

Drag&Flick : タッチスクリーンを用いた 視覚障がい者向け文字入力方式

青木良輔^{†1} 橋本遼^{†1} 瀬古俊一^{†1} 片岡泰之^{†1}
井原雅行^{†1} 渡辺昌洋^{†1} 小林透^{†1}

近年、タッチスクリーンを持つ携帯端末が普及し、視覚障がい者からもこの携帯端末を利用したいという要望がある。しかし、視覚障がい者は、触覚を介して指がスクリーンに触れている位置を正確に把握できないため操作が困難である。本論文では、タッチスクリーン上での文字入力操作に焦点を当て、指がタッチスクリーンに触れた任意の位置から指を離さずにスライドさせることで文字入力を可能とする操作方式『Drag&Flick』を提案する。提案方式は、タッチスクリーン操作において視覚的なフィードバックに依存しない文字入力方式である。提案方式の文字入力精度を向上させるために、指の移動方向を精度よく検知するためのデザイン、および指の移動方向の識別アルゴリズムを実装し、指の移動方向の識別精度と提案方式の操作性を確認するための被験者実験を行った。その結果、提案したデザインおよびアルゴリズムに効果があることを確認できた。

Drag&Flick: A Text Entry Method on a Touch Screen for Visually Impaired People

RYOSUKE AOKI^{†1} RYO HASHIMOTO^{†1} SHUNICHI SEKO^{†1}
YASUYUKI KATAOKA^{†1} MASAYUKI IHARA^{†1}
MASAHIRO WATANABE^{†1} TORU KOBAYASHI^{†1}

The popularity of mobile devices with touch screens is spreading. The visually impaired are also demanding that they be able to use these devices. However, it is difficult for them to use the devices without adequate haptic feedback. This paper focuses on chording text entry by the visually impaired, and proposes "Drag&Flick" as a text entry method. This method allows the people to move a single finger from its arbitrary position on the screen and to remember the movement required to input a character without visual feedback. The validity of this input design and a recognition algorithm are proven in an experiment and interviews.

1. はじめに

近年、スマートフォンやタブレットのようなタッチスクリーンを持つ携帯端末（以後、タッチスクリーン端末と呼ぶ）は、大きな情報閲覧画面と直感的なタッチスクリーン操作により幅広い年齢層に受け入れられ、急速に普及している。日常業務でPCを利用するコンピュータリテラシの高い視覚障がい者の中にも、最新の機能が搭載されるこのタッチスクリーン端末を自力で使いこなしたいという利用意欲の高い人がいる[1]。しかしながら、タッチスクリーン端末の大半の製品は物理的キーボードを排除しており、かつタッチスクリーンは平面であるため、触覚を介してタッチスクリーン上の指の位置を正確に把握するのが困難である。健常者は視覚を介して指の位置とタッチスクリーン上のキー（以後、ソフトウェアキーと呼ぶ）の位置を把握できるが、視覚障がい者は把握できないので、タッチスクリーン端末を容易に操作できない。そこで、音声フィードバックを用いたタッチスクリーン端末のアクセシビリティ向上の取り組みがなされている。

市場に登場しているタッチスクリーン端末の中には音声読み上げソフトウェアが搭載されている製品がある[2]。

これは、指が触れたソフトウェアキー/アイコンの名前を読み上げ、そして指が触れたソフトウェアキー/アイコンに対してダブルタップをすると、そのソフトウェアキー/アイコンが選択される方式である。しかしながら、選択可能なソフトウェアキー/アイコンが増えると、1つずつ音声で確認しないと選択できないため時間がかかる。特にタッチスクリーン端末に標準搭載されているソフトウェアキーボードによる文字入力操作においては選択対象となるソフトウェアキーが多いため、入力対象のソフトウェアキーを探す時間がかかる。ゆえに、音声フィードバックに頼るだけでなく、入力に必要な指の動きを身体が感覚的に覚え、その指の動きを高精度に実行可能な操作方式を取り入れていく必要がある。

本論文では、文字入力操作に焦点を当て、上記の課題を解決するために、身体が感覚的に覚えやすい指の動きを用いた操作方式として『Drag&Flick』を提案する。提案方式では、指がタッチスクリーンに触れた位置から8つの方向（上、右上、右、右下、下、左下、左、左上）のいずれかに、触れた指を規定距離以上移動させると、移動させた方向に対応した子音が選択される。さらに指を離さずに8つの方向（上、右上、右、右下、下、左下、左、左上）のいずれかに指を規定距離以上移動させると、移動させた方向

^{†1} 日本電信電話株式会社 NTT サービスエボリューション研究所
NTT Service Evolution Laboratories, NTT Corporation

に対応した母音が選択される。そして、最後に指をタッチスクリーンから離すと、文字が一意に決定される。ユーザは入力に必要な指の動きを実行できる範囲でタッチスクリーンの任意の位置から操作を開始できる。指の移動距離を正確に身体が覚える必要がなく、8方向に分けて指を移動させるだけであるため、ユーザが入力操作に必要な動きを精度よく容易に覚えることができる。提案方式では、音声フィードバックも併用するが、これはあくまで操作を覚えるための補助として利用される。本研究では、操作対象のタッチスクリーン端末をスマートフォンとする。視覚障がい者にはタブレットの利点である大きな画面の効果が少なく、むしろ軽くて持ち運びしやすい端末が好まれるからである。そこで、スマートフォンを用いて、提案方式に実装した指の移動方向の識別アルゴリズムの精度に関する被験者実験を行い、インタビューを通して提案方式の操作性について検証した。

以下、2章では、関連技術とそれらの課題について述べ、3章では提案方式の詳細について述べる。4章では、被験者実験を通して、提案方式の操作性を検証する。最後に本論文のまとめを行う。

2. 関連研究

視覚障がい者がタッチスクリーン端末で利用可能な文字入力方式には、音声認識を用いた文字入力方式と、音声読み上げ付きソフトウェアキーボードを用いた文字入力方式、スクリーン上のジェスチャによる文字入力方式、および、点字の概念を利用した文字入力方式がある。

音声認識による文字入力方式を用いたサービスがタッチスクリーン端末に標準搭載されており[3][4][5]、ダウンロードして利用可能な音声入力アプリもある。しかしながら、実用化されたとはいえ、音声認識精度が不十分なケースや音声入力ができないという問題がある。そのため、音声認識のみに頼らない方式の検討が必要である。

音声読み上げ付きソフトウェアキーボードを用いた文字入力方式には、すでに実用化されているものがある[2][6][7]。[2]の方式ではソフトウェアキーボードの配置によって多少操作は異なるが、指で触れたソフトウェアキーの種類を音声で確認し、その後にダブルタップする動作が操作の一部に含まれている。しかし、音声を聞きながら指の位置を合わせる作業は時間がかかる。また、小さいソフトウェアキーを正確にダブルタップする動作は、指が離れたときに触感が失われる中で正確な動きが求められる。[6]は、9個のソフトウェアキーをスクリーンの4角や4辺と中心に配置し、キーを2回連続でタップすることでアルファベットを入力する方式である。この方式ではタッチスクリーン端末の4角、4辺は位置の把握がしやすいことを利用している。しかし、端末の4角、4辺を確認するためには、端末を把持している手そのものを移動させる必要があ

る。この移動が頻繁に起こると操作に時間がかかる。また、文字入力方式には文字を入力するだけでなく、SPACEキーやBackSpaceキー（以後、BSキーと呼ぶ）、各種記号（例えばピリオド、カンマ、！、？など）も必要であり、9個のソフトウェアキーではタップする回数が増える。[7]はスクリーンを3つの領域（2種類の母音選択領域、1種類の子音選択領域）に分割し、母音選択領域に指を触れさせ、フリックすることで母音選択を行い、次に子音選択領域に指を触れさせ、フリックすることで子音選択し、文字を一意に決定する文字入力方式である。しかし、領域選択時に間違った領域を選択することや指の可動範囲を考慮していないために、指が触れた位置に応じてフリックする方向を正確に行うことが容易ではないという問題がある。

一方、タッチスクリーンを活かしたジェスチャによる文字入力方式に関する研究例もある[8][9]。[8][9]は、タッチスクリーンに指で文字を描き、その文字を認識することで文字を入力する方式である。[8]では、ひらがなを対象とし、[9]では、各アルファベットを一筆書きで描けるようにカスタマイズされた形状の記号を対象としている。しかしながら、視覚障がい者は健常者と比べて文字や記号を精度よく描くことが困難である[10]。[10]では、視覚障がい者は記号や文字を描くよりもタップ（タッチスクリーンに指が触れた直後に指を離す動作）、フリック（ある方向に指を移動させながら指を離す動作）のような指をある方向に移動させる動き、2本指特有の動き（例えば、ピンチイン/アウトなど）を好む傾向があることを示している。ゆえにドラッグ（スクリーンに触れたまま指を移動させる動作）やフリックを使う方がよいと考えられる。

文献[11]の方式では、次の2つの段階を経てアルファベットを入力する。まずスクリーンの中心を原点とした8分割のパイメニューが表示され、指は中心の円領域の内部に触れてから分割された領域のいずれかに指を離さずに移動させることで、指が移動した領域に対応した文字グループ（例えば、“ABC”など）を選択する。次に、スクリーンに文字グループの文字がリスト表示され、入力したい文字の項目に指を移動させ、その項目上で指を離すと文字が一意に入力される。しかしながら、ソフトウェアキーを押す動きの代わりにジェスチャを用いているが、音声読み上げ付きソフトウェアキーボードと同様に前もって固定された複数の領域の中から1つの領域を選択するのは音声フィードバックなしでは困難である。実際、この技術にはVoiceOver[2]による音声読み上げを利用している。

次に、視覚障がい者特有の文字入力方式として点字の概念を利用した文字入力方式がある[12][13][14][15]。点字は長方形を6つに分割し、その分割された各領域の突起の有無によって作られるパターンによって文字が一意に定められたものである。[12][13]では、スクリーンの4角と左右の辺の中心に領域を作り、入力したい文字の点字の突起が対

応する領域を指で通過させることによって点字を作り文字を入力する。[14]では、点字を左右に分けて、左右それぞれの突起のパターンを 0~7 までの数字を入力することで指定する。つまり、数字を 2 回選択することで一意に文字が決定される方式である。数字の選択にはスクリーン上にパイメニューを表示し、中心の円領域に指を触れさせ、その後、指を 8 つの方向のいずれかに移動させて数字に対応したメニュー項目を選択する。そして中心の円領域に指を戻すことを繰り返すことで数字を選択する。この方式では、スクリーンにメニューが固定されており、指がスクリーンに触れる位置および指の距離を正確に動かす負担が生じる。また、そもそも点字を習得する必要がある。

3. 提案方式

3.1 基本アイデア

2 章で述べた従来技術を踏まえ、タッチスクリーンを用いた視覚障がい者向け文字入力方式に求められる要件を以下に示す。

- I. スクリーンの任意の位置から指の動きを開始できること
- II. 入力に必要な指の動きを容易に覚えることができること
- III. 個人差のある、指の移動距離が入力精度に影響を与えないこと
- IV. 文字以外に頻繁に使われる記号（句読点、濁点、半濁点、？）や専用キー（SPACE キー、BS キー、小文字化）が文字入力と同様の操作で入力できること
- V. タップする回数が少ないこと

上記の要件を満たすために、著者らは『Drag&Flick』という方式を提案する。提案方式における基本操作は、以下の手順になる。

- STEP1 スクリーンに指が触れる。
 STEP2 スクリーンに指が触れた位置から 8 つの方向（上、右上、右、右下、下、左下、左、左上）のいずれかに指を移動させる。
 STEP3 STEP2 で指を規定距離以上移動させた後、さらに指を離さず 8 つの方向（上、右上、右、右下、下、左下、左、左上）のいずれかに指を移動させる。
 STEP4 STEP3 で指を規定距離以上移動後に指をスクリーンから離す。指が離れたときに入力が行われる。
- STEP1 と STEP2 はドラッグと類似しており、また、STEP3 と STEP4 は同時に行うとフリックになるため、『Drag&Flick』と名付けた。図 1 は指を右に移動させた後に、指を右上に移動させた場合の例を示す。

提案方式は指の移動方向を入力とするので、スクリーンの任意の位置から操作を開始できる（要件 I）。これにより音声フィードバック付きソフトウェアキーボードのようにソフトウェアキーの位置を、音声を用いて探す手間が解消

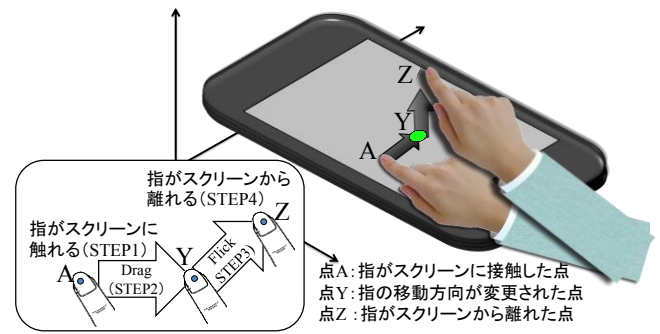


図 1 Drag&Flick

される。

要件 II を満たすためには音声フィードバックや視覚フィードバックがなくても容易に実行可能な動作を組み合わせることが必要である。この条件は、QWERTY 配列の物理的なキーを持つキーボードのタッチタイピング操作による文字入力方式 [16]、親指による十字キーを押下する動作と手首のひねる動作を組み合わせたリモコン操作による文字入力方式 [17] のような手元の動きを見ずに可能な文字入力操作を調査したときに得られる。タッチスクリーン端末では、ユーザは触覚を介してタッチスクリーン上の指の位置を把握することが困難である。そこで、聴覚、視覚そして触覚に頼らずに行える動作として、タッチスクリーン上で指を 8 方向のいずれかに移動させる動作に着目し、この動作を 2 回連続で行うことで要件 II を満足させる。ただし、音声フィードバックは、入力方法を覚えるための補助として、そして、入力結果を伝えるための補助として利用する。詳細は 3.5 節で述べる。

次に、提案方式は指の移動方向を 2 度識別して入力とする。そのため、各方向識別において、方向計測に必要な 2 つの基準点をどのように検知するかが方向識別の精度に影響する。1 回目の方向識別に必要な 2 つの基準点は、指がスクリーンに触れた点（図 1 の点 A、図 2 (i) の点 A）と、指の移動距離が前もって決められた距離（図 2 (i) の円領域 C1 の半径）を超えたときに計測された指の位置（以後、点 B と呼ぶ）とする。2 回目の方向識別に必要な 2 つの基準点は、1 回目に識別された方向と異なる方向に指が移動した時に計測された位置（図 1 の点 Y、図 2 (iv) の点 Y）と、その点からの指の移動距離が前もって決められた距離（図 2 (ii) の円領域 C2 の半径）を超え、かつ指がスクリーンから離れた位置（図 1 の点 Z、図 2 (iv) の点 Z）とする。しかしながら、指の移動距離は個人毎に異なり、また同じ人でも入力毎に異なるため、点 B と点 Y は、必ずしも一致しない。図 2 (iv) の点 Y を精度よく検知可能な方向識別アルゴリズムを提案することでこの問題を解決する（要件 III）。1 回目に識別された方向と同じ方向に指が移動してから指がタッチスクリーンから離れた場合、点 Y の位置が検知できず、1 回目と 2 回目の方向は同じと判断さ

れる（詳細は3.2節で述べる）。

上記の動作で選択可能な数は8方向×8方向の64通りである。日本人はQWERTYキーボードのローマ字入力でもわかるように子音と母音の組み合わせによって文字が決定されることに慣れているため、提案方式においても1回目の方向識別で子音を選択し、2回目の方向識別で母音を選択することが好ましい。しかし、子音の数は10種類あるため、最初に選択できる方向が8種類では不足である。また、頻繁に利用される記号（句読点、濁点、半濁点、！、？）や専用キー（SPACEキー、BSキー、小文字化）も文字入力と同様に入力できるとよい。入力の数を増やすために、タップしないでSTEP1～STEP4の一連の指の動きをするのと、シングルタップした直後にSTEP1～STEP4の一連の指の動きをする場合とで選択可能な文字を切り替えるようにする。これにより、1回目の指の移動で選択可能な数を16通り、2回目の指の移動で選択可能な数は128通りとなる（要件IV）。

本方式により、1文字を入力する時に、タッチスクリーンをタップする回数は多くても1回に抑えることができる。（要件V）。シングルタップのときに指が触れた位置と、シングルタップ直後に指が触れた位置が異なっても提案方式は任意の位置から指の動きを開始できるため、タップ時の指の位置を気にしなくて良い。

3.2 指の移動方向の識別方法

本節では、指の移動方向の識別方法について述べる。

指がスクリーンに触れると、図2(i)のように触れた指の位置（点A）を中心とした円領域C1と、円領域C1の外側で、点Aを通過する直線によって分割された指の方向を識別する8つの領域（図2(i)の領域P0～P7）を作成する。図2を点線で表現しているのはスクリーンに表示する必要がないことを示すためである。円領域C1の枠の内側から外側に指が移動したときに計測された位置（点B）で1回目の方向識別を行う。

次に、1回目の移動方向と2回目の移動方向が異なる場合の2回目の方向識別について述べる。3.1節で述べたが、入力毎に移動距離が異なるため、指の移動方向を変更した位置（点Y）は点Bと必ずしも一致するとは限らない。この点Yを精度よく決定するために、規定距離（図2(ii)のC2の半径）以上指が移動する度に1回目の方向と同じ方向に指が移動しているのかを確認する。

まず、1回目の方向識別の直後に、図2(ii)のように点Bを中心とした円領域C2と、円領域C2の外側で、点Bを通過する直線によって分割された指の方向を識別する8つの領域（図2(ii)の領域R0～R7）を作成する。次に、円領域C2の内側から外側に指が移動したときに計測された位置で、1回目の方向識別結果と同じ方向（図2の場合、領域R0に相当）ならば、この位置を点Y候補とし、円領域と指の方向を識別する8つの領域を作成する。この処理

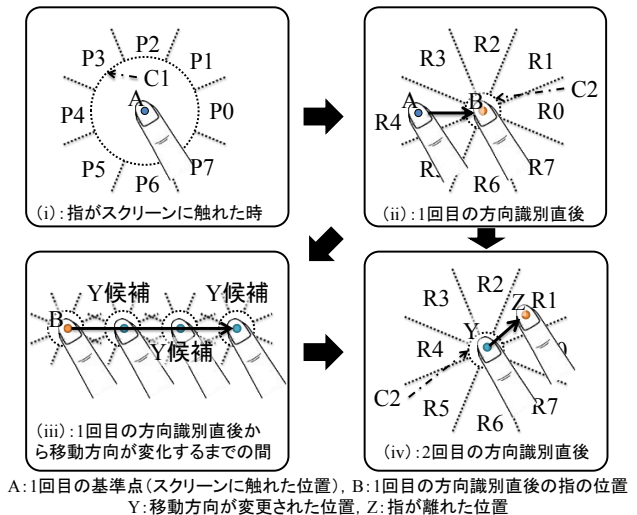


図2 指の移動方向を識別するフロー

を、1回目の方向識別結果と同じ方向に指が移動する限り点Y候補を更新する（図2(iii)）。そして、1回目の方向と異なる方向の領域に移動したら、直近の点Y候補を点Yとする。これにより、精度よく点Yの位置が決定される。また、点Yが決定される前に指が離れたら2回目の方向識別結果は1回目の方向識別結果と同じとする。

この方向識別アルゴリズムを用いることで入力する度に異なる指の移動距離が識別精度に与える問題が解決可能となる。円領域C1と円領域C2の大きさについては4章で検討する。

3.3 日本語の配置

3.1節で述べたように、提案方式では、1回目の移動方向で子音を選択し、2回目の移動方向で母音を選択し、10種類の子音をSTEP2の操作で選択できるように、STEP1の前にシングルタップを行うかどうかで選択可能な子音を変更できるようにした（図3）。子音と母音の割り当て方によって、操作の覚えやすさと操作精度に影響を与えるため、図2のP0～P7領域に対応する子音、そして図2のR0～R7領域に対応する母音の割り当て方について検討する。

多くの人々が慣れ親しんでいる50音表は右から左に子音が並んでいることが多いため、図3のように提案方式の子音も右から左に並べる。

次に、母音の配置は、覚えやすさと操作しやすさの両方の観点から配置を決定する必要がある。予備実験において、提案方式の文字の配置に関する調査を行った[18]。子音の選択方向に関わらず母音の配置が固定されている場合と、子音の選択方向に対して相対的に母音を配置する場合とどちらが覚えやすいかインタビューしたところ、前者の方が好ましいという結果が得られた。そこで、提案方式の母音の配置を1回目の方向識別の指の移動方向に関わらず図4のように固定した。この配置にした理由は2つある、1つ

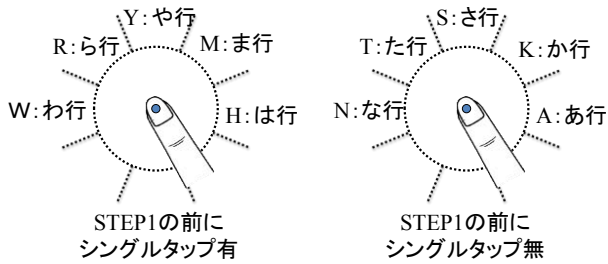


図3 子音の配置

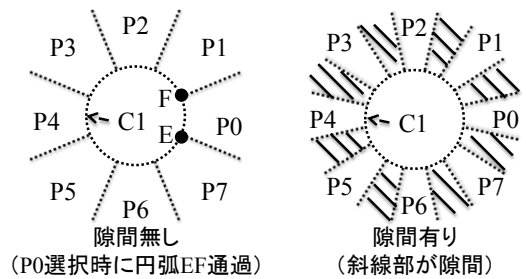


図5 隙間の領域

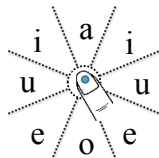


図4 母音配置

表1 日本語配置の一例

	シングルタップ有り									シングルタップ無し							
	P7	P6	P5	P4	P3	P2	P1	P0		P7	P6	P5	P4	P3	P2	P1	P0
R0	、	Space	小	を	る	ゆ	む	ふ	R2	、	BS	小	ぬ	つ	す	く	う
R1	°	Space	小	一	り	!	み	ひ	R1	°	BS	小	に	ち	し	き	い
R2	"	Space	小	わ	ら	や	ま	は	R0	"	BS	小	な	た	さ	か	あ
R3	°	Space	小	一	り	!	み	ひ	R1	°	BS	小	に	ち	し	き	い
R4	、	Space	小	を	る	ゆ	む	ふ	R2	、	BS	小	ぬ	つ	す	く	う
R5	°	Space	小	-	れ	?	め	へ	R3	°	BS	小	ね	て	せ	け	え
R6	&	Space	小	ん	ろ	よ	も	ほ	R4	&	BS	小	の	と	そ	こ	お
R7	°	Space	小	-	れ	?	め	へ	R3	°	BS	小	ね	て	せ	け	え

小:小文字変換キー, BS:BSキー, -:ハイフン, -:パー

は、50音表の母音は上から下に並んでいるということ。もう1つは、サイズの小さいスクリーンから指が外側に出ないように、1回目の指の移動方向に応じてユーザが適宜使い分けられるようにするためである。

さらに、入力エラー時に文字を削除するためのBSキーや小文字変換キーは文字入力時に頻繁に利用され、これらのキーが選択しやすい方が好ましい[17]。そこで、子音選択時に使われていない領域にこれらのキーを対応させ、すぐに入力できるようにする。以上のことを踏まえ、提案方式の日本語入力の配置例を表1のようにした。

3.4 子音選択の精度向上

ユーザは指を動かす際、直線移動していると感じていても必ずしも直線になっているとは限らない。また、[19][20]のようなペン操作と異なり、タッチ操作はタッチスクリーンに接触する領域が広いので、指の位置検出にズレが生じる。ゆえに1回目の方向識別の検知に使われる円領域C1の大きさは子音選択精度と、提案方式の操作の可動範囲に影響する。

C1が大きいと、図2(i)のP0~P7の各領域を選択する時に通過するC1の円弧(P0の場合、図5の円弧EF)が大

きくなるため、子音選択精度が高くなる。しかし、C1を大きくしすぎると、指の移動量が増え、ユーザの負担が増える。さらにスマートフォンのスクリーンは小さいため、2回目の方向識別に必要な指の移動量が確保できない場合が生じる。

一方、C1の大きさを小さくすると、P0~P7の各領域を選択する時に通過するC1の円弧の大きさが小さくなる。その結果、選択対象の領域に隣接する領域を通過してから選択対象の領域に指が移動することが生じる場合がある。提案方式では、最初に選択された領域によって子音を選択されるので、誤入力につながる。

この問題を解決するために、子音を選択する領域間に隙間を作り(図5の右図)、その領域内に指が有る場合、子音は選択されず、その領域から子音を選択できる領域に移動した場合に子音の選択を行うようにした。また、その隙間で指を離れた場合は、文字が入力されないようにした。図5の左図のように隣接する領域の境界付近での誤入力を防ぐことができるので、ユーザの指の移動が曖昧でも高精度な子音選択が可能となる。なお、この隙間は子音選択時のみに適用する。

3.5 音声フィードバック

提案方式では、入力するのに必要な指の動きを身体が覚え、各指の動きに対応した文字を覚えると音声フィードバックに頼らなくても操作できる。しかし、指の動きを覚え、各指の動きが対応する入力内容を覚えるまでは、音声フィードバックに頼る必要がある。そこで、提案方式では、2カ所で音声フィードバックする。1つ目は、子音選択時(図2の点Bに指がある場合)に選択された子音を音声で出力する。2つ目は、指がスクリーンから離れる前に選択された子音と母音の組み合わせで一意に選択された文字を音声で出力する。しかし、指が離れるタイミングを検知できないので、点Yから規定距離D以上離れた時に音声を出力することとする。

これにより、初心者は母音選択時に指を離す前に選択した文字を確認でき、母音選択に失敗していたらそのまま隣接領域に指を動かすことで選択する文字を修正することが可能となる。例えば母音のiを選択する目的で仮にaを選

択した場合、図6のように時計回り/反時計回りに指を移動させることで母音の*i*を選択できる。ただし、指を離していないことが条件となる。

4. 提案方式の評価

4.1 実験目的

本章では、視覚障がい者を対象に行った実験について述べる。実験では、提案方式を用いて文字を入力してもらい、エラーの回数を計測する。その上で、3.2節で述べた指の移動方向が変化した点を検知するアルゴリズムと、3.4節で述べた子音を選択する領域間の隙間の効果を検証する。

4.2 被験者

実験には、3名の視覚障がい者（男性3名）に被験者として協力してもらった。各被験者の特徴を表2に示す。被験者は全員目が見えず、白杖を用いて歩行している。日常生活でPCを用いているため、QWERTYキーボードによるローマ字入力に慣れている。

4.3 実験装置

実験では、タッチスクリーン端末として第4世代の iPod Touch。このタッチスクリーン端末のスクリーンのサイズは7.5cm×5.1cm（解像度1024pixels×768pixels）であり、無線LANを介してサーバPCに接続した。文字入力のアプリケーションはHTML5とJavascriptで実装し、iPod Touchの標準ブラウザ（Safari）でサーバPCにアクセスすることで利用する方式とした。サーバPCはOSとしてWindows 7を搭載し、CPUとしてクロック2.4GHzのIntel Core i5を搭載している。文字入力操作時に流れる音声はサーバPCから再生した。この理由は、ブラウザから音声が出力するまでに時間のタイムラグがあり、操作性が低下するからである。

次に、3章で述べた識別アルゴリズムのパラメータの設定値を示す。C1の半径は100pixels（人差し指のおおよそ半分の大きさ）とした。C2の大きさは、小さいほど点Yの検知精度は高いが、小さすぎると点Yの誤検知につながる。この実験では、C2の半径は、予備実験を行った中で高精度に点Yを検知した値20pixelsとした。同様に予備実験の結果から、図5の隙間の角度は20度とし、選択文字の音声フィードバックする点Yからの規定距離DはC1の半径と同じとし、シングルタップしてから500ms以内にSTEP1を行ったとき、『は』～『わ』行に子音を変更した。

4.4 実験手順

被験者には、操作手順の説明を10分間行った。タッチスクリーン端末の持ち方は、左手でその端末を把持し、右手の人差し指で操作してもらった。各被験者には、ローマ字入力のように子音と母音の組み合わせで文字が入力でき、各子音と各母音の配置を時計の針の方向（例えば、子音のSは12時の方向）を用いて説明した。子音と母音の配置は3.3節で述べた配置を用いた（表1参照）。「あ行」～「な行」の子音の選択から「は行」～「わ行」の子音の選択へ

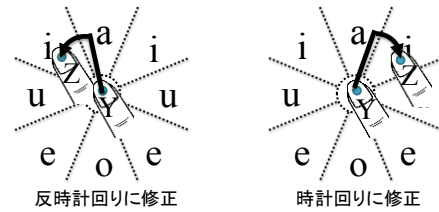


図6 母音選択時の修正方法

表2 被験者の特徴

ユーザ名	ユーザA	ユーザB	ユーザC
年齢	34才	37才	47才
障がいの程度	視覚障がい1種1級	視覚障がい1種2級	視覚障がい1種1級
視力	左目:光覚 右目:0	左目:0.02 右目:0.01	左目:0 右目:0
視覚障がいになった年齢	3才 (現視力24才)	24才	22才
PCの利用頻度	日常生活、仕事でPCを利用		
利用携帯電話	物理的なキーと音声読み上げ機能を持つ携帯電話		
らくらくフォンの利用経験	経験あり (7年半使用)	経験あり (10年使用)	経験あり (10年使用)
スマートフォンの利用経験	経験なし	経験なし	短時間 操作した程度
点字を覚えている	駅の手摺りやエレベータの点字を読む程度	覚えていない	覚えていない

の切り替えには3.1節で述べたようにシングルタップの有無を用いた。音声フォードバックは3.5節で説明した通りに行った。指がスクリーンに触れ、1回目の方向識別を行ったと同時に選択された領域の結果をサーバPCに自動的に送信し、サーバPCから音声を出力した。「か行」であればK（ケー）と出力した。次に2回目の方向識別を行ったと同時に、1回目の方向識別の結果と2回目の方向識別の結果を自動的にサーバPCに送信し、一意に決定された文字に関する音声サーバPCから出力した。ただし、1回目の方向識別と2回目の方向識別が同じ場合は、指が離れたときに一意に決定される音声が出力される。操作説明終了後、10分間自由に操作してもらい、実験前に今回入力する50種類の文字（表1のひらがなの『あ』～『ん』と記号—、-（ハイフン）、?、!）を各文字3回ずつ入力してもらった。

実験手順について説明する。実験の指示者が文字を1文字音声で指定し、各被験者はその指定された文字を入力した。入力時に子音を間違えたら、母音の選択を行わず指を離すように指示した。指示する文字はランダムで決定した。1試行あたり50回の入力を行い、合計7試行（350文字）の入力を行ってもらった。全体の実験時間は、約1時間であった。

4.5 実験結果と考察

図7に実験結果のエラーの回数と種類をユーザごとに示す。エラーは3種類発見され、子音の選択エラー、母音の選択エラー、タップエラーであった。タップエラーとは、

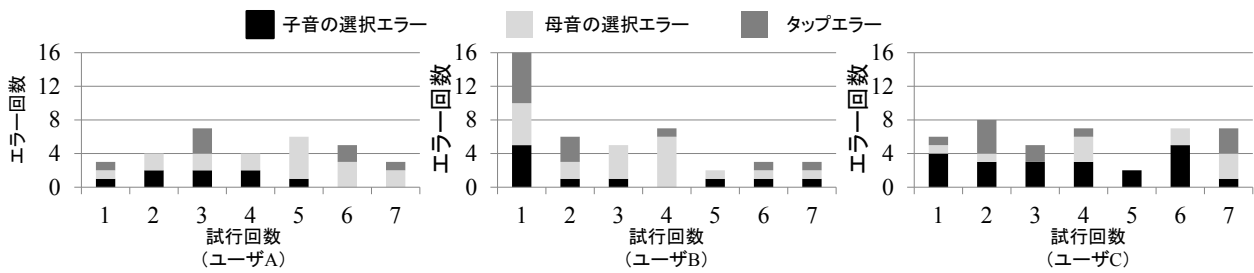


図7 各ユーザーの各試行のエラー回数

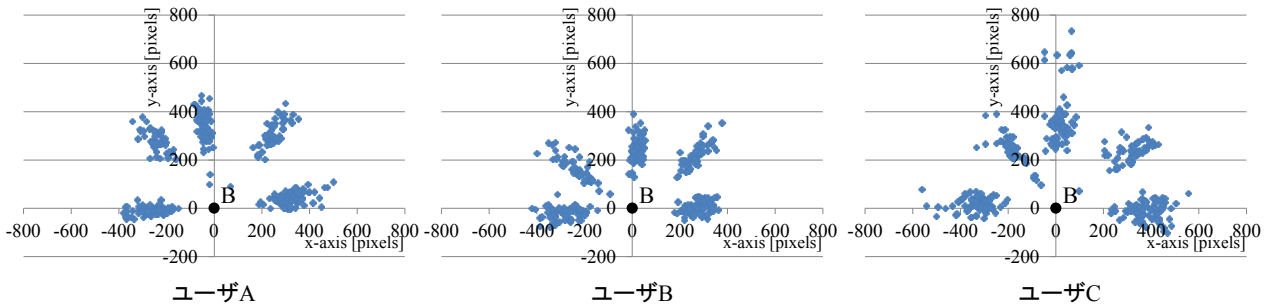


図8 各ユーザーの1回目の方向識別をした点(点B)に対する2回目の移動方向へ変化した点(点Y)

シングルタップしてから500ms以内にSTEP1の状態に遷移することがうまくできずに生じたエラーである。ユーザーAとユーザーBはタッチスクリーン端末を利用していないため、タップエラーが1試行~3試行で目立つ。しかし、4試行以降ではこのタップエラーの回数は減少した。また、3試行まで記号入力時のエラーが含まれていたが、4試行以降は含まれなかった。インタビュー時に全ユーザーは、最初は記号の位置が覚えられなかったが、入力していく最中で覚えられたと述べた。つまり、4試行以降、記号の配置が覚えられなかったことによる入力エラーが減少したと考えられる。

図8に1回目の方向識別を行った点(図2の点B)に対する1回目の移動方向から2回目の移動方向へ変化した点(点Y)の座標を被験者ごとに示した。方向が変化した点(点Y)と1回目の方向識別した点(点B)は離れる傾向がある。被験者間で比較すると、各自、指の移動距離が異なることが示されている。また、各被験者に着目したときに、1入力毎に、指の移動距離が異なることが示されている。ゆえに、2回目の移動方向を計測するのに、移動方向が変化する点を小刻みに検知するアルゴリズムは有効であることが示された。

図9に各試行で各被験者が子音選択時に3.4節で述べた領域間の隙間を通過してから子音に対応する領域を選択する回数を示す。全体の平均で15%の割合で隙間を通過した。ユーザーAとユーザーCにおいて、試行回数が5試行以降になると隙間を通過する回数が増加した。インタビュー時に、ユーザーAとユーザーCは、操作に慣れてくることにより後半

は、動きが粗くなったと述べたことから、これが隙間の通過回数を増やした原因と考えられる。しかし、図7の子音の選択エラー回数は5試行以降、大きな変化はない。ゆえに、選択する領域間に隙間をつくることにより子音の選択エラーの回数を改善する効果があると考えられる。

図10に提案方式を用いたときの各被験者の入力速度を示す。平均入力速度は1500ms、つまり1分間あたり40文字入力相当になった。

4.6 提案方式の操作性に関する検証

提案方式の操作性について検証するために被験者にインタビューを行った。

検証方法として、実験終了後、各被験者に視覚障がい者向けの文字入力として導入されている音声読み上げ付き10キーのソフトウェアキーボードのフリック操作方式(以後、フリック操作方式と呼ぶ)[2]の操作方法を10分間説明し、実際に10分間操作してもらった。その上で、提案方式とフリック操作方式の使いやすい点、使いにくい点、覚えやすい点、覚えにくい点を比較してもらった。その結果を表3に示す。

提案方式の利点は、3章で述べた視覚障がい者向け文字入力方式を実現するための要件と一致しており、提案方式がその要件を満たしていることを示している。また、提案方式の覚えやすさは、50音表と対応できる点があげられ、3.3節で述べた配置が効果的であったことを示す。

今後は、表3で述べられた欠点に対して改善する必要がある。記号の配置の覚えにくさに対する解決案の1つとして、濁音、半濁音の文字も子音と母音の組み合わせで入力

するようにし、小文字や SPACE キー、BS キーはマルチタッチ操作で入力する方法があげられる。また、2 回目の指の移動の精度の向上にかかる時間がどの程度かを調査する必要がある。

5. おわりに

本論文では、タッチスクリーンを用いた視覚障がい者向け文字入力方式を提案した。提案方式の文字入力精度を向上させるためのアルゴリズムを実装し、そのアルゴリズムの効果を被験者実験で確認した。また、提案方式の操作性についてインタビューを行い、提案方式が従来方式の問題を解決する 1 つの方式であることが示された。今後は、提案方式を用いた文字入力アプリケーションを実現していくために、漢字変換操作、文字種類の変更操作、入力した文字を決定する操作などを組み入れたデザインを検討していく予定である。

本論文の内容は、総務省の先進的 ICT 国際標準化推進事業「次世代ブラウザ技術を利用した災害時における情報伝達のための端末間情報連携技術」の受託研究の成果である。

参考文献

- [1] 橋本遼, 渡辺昌洋, 浅野陽子, “タッチパネル式インタフェースを持つ情報端末のアクセシビリティ,” 2012 信学技報, vol. 112, no. 45, HCS2012-15, pp. 109-114, May 2012.
- [2] Apple Computer Inc., “視覚に障害のある方に,” <http://www.apple.com/jp/accessibility/iphone/vision.html>, 参照 Apr.4,2012.
- [3] Apple Computer Inc., “Siri”, <http://www.apple.com/jp/ios/siri>, 参照, Oct, 2012.
- [4] Google Inc., “音声検索,” <https://play.google.com/store/apps/details?id=com.google.android.voicesearch>, 参照, Oct, 2012.
- [5] NTT docomo, “しゃべってコンシェル,” http://www.nttdocomo.co.jp/service/information/shabette_concier/
- [6] J. Sánchez and F. Aguayo, “Mobile Messenger for the Blind,” ERCIM'06 Proceedings of the 9th conference on User interfaces, pp.369-385, 2007.
- [7] 深津佳智, 志築文太郎, 田中二郎, “携帯情報端末のタッチパネルにおけるアイズフリーな片手文字入力システム,” 2012 情報処理学会研究報告, Vol.2012-HCI-149, No.5, pp. 1-8, 2012.
- [8] CELL レグザリモコン用タッチパッドのストローク認識技術 http://www.toshiba.co.jp/tech/review/2010/04/65_04pdf/a09.pdf
- [9] H. Tinwala and I. S. MacKenzie, “Eyes-free Text Entry on a Touchscreen Phone,” Science and Technology for Humanity (TIC-STH) 2009 IEEE Toronto International Conference, pp. 83-88, 2009.
- [10] S. K. Kane, J. O. Wobbrock and R. E. Ladner, “Usable Gestures for Blind People: Understanding Preference and Performance,” CHI '11 Proceedings of the 2011 annual conference on Human factors in computing systems, pp. 413-422, 2011.
- [11] M. Bonner, J. Brudvik, G. Abowd and W. K. Edwards, “No-Look Notes: Accessible Eyes-Free Multi-Touch

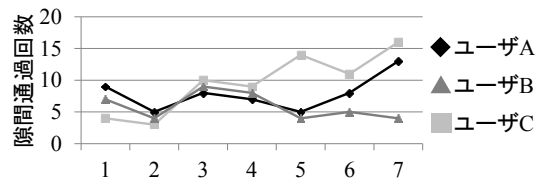


図9 隙間の通過回数

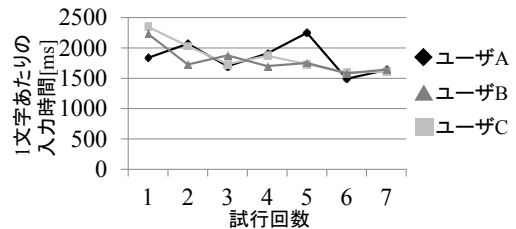


図10 1文字あたりの入力時間

表3 インタビューによる従来技術と提案技術の比較

		従来方式	提案方式
使い易さ	利点	なし	<ul style="list-style-type: none"> • タップ数が少ない • 指の移動角度のみを覚えればよい • 画面上の細かい位置をきにしなくてよい
	欠点	<ul style="list-style-type: none"> • キーの位置を探すのが手間 • 入力にかかる時間が長い • キー配置が覚えられない • 直感的でない 	<ul style="list-style-type: none"> • 2回目の指の移動方向が苦手 • シングルタップしたら即座にSTEP1に遷移するのが苦手
覚え易さ	利点	携帯電話のキー配置と同じ	記号以外の配置は覚えやすい
	欠点	ソフトウェアキーが多い	記号の配置が覚えにくい

Text Entry,” Pervasive'10 Proceedings of the 8th international conference on Pervasive Computing, pp. 409-426, 2010.

- [12] 牛田啓太, 阿佐見聡士, 長谷川貞夫, “タッチパネルを用いた点字マッピング一筆書き文字入力方式の提案と開発,” 2011 信学技報, vol. 110, no. 418, WIT2010-77, pp. 57-61, 2011.
- [13] IPPITSU 一筆書き点字入力 <http://video.mainichi.co.jp/viewvideo.aspx?Movie=48227968/48227968peevee392442.flv>
- [14] IPPITSU 8/2R 文字入力方法の解説 <http://www.youtube.com/watch?v=jBk0tQnFGFM>
- [15] B. Frey, C. Southern and M. Romero, “BrailleTouch: Mobile Texting for the Visually Impaired,” 6th International Conference UAHCI 2011, pp. 19-25, 2011
- [16] M. V. Longley, “Type-Writer Lessons for the Use of Teachers and Learners Adapted to Remington’s Perfected Type-Writers,” Cincinnati, 1882.
- [17] R. Aoki, M. Ihara, A. Maeda, M. Kobayashi, S. Kagami, “Expanding Kinds of Gestures for Hierarchical Menu Selection by Unicursal Gesture Interface,” IEEE Trans. on Consumer Electronics, vol. 57, no. 2, pp. 731-737, 2011.
- [18] 青木良輔, 橋本遼, 瀬古俊一, 片岡泰之, 井原雅行, 渡辺昌洋, 小林透, “マルチタスクリーンを用いた視覚障がい者向け文字入力方式,” 2012 HCG シンポジウム, pp. 153-158, 2012
- [19] 丸山泰史, 五十嵐健夫, “日本語入力のためのフィードバック付き一筆書き手法,” インタラクシオン 2003, pp. 55-56, 2003.
- [20] 佐藤大介, 志築文太郎, 三浦元樹, 田中二郎, “ペンによるメニュー選択に基づく日本語入力手法,” 日本ソフトウェア科学会第20回大会論文集, pp.1-5, 2003.