いて非打鍵手の入力がされた時は、その非打鍵手入力を含まない単語を候補から除外する。よって、打鍵手入力‘k<sub>0</sub>’、‘k<sub>1</sub>’の入力後に非打鍵手入力‘k<sub>0</sub>’があった場合は、“[a-gq-tv-xz]\*k<sub>0</sub>[a-gq-tv-xz]\*k<sub>1</sub>[a-gq-tv-xz]\*k<sub>0</sub>[a-z]\*”を満たす単語が候補となる。具体例を挙げると、‘k’、‘y’、‘o’の入力の後に反転入力機構を用いて‘s’の入力があった場合は、“keyboard”は候補として提示されるが、“keyboard”は候補から外される。

### 3.4 優先度計算

提案手法では候補の単語の提示順を決定するために、打鍵手の打鍵間隔からその間に含まれる非打鍵手の打鍵数（以下、非打鍵手入力数）を推測して優先度を計算する。

優先度計算のために、システムはユーザの打鍵速度のデータとして、非打鍵手入力数  $n$  に応じた打鍵手の打鍵間隔  $t_n$  を取得する。例えば、右手を打鍵手とした時は“keyboard”という入力から、‘k’と‘y’、‘y’と‘o’の間にはそれぞれ‘e’と‘b’の一文字が含まれるので、それらの打鍵間隔を  $t_1$  に記録し、‘o’とスペースキー間には‘a’、‘r’、‘d’の3文字が含まれるので、その打鍵間隔を  $t_3$  に記録する。ここで、非打鍵手入力が5回以上連続で続くことは非常に稀で、十分なデータが集まらないため  $n$  の上限を4に設定する。以下、 $n \geq 5$  のデータは全て  $n = 4$  に置き換えて扱う。そして、 $n$  毎の打鍵速度の平均  $t_{\mu_n}$  と標準偏差  $t_{\sigma_n}$  を計算し、非打鍵手入力数の推測に利用する。

非打鍵手入力数の推測には正規分布の確率密度関数の高さを用いる。打鍵手での打鍵間隔  $x$  が与えられた時の非打鍵手入力数が  $n$  の確率は、以下のように平均  $t_{\mu_n}$ 、分散  $t_{\sigma_n}^2$  の正規分布の確率密度関数  $f_n$  の高さ  $f_n(x)$  で表される。

$$f_n(x) = \frac{1}{\sqrt{2\pi}t_{\sigma_n}} \exp\left(-\frac{(x-t_{\mu_n})^2}{2t_{\sigma_n}^2}\right)$$

そして、打鍵間隔  $x$  に対して全ての  $n$  について  $f_n(x)$  を計算した後、以下に示す式で打鍵間隔  $x$  の時の非打鍵手入力  $n$  の重み  $p_n(x)$  を計算する。

$$p_n(x) = \frac{f_n(x)}{\sum_{i=0}^4 f_i(x)} + \alpha$$

ここで、 $\alpha$  を加えることで、タイピングにもたついて打鍵間隔が平均値から大きくずれた場合の影響を抑える。

最後に全ての打鍵間について  $p_n(x)$  を計算した後、単語ごとの優先度  $W$  を計算する。単語の優先度の初期値  $W_{base}$  は Web ページのクローリング中の出現回数とする。そして、打鍵手入力が“ $k_1 k_2 \dots k_n k_{n+1}$ ” ( $k_{n+1}$  は非アルファベットキー) で  $k_i$  と  $k_{i+1}$  の打鍵間隔が  $x_i$  であった時、“[非打鍵手入力  $k'_0$  文字] $k_1$ [非打鍵手入力  $k'_1$  文字] $k_2 \dots k_{n-1}$ [非打鍵手入力  $k'_{n-1}$  文字] $k_n$ [非打鍵手入力  $k'_n$  文字][非アルファベットキーの入力]”で表される単語の優先度  $W(k'_0, k'_1, \dots, k'_n)$  は以下のように計算される。

$$W(k'_0, k'_1, \dots, k'_n) = W_{base} * \prod_{i=1}^n p_{k'_i}(t_i)$$

なお、 $k_1$  の前の打鍵間隔  $x_0$  は考慮しない。これは、 $x_0$  には  $k'_0$  の入力に要した時間に加えて、直前の単語の選択時間が含まれ、安定せず推測できないためである。最後に、辞書に載っていない単語の入力を可能にするために、入力された通りの文字列“ $k_0 k_1 \dots k_{n-1} k_n$ ”を候補の末尾に追加する。

ユーザは提示された候補の中からあらかじめ設定した変換キーあるいは上下の矢印キーで自分が入力した単語を選択して入力する。また、入力単語が候補の上位に表示されていない時の探索を容易にするために、候補の提示をアルファベット順や文字数順に並び替える機能を実装した。入力単語の決定は変換キー以外のキーの入力があった時に決定されるため、入力単語が最上位に提示されれば、通常キーボードの入力時に対して追加の動作を必要としない。最後に選択された単語を元にユーザの打鍵速度データを更新する。

## 4. 評価

### 4.1 提案手法の有効性

提案手法の有効性を評価するために、提案手法を通常キーボード上で用いた際の入力速度を調べた。その際、打鍵手はキーボードを打鍵し、非打鍵手は机の上を空打ちする。評価では一般の辞書から取得した30284語の基本形と、WikipediaのWebクローリングによって取得した128980語を結合した140317語の出現回数を格納した辞書ファイルを用いる。

本評価ではメーラでの利用を想定し、以下の5つのショートメッセージを使用した。

表 1 入力に要した時間  
Table 1 Time taken to input

| 被験者 | 文章  | 要した時間(秒) |       |       |       |          |       |
|-----|-----|----------|-------|-------|-------|----------|-------|
|     |     | 通常キーボード  | 提案手法  |       | 左手鍵盤  | ソフトキーボード | めくり方式 |
|     |     |          | 左     | 右     |       |          |       |
| A   | (1) | 6.5      | 22.9  | 31.6  | 43.5  | 21.4     | 28.5  |
|     | (2) | 5.5      | 20.5  | 14.1  | 22.5  | 12.1     | 20.5  |
|     | (3) | 7.5      | 18.6  | 21.8  | 27.7  | 26.5     | 32.3  |
|     | (4) | 7.6      | 23.7  | 17.7  | 30.8  | 20.3     | 36.1  |
|     | (5) | 11.5     | 25.4  | 48.8  | 48.0  | 54.9     | 54.0  |
| B   | (1) | 9.8      | 35.9  | 37.7  | 39.7  | 17.8     | 36.0  |
|     | (2) | 8.8      | 20.4  | 22.0  | 25.4  | 18.9     | 23.3  |
|     | (3) | 10.3     | 20.1  | 43.7  | 33.7  | 20.2     | 43.3  |
|     | (4) | 12.3     | 24.5  | 36.3  | 38.2  | 19.7     | 41.6  |
|     | (5) | 13.9     | 36.2  | 75.3  | 45.4  | 27.8     | 55.5  |
| C   | (1) | 7.1      | 23.9  | 16.3  | 34.0  | 22.1     | 26.6  |
|     | (2) | 5.3      | 11.3  | 15.0  | 17.0  | 19.0     | 21.4  |
|     | (3) | 7.7      | 18.3  | 19.0  | 27.7  | 25.6     | 26.4  |
|     | (4) | 7.4      | 20.8  | 21.9  | 24.7  | 31.7     | 35.8  |
|     | (5) | 10.1     | 22.8  | 39.5  | 36.5  | 46.5     | 58.2  |
| A   | 合計  | 38.6     | 111.1 | 134.0 | 172.5 | 135.2    | 171.4 |
| B   |     | 55.1     | 137.1 | 215.0 | 182.4 | 104.4    | 199.7 |
| C   |     | 37.6     | 97.1  | 111.7 | 139.9 | 144.9    | 168.4 |
| A-C |     | 131.3    | 345.3 | 460.7 | 494.8 | 384.5    | 539.5 |

- (1) Thank you for inviting me today.
- (2) Do you have time now?
- (3) Please tell me the meeting place.
- (4) It is not convenient for me today.
- (5) I overslept and will be late getting to the office.

これらの文章を通常のキーボード、提案手法、Half Keyboard 方式、スマートフォンのタッチパネル上のソフトキーボード、めくり方式の携帯電話で入力した。なお、提案手法については左右それぞれのハーフキーボードで入力を行い、10 分程度の入力から作成した打鍵速度データを用いる。Half Keyboard 方式では左手のみで入力を行い、左半分のキー配列はそのまま使用し、スペースキーが押されている間は図 1 に示すキー配列に置き換える。被験者は 20 代男性 3 人で被験者 A,B は右利き、被験者 C は左利きである。被験者はいずれも毎日キーボードを使って文字入力を行っている。それぞれの入力に要した時間を表 1 に示

す。表 1 にはそれぞれの文章について最も素早く入力できたものから順に 3 つの入力方法に、  
、  
、  
をつけている。

表 1 から、全ての被験者の全ての文章において通常のキーボードの入力時間が一番短くなり、合計入力時間は被験者 A, B, C それぞれで 38.6 秒, 55.1 秒, 37.6 秒となった。通常のキーボードを除いた各被験者ごとの合計入力時間を調べると、被験者 A は左手用ハーフキーボード 111.1 秒, 右手用ハーフキーボード 134.0 秒の順で、被験者 B はスマートフォンのソフトキーボード 104.4 秒, 左手用ハーフキーボード 138.1 秒の順で、被験者 C は左手用ハーフキーボード 97.1 秒, 右手用ハーフキーボード 111.7 秒の順に短くなった。

提案手法とソフトキーボードの入力速度は、本実験ではユーザの好みや入力する単語によって前後することが分かった。両者の最も大きな違いはタッチタイピングの実現にある。本評価に使用したソフトキーボードは打鍵の際に振動を与えることでユーザは打鍵感を得るが、ホームポジションの概念がないため、指先の感覚では自分の指の位置を把握できない。それに対して、提案システムでは通常のキーボードを用いる時と同様に、指先でキーをなぞることでホームポジションを把握してタッチタイピングができる。ウェアラブルコンピューティング環境では、街中で入力を行うので入力デバイスを注視しながらの入力は危険を伴う。よって、タッチタイピングが出来る提案手法はよりウェアラブルコンピューティング環境向きの入力方法だと言える。

提案手法のみに着目すると、全 15 文中で 12 文で左手用ハーフキーボードの方が入力時間が短くなり、被験者ごとの合計入力時間も左手用ハーフキーボードが右手用ハーフキーボードよりも短い時間で入力を終えている。この特徴は被験者の利き手に関わらずに見られた。その理由として、キーボードの左半分に使用頻度が高い文字が偏っていることが考えられる。文献 10) によると、アルファベットの文字のうち使用割合が高い文字は 'e', 't', 'a' であり、その 3 文字の出現割合は順に 12.702%, 9.056%, 8.167% である。また、キーボードの左半分にあるアルファベットの使用割合の合計は 58.7% である。提案手法では打鍵手の入力が多くなると入力単語の候補が絞り込まれ、スムーズな入力の実現できる。

#### 4.2 実機プロトタイプを用いた習熟実験

前節の評価実験から提案手法の有効性が確認できたため、実機のプロトタイプを作成した。作成したプロトタイプを図 5 を示す。プロトタイプは前節で入力速度が高かった左手用ハーフキーボードを採用しており、キーボードの 'h' と 'y' の位置に '、' と '、' を、'6' と [caps lock] キーの位置に [Backspace] キーと [Enter] キーを配置している。さらに、スペースキーの左のキー (図 5 では [alt, opt] キー) は候補の順送り選択キー、さらにその左のキー (図 5



図 5 作成した実機プロトタイプ

Fig.5 A snapshot of prototype system



図 6 実機プロトタイプの使用例

Fig.6 An usage example of prototype

表 2 習熟実験結果

Table 2 The result of proficiency study

| 被験者 | 入力速度 (WPM) |      |      |      |      |      |
|-----|------------|------|------|------|------|------|
|     | キーボード      | 1 日目 | 2 日目 | 3 日目 | 4 日目 | 5 日目 |
| A   | 59.6       | 17.0 | 16.3 | 19.6 | 18.8 | 16.2 |
| B   | 52.2       | 13.9 | 13.5 | 14.5 | 14.7 | 18.3 |
| C   | 20.2       | 8.4  | 11.1 | 12.4 | 12.6 | 12.9 |
| D   | 21.4       | 8.4  | 10.7 | 11.8 | 12.5 | 12.4 |

では [win, cmd] キー) は候補の逆送り選択キー、一番左下のキー (図 5 では [ctrl] キー) は候補の並び替えキー、その上のキー (図 5 では [shift] キー) は反転キーに設定した。プロトタイプは図 6 のように大腿部に装着した利用を想定している。

作成したプロトタイプを用いて提案手法の習熟の評価を行った。被験者は 20 代男性 4 名で被験者 A, B はキーボード入力の上級者、被験者 C, D は中級者である。期間は 5 日間で 1 回の実験は 10 分程度の練習と 15 分程度のデータ採取から成る。なお、入力する文章はメーラを想定した 200 語前後のものを用いて、反転入力による絞り込みは用いずに入力させた。以下に結果を示す。

入力速度:各被験者のキーボードと提案手法の入力速度を表 2 に示す。表中の WPM(Word Per Minute) とは文字入力デバイスの入力速度の単位で 1 分間に何単語 (一般的に 5 文字を 1 単語とする) を入力できるかを表す。また、各被験者のキーボード入力速度と比べた提案手法の入力速度の割合を図 7 に示す。表 2 から、被験者 A, B は 1 日目から入力速度が安定しており、被験者 C, D のキーボード使用時と同等の入力速度を実現しているが、自身のキーボード入力の約 30%の入力速度に収まった。一方、図 7 から、被験者 C, D は 3 日

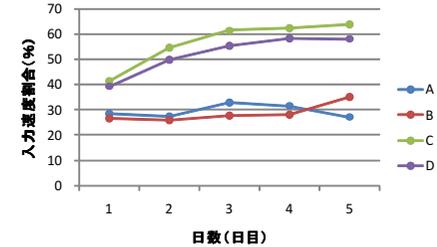


図 7 キーボードと比べた提案手法の入力速度割合

Fig.7 The input speed ratio of proposal method to keyboard

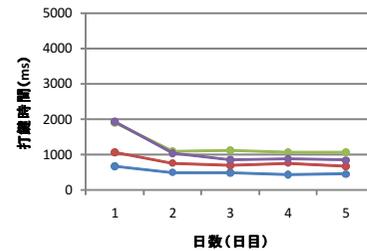


図 8 打鍵時間の推移

Fig.8 The result of keying time

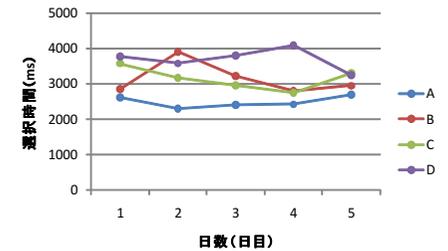


図 9 選択時間の推移

Fig.9 The result of selecting time

目あたりで入力速度が安定し、5 日目には 1 日目の 1.5 倍、自身のキーボード入力速度の約 60%の入力速度を実現した。

入力時間の内訳:入力時間の内訳として、打鍵時間 (打鍵手の最初の打鍵からアルファベット以外のキーを押して候補選択に入るまでの時間) と選択時間 (候補選択に入ってから次の単語の打鍵手の最初の打鍵を行って候補を決定するまでの時間) を調べた。各被験者の 1 単語あたりの平均打鍵時間と平均選択時間の推移をそれぞれ図 8 と図 9 に示す。図 8 から、全ての被験者の打鍵時間は 2 日目には収束している。これから提案手法の利用に伴う非打鍵手の空打ち動作はすぐに習得できると言える。一方で、図 9 から選択時間は安定せず、打鍵時間と比べて非常に大きくなっており、最終日では平均で入力時間の 80.7%を選択時間が占めている。これから提案手法の入力速度の上昇には選択時間の短縮が絶対条件であると言える。

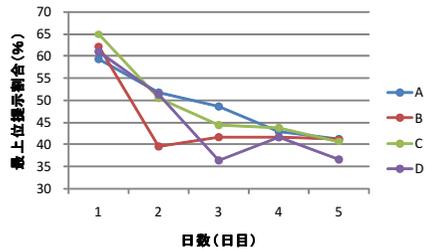


図 10 最上位提示割合の推移

Fig.10 The result of top-presented rate

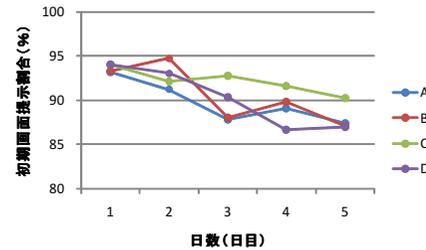


図 11 初期画面提示割合の推移

Fig.11 The result of initial-presented rate

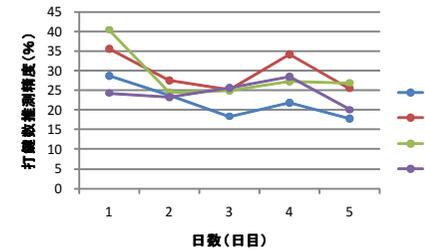


図 12 打鍵数推測精度の推移

Fig.12 The accuracy of keying number

入力単語の提示位置：選択時間が安定せずに大きくなった原因を明らかにするために入力単語の提示位置を調べた。各被験者における入力単語が最上位に提示された割合と初期画面（10個の単語を提示）に提示された割合をそれぞれ図10と図11に示す。図10から、1日目では入力単語が平均61.9%の割合で最上位に提示されていたにも関わらず5日目では平均39.9%まで下がっていることが分かる。同様に、図11から、1日目では入力単語が平均93.6%の割合で初期画面に提示されていたにも関わらず5日目では平均87.9%まで下がっていることが分かる。この原因として入力単語の候補数が多かった、あるいは、打鍵数の推測を間違えて入力単語の優先度が下がった、という2種類の原因が考えられる。

非打鍵手入力数の推測：各被験者における非打鍵手入力数の推測の精度の推移を図12に示す。ここで、精度は実際の手入力数  $i$  の重み  $p_i$  が  $p_0$  から  $p_4$  の中で最大になった時を正解として求めた。我々が行った通常キーボードの使用時の打鍵数予測アルゴリズムの予備実験では90%以上の精度が出ていたが、図12より、本評価時の非打鍵手入力数の推測精度は50%を下回る結果となった。また、打鍵データの蓄積によって推測精度が向上すると予測していたが、実際には全ての被験者において1日目と5日目を比べた時に推測精度が下がっていることが分かる。そこで、5日目における全被験者の推測の計算結果を表3に示す。表3より、過半数の打鍵間で非打鍵手入力数が0と推測されている一方で、約10%が4と推測されている。これは、外れ値が打鍵速度データの中に取り込まれることによって引き起こされていると考えられる。具体的には、打鍵のもたつきなどが挙げられる。そのような外れ値が打鍵速度データの中に取り込まれた際には非打鍵手入力数は4と推測され、同時に  $t_n$  の平均を大きくあげる。その後、スムーズな打鍵がされた際に非打鍵手入力数は0と推測される。打鍵速度データの収集時に外れ値を外す機構と、非打鍵手入力数の推測時に

表 3 5日目の打鍵数推測結果  
Table 3 The result of keying number (5th day)

| 実際の打鍵数 \ 推測された打鍵数 | 0     | 1     | 2    | 3    | 4     |
|-------------------|-------|-------|------|------|-------|
| 0                 | 57.4% | 25.2% | 3.0% | 1.5% | 12.9% |
| 1                 | 65.8% | 17.6% | 2.5% | 1.7% | 12.4% |
| 2                 | 61.7% | 19.8% | 3.6% | 3.6% | 11.4% |
| 3                 | 52.2% | 22.6% | 6.5% | 5.4% | 13.4% |
| 4                 | 60.0% | 17.1% | 8.6% | 2.9% | 11.4% |

外れ値を計算から除外する機構の実装によってこの問題を解消できると考えられる。

## 5. 考 察

本章では提案手法の入力速度の向上のための改良案を述べる。前章における評価より、入力速度の向上には単語の選択時間の短縮が必須である。以下で、具体的な方法を考察する。

### 5.1 利用環境に応じた辞書の利用

今回の評価では Wikipedia 上の単語を学習した辞書ファイルを用いたが、ユーザの利用シーンに合わせて様々な辞書ファイルを用意することで入力速度の向上につながる可能性がある。具体的には、本稿の評価の際にはメーラを想定した文章を用いたので、“I” や “my” などの一人称や “you” や “your” などの二人称が多く出現した。これらはいずれも左手の打鍵数が少ない単語なので、打鍵間隔による絞り込みが十分にできず、これらの提示位置は優先度の初期値（今回は Web クローリング時の出現回数を用いた）が大きく影響した。その優先度の初期値を多数のメール文章から求めることで、メール文章作成がスムーズになる。



図 13 ユーザインタフェースの改良案  
Fig. 13 Improvement plan of UI

あるいは、ユーザの好みや単語の使用履歴を考慮した辞書ファイルを用いることで、提案システムの入力速度は格段に向上すると考える。

### 5.2 単語候補の提示方法の改良

提案システムでは図 3 に示すように、打鍵があるたびに候補を絞り込み、その時点で入力されている可能性がある単語を優先度順に提示している。そのため、例えば打鍵手の入力  $k_0k_1k_2$  と非打鍵手の入力  $k'_0k'_1$  から成る単語 “ $k_0k_1k'_0k_2k'_1$ ” の入力をする時、 $k_0$  と  $k_1$  を入力した時点で “ $k_0k_1k'_0k_2k'_1$ ” が最上位に提示されると、ユーザがそれに影響されて  $k_2$  の入力をしないまま候補の選択に入ろうとアルファベットキー以外のキーを押してしまい、目的の単語が候補から外れてしまうケースが存在した。この問題には以下のような解決策が考えられる。

- (1) 単語の入力終了時まで単語の提示をしない。
- (2) アルファベット以外のキーを押した時の候補の絞り込み排除する。
- (3) 図 13 のようにユーザインタフェースの文字色で未入力文字を区別する。

(1) の方法では入力の途中でユーザが意図する単語が適切に重み付けされているかが分からず、(2) の方法では短い単語を入力するときの候補数が大幅に増えてしまう。また (3) の方法では単語が読み取りづらくなってしまうと考えられる。この問題を解決することで、入力単語が候補から外れてしまったことに気づき、再び単語を入力しなおすまでの時間を短縮できる。

### 5.3 予測変換アルゴリズムの適用

今回は、打鍵間隔と辞書ファイルの作成時の出現回数という 2 つの要素のみを用いて単語の優先度計算を行った。しかし、実際にユーザが入力する単語は、英語の構文上の制約で直前の単語の影響を大きく受けて、さらにメールの返信であれば参照元メールの影響も受けている。それらに対してユーザの入力単語の予測を行うアルゴリズムはこれまで数多く提案されてきており<sup>11)</sup>、現在では携帯電話を中心に幅広く普及している。提案システムでもこ

れらのアルゴリズムを組み込むことで、候補の絞り込みや優先度の計算の精度が向上する。

## 6. おわりに

本研究では、文字入力デバイスの小型化の一つの方法として、既存のキーボードのキー配列をそのままにしながらキーボードを左右に分割し、その片側の入力情報のみから残りの入力を推測することで文字入力を行う手法を提案した。提案手法によってユーザビリティを低下させることなく、ユーザがすでに体得しているキーボード入力の能力を活かして、キーボードの大きさを半減できる。評価によって提案手法の有効性が確認でき、実機のプロトタイプの評価から入力速度を改善する余地があることが分かった。今後は入力速度の改善と日本語入力への対応を予定している。

## 参考文献

- 1) 塚本昌彦, 板生知子: ウェアラブルコンピューティングとユビキタスサービス, オペレーションズ・リサーチ: 経営の科学, Vol. 49, No. 4, pp. 210–216 (2004).
- 2) 塚本昌彦: ウェアラブルコンピューティング, 基礎講座ユビキタスコンピューティング, ヒューマンインタフェース学会誌, Vol. 5, No. 1, pp. 27–32 (2003).
- 3) Y. Liu, X. Liu, and Y. Jia: Hand-Gesture Based Text Input for Wearable Computers, In *Proceedings of the ICVS 2006 IEEE International Conference on Computer Vision Systems*, p. 8 (2006).
- 4) 久米祐一郎, 島田正文: 腕装着型カメラと反射指標を用いた装着型入力デバイスの検討, 映像情報メディア学会誌, Vol. 60, No. 2, pp. 249–253 (2006).
- 5) CUT key によるこそ: <http://www.cutkey.jp/>.
- 6) Matias Wearable Half Keyboard: <http://www.halfkeyboard.com/wearable/index.html>.
- 7) Twiddler 2: <http://www.handykey.com/>.
- 8) E. Matias, I. S. MacKenzie, and W. Buxton: Half-qwerty: Typing with One Hand using Your Two-Handed Skills, *Companion of the CHI 1994 Conference on Human Factors in Computing Systems*, pp. 51–52 (1994).
- 9) 小谷賢太郎, 堀井健, 法岡泰樹: テンキーパネルを用いた打鍵認証システムの構築と評価, ヒューマンインタフェース学会論文誌, Vol. 7, No. 1, pp. 149–156 (2005).
- 10) R. E. Lewand: Cryptological Mathematics, Mathematical Society of America.
- 11) T. Masui: POBox: An Efficient Text Input Method for Handheld and Ubiquitous Computers, In *Proceedings of the HUC 99 International Symposium on Handheld and Ubiquitous Computing*, pp. 289–300 (1999).