

# Fix and Slide：文字列全体の移動によるキャレットのポインティング手法

鈴木 健司<sup>1,a)</sup> 岡部 和昌<sup>1</sup> 坂本 竜基<sup>2</sup> 坂本 大介<sup>3</sup>

受付日 2018年4月19日, 採録日 2018年11月7日

**概要：**本稿は、スマートフォンやタッチディスプレイにおいてポインティングを行う際、ポインタや指示したいオブジェクトが指に隠れて正確な操作が難しくなるという問題、いわゆる Fat Finger Problem に対して、ポインタそのものの位置を指で移動させるのではなく、ポインタが存在する画面全体を移動させ相対的にポインタの位置指定を行う手法について述べる。本研究では特にスマートフォン等の小型携帯端末における文字列選択の場面を例題として提案手法の実装と評価を行う。評価は、本手法と iOS が標準で提供している手法それぞれで、iOS 上で表示される文章の文字列選択を行えるプロトタイプを作成したうえで、比較評価実験を行った。その結果、提案手法を初めて操作したユーザであっても従来の手法と作業実行時間に有意な差が確認されなかった、もしくはフォントサイズが小さい場合には提案手法の方が有意に早いことが確認され、また通常の手法よりも理解しやすく使いやすいことが示唆された。

**キーワード：**キャレット, テキストカーソル, 文字選択, モバイル端末, 可動式テキストフィールド

## Fix and Slide: Caret Navigation with Movable Background

KENJI SUZUKI<sup>1,a)</sup> KAZUMASA OKABE<sup>1</sup> RYUUKI SAKAMOTO<sup>2</sup> DAISUKE SAKAMOTO<sup>3</sup>

Received: April 19, 2018, Accepted: November 7, 2018

**Abstract:** We present a concept of using a movable background to navigate a caret on small mobile devices. The standard approach to selecting text on mobile devices is to directly touch the location on the text that a user wants to select. This is problematic because the user's finger hides the area to select. Our concept is to use a movable background to navigate the caret. Users place a caret by tapping on the screen and then move the background by touching and dragging. In this method, the caret is fixed on the screen and the user drags the background text to navigate the caret. We compared our technique with the iPhone's default UI and found that even though participants were using our technique for the first time, average task completion time was even faster than Default UI in the case of the small font size and got a significantly higher usability score than Default UI.

**Keywords:** Caret, text cursor, text selection, mobile device, movable background

### 1. はじめに

個人が日常的に使用する情報端末としてのフロントエンドは、デスクトップ機における web ブラウザからスマート

フォンにおけるアプリへとシフトしてきた [1]。これらの端末が受信するメディアは、高解像度な画像や映像にまで及び始めているが、ニュースやメールサービス等の文字を主体とするものもいまだ根強い。しかし、スマートフォンに代表される小型携帯端末の主要インタフェースであるタッチパネルは文字列操作に向いていない。なぜならば、小型のモニタに映し出される小さい文字群に対して、それを指示する指先はあまりに大きく、いざ正確なポインティングをしようとする際には「指が操作対象を隠してしまう」い

<sup>1</sup> ヤフー株式会社  
Yahoo Japan Corporation, Chiyoda, Tokyo 102-8282, Japan

<sup>2</sup> 電駆ビジョン株式会社  
Denqvision Inc., Nerima, Tokyo 176-0006, Japan

<sup>3</sup> 北海道大学  
Hokkaido University, Sapporo, Hokkaido 060-0814, Japan

a) kensuzuk@yahoo-corp.jp

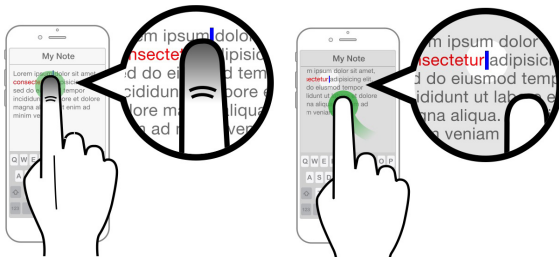


図 1 一般的なキャレット操作手法 (左) と提案手法 (右) : ユーザは文字列全体を背景としてドラッグすることで、任意の場所にキャレットを移動することができる。

Fig. 1 Standard caret navigation (left) and Fix and Slide (right). User moves the background text to navigate a caret instead of moving the caret itself.

いわゆる “Fat Finger Problem” が発生するためである [2].

この問題は文字列の選択を行う場合にも生じる。一般的な小型携帯端末において文字列選択をする場合、ユーザはキャレットと呼ばれる文字列用のカーソルを選択開始位置に移動させる。続いて長押し等で文字列選択モードにした後、キャレットを選択終了位置まで移動させる (図 1 左)。しかし、Fat Finger Problem が 1 度のタップで正確なキャレットの配置を困難にさせるため、選択開始位置に配置するまでに何度かタップやドラッグを繰り返す必要がある。同様に、文字列選択の終了位置までキャレットを移動させる場合も 1 度で正確な操作が難しく、ここでも調整のための操作を繰り返す必要がある。つまり、一般的に小型携帯端末における素早く正確なポインティングや文字選択はデスクトップ機よりも困難である。

本稿では、小型携帯端末における文字列選択操作について、対象となる文字列を直接接触してキャレットを操作する直接操作手法ではなく、文字列全体を背景として移動させキャレット位置を間接操作する Fix and Slide と呼ぶポインティング手法を提案する (図 1 右)。Fix and Slide では、画面のどこをドラッグしても文字列全体を移動させることができる。このときキャレットはドラッグ開始時に位置する画面上の点  $p$  で固定され、ドラッグ操作が終わり指を画面から離れたとき、 $p$  に位置する文字と文字の間にキャレットがくっつき、キャレットを含む文字列全体は元の位置に戻る。このドラッグ操作中はキャレットが指で隠れないため Fat Finger Problem は発生せず、適切な位置にキャレットを移動させることが可能となる。

この Fix and Slide の有用性を調査するため、プロトタイプシステムを iOS 上で実装し、調査実験を通して作業実行時間計測やユーザビリティ調査を行った。まず、日常的に iOS を使用している成人に対して量的な比較検討を行い [3]、その後、小型携帯端末上での文字列ポインティングを行ったことがない児童に対して質的な観察を実施した。本稿ではこれら実験の結果も報告する。

## 2. 関連研究

Fat Finger Problem は小型携帯端末のインタフェースに関する研究においてよく知られた問題であり、これまでに様々なアプローチの研究が行われてきている。Shift はその中でも最も有名な研究であり、この手法では画面を触っている指の下の領域、すなわちユーザが画面を見る際に指が隠ってしまう領域が「ふきだし」として指の周囲に表示される [2]。本手法は現在の iOS 等に搭載されている。TapTap は画面をタップすることにより、タップされた領域を拡大して表示するポップアップを利用した手法である [4]。他にもタッチ操作可能な機器に対してオフセットカーソルを利用する手法が存在するが、直接操作手法よりも操作時間がかかっていた [5]。これまでは小型携帯端末の「表面」のタッチスクリーンを利用するものであったが、反対の「裏面」を活用した研究も存在する [6]。Hasan らもモバイル端末の裏面を用いた研究を行っている [7]。これらの研究はポインティング操作において Fat Finger Problem を回避するための手法であり、キャレット操作や文字列選択に注目した研究ではなかった。

正確なポインティングに関する手法についても研究が行われてきている。これまでに高精度にタッチ操作をセンシングする技術 [8] や、確率モデルを採用してタッチされるポイントを予測する手法 [9]、マルチタッチスクリーン上で両手を利用して正確に対象の選択を行う技術の提案 [10]、タブレットを握る手で画面上の離れた位置をポインティングする手法の研究 [11] が行われてきている。最近になって Eady らによって小型携帯端末上のキャレット操作に関する手法について提案 [12] があったが、端末の周囲に曲げセンサーを取り付ける必要がある等、端末のハードウェア的な拡張が必要であった。また、形態素解析を用いた日本語文字列選択手法についての提案 [13] もされてきているが、それぞれ本研究の目的とは大きく異なる。また小型携帯端末のモニタ内での仮想スティック [14] やジェスチャを用いた文字列選択手法 [15] も提案されている。これらは小型携帯端末でのキャレット操作や文字列選択の手法であるが、本研究は「文字列の背景を動かすことによるキャレット操作手法および文字列選択手法」の検討を行う点で特徴的である。

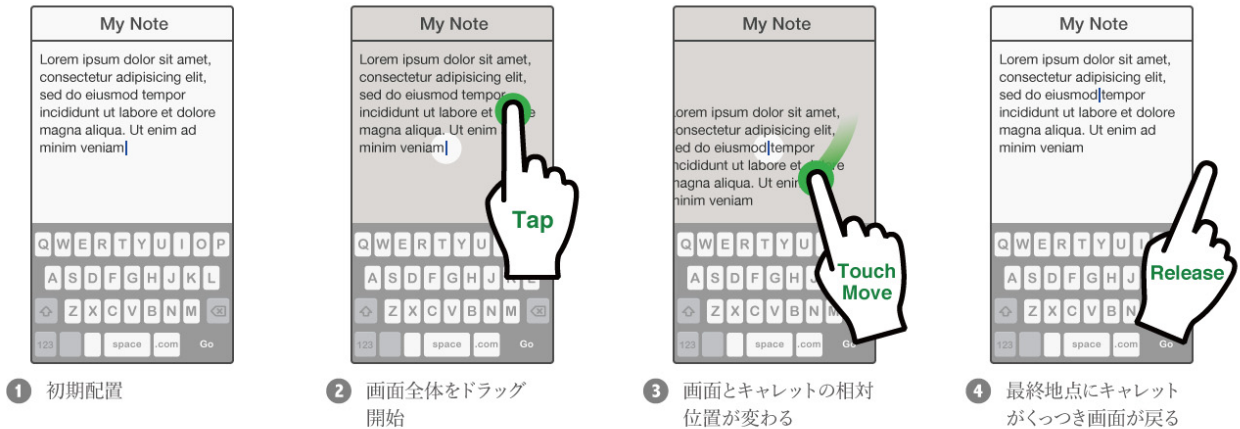
## 3. Fix and Slide

本稿で提案する Fix and Slide の主たるコンセプトはキャレットの移動や文字列選択手法として文字列全体を背景として操作するという点である (図 2)。以下でそれぞれの操作について説明する。

### 3.1 キャレット操作

小型携帯端末上でキャレット操作を行う際に問題となる

キャレット移動



文字列選択

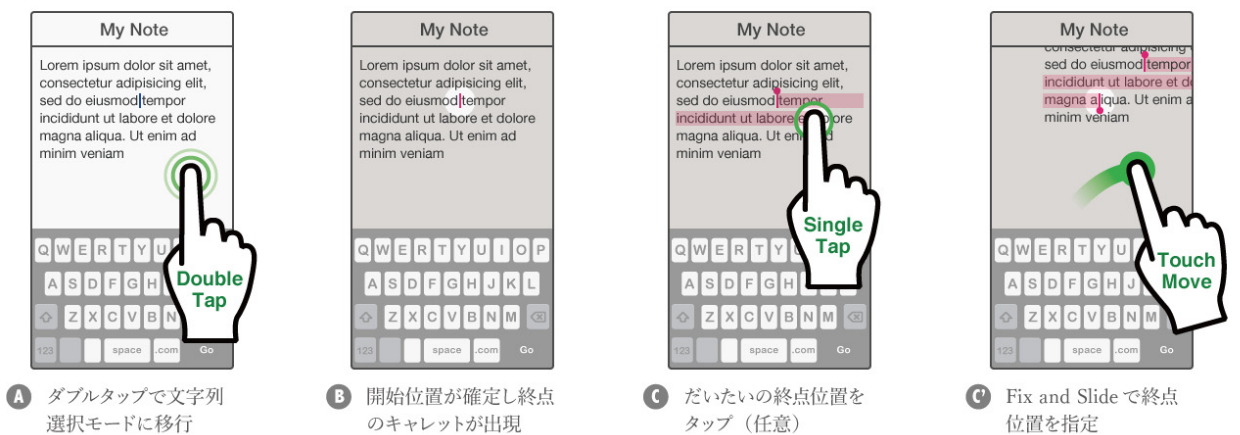


図 2 提案手法の概要. キャレット移動 (上段) と文字列選択 (下段). タッチを起点に, キャレットを固定 (Fix) し, 背景文字列を移動 (Slide) することで, キャレット位置を操作できる.

Fig. 2 Overview of Fix and Slide technique: Caret navigation (top) and text selection (bottom). A user first “fix” a caret in text by tapping on a display and then “slide” his/her finger to move background text for navigating caret.

のが, ユーザがスクリーン上をタップしても文字列中の目的の位置に1度で正確にキャレットを置けないことである. この場合, 多くは目的の位置のすぐ側に置かれてしまうため, 何度かタップするか少しずつドラッグするかのどちらか, もしくは両方の操作を行うことが求められることが多い. 目的の位置が指の下に来てしまうようなインターフェースの場合にはさらに操作が難しくなる. 前述した Shift [2] を適用した方法の場合には, 指の周囲の適切な場所に「ふきだし」が表示され, その中に指の下に位置する情報が提示されるが, 操作の難しさを解決することには直結しない. 我々はこれが直接操作手法の問題点であると考えている.

これに対して本稿で提案する Fix and Slide は間接操作手法である. まずユーザは直接操作手法と同様にスクリーンをタップしてキャレットを配置する (図 2 上段). その後, スクリーン上の任意の場所にタッチしてドラッグすることで, 文字列全体を背景として動かすことができる. こ

のとき, キャレットは最初にタップした時点での位置に固定され移動しない, すなわち背景のみが移動することになる. ユーザはキャレットを文字列中で移動させるのではなく, 移動する文字列 (背景) によってキャレット操作を行うことが本提案手法の特徴である.

3.2 文字列選択

ユーザが文字列選択を開始したい位置にキャレットを移動させた後, 文字列選択操作に移行する. 画面をダブルタップ (画面を素早く2度触る) することで, キャレットの色が変わり, 文字列選択モードに変更することができる (図 2 下段). 続いて, 文字列選択を終了したい位置にキャレットを移動させる. これはキャレット操作手法と同様で, 最初にスクリーンをタップし, そのままドラッグすることで文字列全体を背景として移動させ, 目的の位置へと導く. この操作はダブルタップを挟んでキャレット操作手

法を2度実行していることと同義である。

#### 4. 既存手法との比較実験

提案手法の有効性を検証するために、現在一般的に利用されている既存のインタフェース、具体的にはiOS上で実装されているインタフェースとの比較実験を実施した。ここでは文字列選択をタスクとして扱うことにした。すなわち、実験参加者はFix and SlideとiOSが標準で提供している文字列選択手法を使用して文字列選択を行うよう指示される。以降はiOS上で実装され利用可能な文字列選択手法を既存手法と呼び\*1、Fix and Slideを提案手法と呼ぶ。実験は、この両手法における文字列選択操作を行う際の作業実行時間（パフォーマンス）およびユーザビリティに関するフィードバックを得ることを目的とした。

##### 4.1 実験で使用する小型携帯端末

Fix and SlideはiOSバージョン8.2のiPhone6上で実装した。ディスプレイは解像度1,344×750で4.7インチだった。実験参加者へは自由な姿勢でスマートフォンを持つよう促し、片手で持つことも両手で持つことも指示していない。実験は既存手法と提案手法それぞれが実装されたiOSアプリケーションで行い、このアプリケーション内で作業実行時間を自動で計測し収集した。

##### 4.2 タスク：文字列選択

本研究で実施する実験では、文字列選択タスクを採用する。使用する言語は英語とする。英語は単語間がスペースで区切られていることが特徴である。一方で、日本語は単区切られておらず、句読点で文章の終わり、もしくは文章の切れ目を表現することが特徴である。今回の実験では「日本語の文字列選択は、英語における長い単語の途中を切り出すこと」と見なすことができることに注目し、英語を使用することとした。

実験で使用する文字列はランダムに生成される\*2。本実験では生成された文字列のうち、最初の2から4段落を使用する。ただし、後述するように特殊なケース、具体的にはURLについては手動で挿入する。実験において我々は5つの文章を用意した。

##### 4.3 実験条件

実験では2つの文字列選択手法（提案手法と既存手法）に加えて、以下の条件を設定する。

###### 4.3.1 フォントサイズ

文字列選択タスクの作業実行時間はフォントサイズに大

\*1 誌面の都合上、既存手法の文字列選択方法はこちらを参照されたい。端的にはiOS上には文字列選択手法は複数存在し、実験ではすべて利用可能であった。https://support.apple.com/kb/PH3577?locale=ja\_JP

\*2 http://randomtextgenerator.com/



図3 実験で使用するフォントサイズ3種：12ポイント（上段）、14ポイント（中段）、16ポイント（下段）。中段の図は文字列の部分選択条件の例を示し、下段は外側選択条件の例を示している。

Fig. 3 Three font sizes of 12 (top), 14 (middle), and 16 (bottom). Note that the middle image is “inside” word selection example and bottom is “outside” word example for supplementary information.

きく依存することが考えられる。このため、本実験では12ポイント、14ポイント、16ポイントの3つのフォントサイズを用意する（図3）。これらのフォントサイズはApple社のiOS Human Interface Guidelines [16]に従って選択した。すなわち：

“ユーザが「極小」を選択しても、11ポイント未満にしないでください。ちなみに「本文」スタイルでは、「大」のとき17ポイント（デフォルト値）で表示するようになっています”

とされており、12、14、16ポイントは妥当なフォントサイズの選択であると考えている。

###### 4.3.2 文字列選択の位置

フォントサイズと同様に、文字列中のどの部分を選択するのも重要な要素となることが考えられる。本実験では「文字列の部分選択」と「文字列の外側選択」の2条件を設定する。

###### 文字列の部分選択

現代的な小型携帯端末用のOSにおいては、文字列を選択する際にユーザの指の動き、すなわちキャレットの位置

表 1 文字列選択の位置  
Table 1 Position of text selection.

部分選択	Admitted add peculiar get 83,566 joy doubtful. http://www.Koqiksmkis.test/a.php?u=gaw&d=dplzk http://www.hksllaz.test/test.html?x=ddsa
外側選択	Maids hoped yet asked blind dried 2,000,000 point. Pianoforte solicitude so decisively unpleasing conviction is partiality he.

を認識し、文字列の最初（文字列間にスペースがある部分）もしくは最後（同様に文字列間にスペースがある部分）に賢く移動（スナッピング）する機能が実装されている。一方でそれは単語や文章単位で選択を行う場合のみ有効である。すなわち、文字列の部分選択を行う際にはあまり役に立たない。たとえば、英語の場合には数字の部分的な選択や、長い URL でホスト名だけを抜き出す場合には有効に機能しない。日本語の場合には基本的に部分選択となるため、いわゆる賢い選択の恩恵を受けることはあまりない。さらに悪いことに、賢い選択の場合には文字列の最初と最後に吸着（スナップ）する性質があり問題が生じることもある。前述した例と同様に、部分選択の例を表 1 に示す。

#### 文字列の外側選択

現代的な OS に搭載されている賢い機能（単語の最初と最後を認識し、キャレットがジャンプしたりする機能）は、一般的にユーザは文字列の最初から最後までを選択することを考えると、この機能によりユーザに素早く文字列選択することを可能としている。ここで、このような文字列選択を「外側選択」と呼ぶ。

実験において参加者は、表 1 の外側選択に示される文字の緑色の背景となっている部分を選択するように指示される。実際の表示では図 3 のように最初と最後の緑色の括弧が表示される。実験参加者が指示どおりの範囲を選択した場合にのみ、次の問題を開始するダイアログが表示される。

#### 4.4 実験参加者と手続き

30 名のボランティアが実験に参加した。男性は 19 名、女性は 11 名であり、年齢は 23 歳から 42 歳であった (mean = 30.6, S.D. = 4.9)。すべての実験参加者はスマートフォンを所有しており、1 名が Android 端末 (3.3%)、23 名が iPhone (76.7%) を、両方を所有しているのは 6 名であった (20.0%)。すなわち、1 名を除いてすべての参加者が iPhone を所有していた。また、すべての実験参加者はスマートフォン上での文字列選択を行ったことがあり、20 名については文字列選択に慣れていると回答した。どのような場面で文字列選択を行うかについては、URL をコピーする場合、インターネット検索をする場合、ツイートやブログに書く場合と回答した例が多かった。さらに、多くの参加者が文字列選択する際に、指で選択する領域が隠

れてしまうことに不満を持っており、同様に iOS の「賢い選択」によって不用意にキャレットがジャンプしてしまうことに不満を持っていた。

実験の手続きについて簡単に述べる。実験参加者は既存手法と提案手法の両方をそれぞれ 6 回ずつ使用する。実験では 3 つのフォントサイズと、2 つの文字列選択の位置の条件が存在する。文字列選択手法 2 種類とフォントサイズ 3 種類については、参加者が経験する順番のバランスをとった。一方で、文字列選択の位置 2 種類については学習効果がないため固定とし、最初に部分選択タスク、続いて外側選択タスクとした。参加者は 1 つの試行について、5 つの文字列選択タスクを行う。各タスクは実験用アプリケーション内で 1 タスクごとに時間が自動で計測される。以上を簡単にまとめると、1 つの手法について都合 30 回の文字列選択を実施し、実験参加者は合計で 60 回の文字列選択を行う。文字列選択を行う位置や範囲（文字数）はランダムではなく、統制をとるためにあらかじめ決められており、被験者間で同じ文字列を選択することとなる。それぞれの手法を利用する前には、操作に慣れるためのトレーニングの時間を設けた。このとき使われている文字列は本番のものとは別のものであった。

#### 4.5 既存手法と提案手法の違いの詳細

##### 既存手法

iOS には標準機能として下にあげる文字列選択手法がすでに実装されている [17]。

- 単語を 2 回タップすると単語全体が選択される。
- 単語を 4 回タップすると段落全体が選択される。
- 2 本指でタップすると文章が選択される。
- 選択したい文章の先頭を 2 回タップするとすぐ選択範囲伸縮が開始される。

上にあげた 4 つのタップを用いた選択手法に加え、我々はもう 1 つ手法を追加した。

- ロングタップで選択アイテムを持つポップアップメニューを表示。

本実験では最も慣れた選択手法と提案手法を比較するために、文字列選択のやり方について指示はせず、実験参加者へ iPhone をふだんどおり使うよう伝えた。

##### 提案手法

実験での提案手法は、iOS が標準で提供する文字列選択やキャレット移動等の各種機能は使えるが、文字列選択モードに変更するためにダブルタップによる選択機能を実装していない。ユーザはスクリーンをダブルタップし、モードを変更することで文字列選択を開始できる。

#### 4.6 実験結果

本実験においては、各タスクの作業実行時間とユーザビリティ評価に注目して分析を行う。

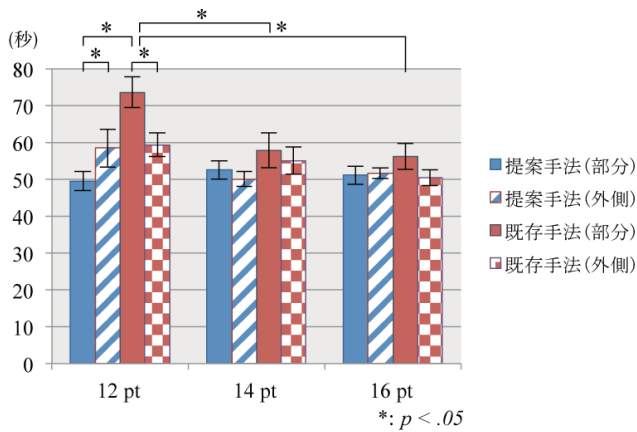


図 4 3 要因分散分析の結果と標準誤差

Fig. 4 Results of three-way ANOVA and standard error.

#### 4.6.1 作業実行時間

各条件の 1 試行 (5 タスク) における作業実行時間 [秒] の平均について、フォントサイズごとにまとめたものを図 4 に示す。既存手法および提案手法の操作手法ごとに、部分選択、外側選択と比較するように示している。これに対して 3 要因分散分析 (被験者内計画; 独立変数として、操作手法、フォントサイズ、および文字列選択の位置。従属変数として、作業実行時間) を実施した。この結果、有意な 2 次の交互作用 (操作手法 × フォントサイズ × 文字列選択の位置 [ $F(2, 58) = 8.60, p < .01$ ]) と、1 次の交互作用が確認された (操作手法 × 文字列選択の位置 [ $F(1, 29) = 16.07, p < .01$ ], および、操作手法 × フォントサイズ [ $F(2, 58) = 3.41, p < .05$ ])。また、単純主効果の操作手法 [ $F(1, 29) = 11.19, p < .01$ ] とフォントサイズ [ $F(2, 58) = 9.55, p < .01$ ] に有意差が確認された。文字列選択の位置については有意差が確認されなかったが、有意傾向が確認された [ $F(1, 29) = 3.13, p < .1$ ]。

##### 操作手法 × 文字列選択の位置

2 次の交互作用について、操作手法と文字列選択の位置に注目して 2 要因の分散分析を行った (被験者内計画)。この分析は 3 つのフォントサイズごとに実施した。この結果、12 ポイントの場合に交互作用が有意であった [ $F(1, 29) = 25.75, p < .01$ ] (図 4 の 12 ポイントの部分)。そこで、各要因の単純主効果を分析した結果、操作手法 × 文字列選択の位置 (部分選択) [ $F(1, 29) = 51.50, p < .01$ ], 文字列選択の位置 × 操作手法 (提案手法) [ $F(1, 29) = 6.68, p < .05$ ], 文字列選択の位置 × 操作手法 (既存手法) [ $F(1, 29) = 16.69, p < .01$ ] について有意な差が確認された。

12 ポイントの場合に、提案手法の部分選択と外側選択との間で有意差が確認された。実験時の様子から、これは英単語間にあるスペースの影響が考えられる。12 ポイントのスペースは幅が狭く、スペースを含んでいることに気づかないまま選択をしてしまう、もしくは回避するために微調整が必要とされるシーンが多く見られた。これが原因とし

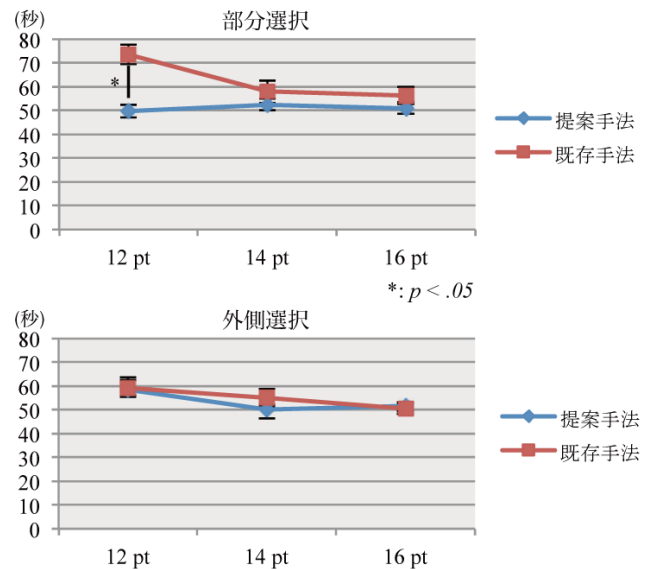


図 5 操作手法 × 文字列選択の位置の結果と標準誤差

Fig. 5 Result of Method × Font Size and standard error.

て作業実行時間に有意な差が生じた可能性がある。

##### 操作手法 × フォントサイズ

図 5 に操作手法 × フォントサイズについて、文字列選択の位置に注目した結果を示す。部分的な文字列選択の場合、提案手法はフォントサイズに依存せず、作業実行時間は一定だが、既存手法の場合はフォントサイズが大きくなるにつれて作業実行時間が短くなる傾向が確認された。ここで部分選択に注目して 2 要因分散分析 (被験者内計画) を実施した結果、交互作用が確認された [ $F(2, 58) = 8.89, p < .01$ ]。Bonferroni 法を用いた多重比較を実施した結果、既存手法の場合、フォントサイズが 14, 16 ポイントの場合に比べて 12 ポイントの場合に有意に作業時間が長いことが確認された ( $MSe = 206.26, p < .05$ )。また、外側選択に注目して 2 要因分散分析を実施した結果、主効果 (フォントサイズ) が有意であった [ $F(2, 58) = 5.82, p < .01$ ]。ここで Bonferroni 法を用いた多重比較を実施した結果、12 ポイントの場合が有意に長い結果であった ( $MSe = 179.54, p < .05$ )。

#### 4.6.2 ユーザビリティ評価

続いて、各操作手法を使用した後に実験参加者に回答を依頼したアンケートの分析を行う。ここでは System Usability Scale (SUS) [18] をもとに、本実験用に微修正したアンケートを実施した\*3。SUS は 5 段階のリッカート尺度の質問 10 項目で構成される、ユーザビリティ評価のためのアンケートであり、最終的に 100 点満点でスコアが得られるものである。この結果、提案手法は 68.67 点、既存手法は 49.5 点と評価され、これらについて分散分析を実施した結果、提案手法が有意に高い結果であることが確認さ

\*3 誌面の都合上、それぞれのアンケート項目は文献 [18] を参照されたい。なお、SUS を日本語にする際に質問が曖昧な表現になる部分を補う修正をした。

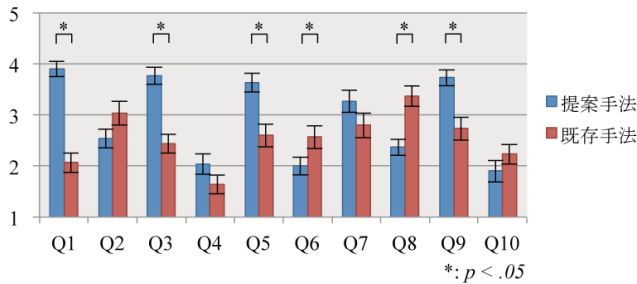


図 6 SUS の結果と標準誤差

Fig. 6 Results of SUS questionnaire and standard error.

れた [ $F(1, 29) = 19.62, p < .01$ ]. それぞれの項目についても分散分析を行った結果, Q1: “頻繁に使用したいと思った” [ $F(1, 29) = 50.27, p < .01$ ], Q3: “使いやすいと思った” [ $F(1, 29) = 28.29, p < .01$ ], Q9: “自信を持って使うことができた” [ $F(1, 29) = 11.15, p < .01$ ] で有意な差が確認された (図 6). また, Q5: “この操作方法がスマートフォンの操作方法としてよく統合されていると思った” という項目でも有意な差が確認されたこともあわせて, 提案手法は既存手法と比較して使いやすく, またユーザはスマートフォンのインタフェースとして提案手法を統合していくことを歓迎していることが示された.

#### 4.6.3 実験中の参加者の観察結果

既存手法において実験参加者の多くは, まず指示された範囲の始点もしくは終点近辺をロングタップし, タッチ領域を拡大表示させた状態で指をスライドし, キャレットを目的の位置にゆっくり導いた. 続いて指を離れた際に表示されるポップアップメニューから選択機能を実行し, 選択された範囲の前後にあるグラフポイントをドラッグしタスクを完了させることが多かった.

提案手法においては実験参加者の多くは, まず文字列全体を大きく動かし指示された範囲の始点もしくは終点付近にキャレットを近づけ, 至近距離となるとゆっくり操作することが多かった. 続いてダブルタップし選択モードに移行後, 同様の操作を行いもう一方の指示点に合わせていた. 回数を経るに従って 1 回の操作で指示された位置に直接合わせられるようになった参加者も少なくなかった.

### 5. 初学者の利用の観察実験

本実験では小型携帯端末の操作に不慣れである児童を対象とした観察実験を実施した. 対象は Fat Finger Problem を強く経験したことがない小型携帯端末の未習熟者であり, タスクとして提案手法を自由に操作してもらい, 提案手法がどう使われるか観察し, 有用性を検討することを目的としている. このため, 文字列操作をまったく行っていないユーザの質的な観察が主目的であり, 量的な作業実行時間は計測しなかった. 本観察実験は前述の比較実験と同じ機器およびプロトタイプシステムを用い, タスクと条件を簡略化した内容で実施した.

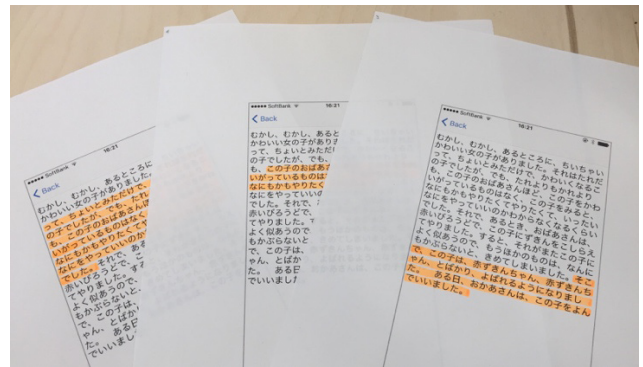


図 7 文字列選択範囲を指示する紙面の例

Fig. 7 Examples of printed instruction for text selection.

#### 5.1 タスク：文字列実験

観察実験で実施する実験では, 文字列選択タスクを採用し, 使用する言語は日本語とした. 今回の実験では「小型携帯端末の未習熟者が提案手法をどのように扱うかの観察」が目的であるため, 実験参加者が読みやすい文章として, 小学校低学年向けに分類される童話とした\*4.

#### 5.2 条件

観察実験では 2 つの文字列選択タスク (提案手法と既存手法) に加えて, 以下の条件を設定した.

##### 5.2.1 フォントサイズ

観察実験は選択操作の観察が目的であるため, iOS Human Interface Guidelines [16] における「大」サイズ相当の 18 ポイントとした. これはタスク実行時に文字の読みにくさによるタスクの難易度上昇を招かない妥当なサイズと考えている.

##### 5.2.2 文字列選択の位置

フォントサイズと同様に意図せぬタスクの難易度上昇を防ぐため, 文字列選択の対象は文章単位とした. すなわち, 選択してもらう範囲は 1 文の文頭から文末までとなる. 文字列の選択タスクは, 画面上に表示されている文章と同じものが印刷された B5 サイズの紙に蛍光ペンでマークし指定した (図 7). 実験参加者はこれを見ながら指定された範囲を選択する.

#### 5.3 実験参加者と手続き

3 名のボランティアが実験に参加した. 男子児童 3 名で, 年齢は 7, 9, 11 歳であった. 11 歳の児童は 1 名で, 7, 9 歳の児童は 2 名で実験を行った.

すべての実験参加者はスマートフォンを所持していなかった. 7 歳と 11 歳の男児は頻繁に, 9 歳の男児はときどき保護者が所有するスマートフォンを用いゲームを行った経験があった. すべての実験参加者はスマートフォン上で文字列のポインティングと選択操作をほぼ行ったことが

\*4 青空文庫, グリム兄弟, 楠山正雄訳, 赤ずきんちゃん, [http://www.aozora.gr.jp/cards/001091/files/42311\\_15546.html](http://www.aozora.gr.jp/cards/001091/files/42311_15546.html)

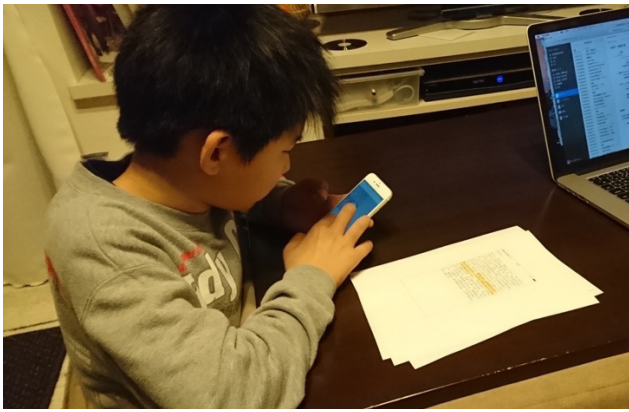


図 8 実験参加者と実験の様子  
Fig. 8 Experimental situation.

なかった。ゲームや動画等のコンテンツを検索する際は、サービス上に初期表示されるレコメンドからたどる手法をとっており、文字列操作に関する Fat Finger Problem を強く体験したことはほぼなかった。

実験の手続きについて簡単に述べる。実験参加者は2つの文字列選択手法（提案手法と既存手法）のやり方をそれぞれステップバイステップで教わる。続いて、実験参加者に選択範囲が示された紙を渡し、そこにマークされた箇所をプロトタイプシステム上で選択してもらった（図 8）。実験参加者は各手法をそれぞれ 15 分程の時間で学習とタスクの実施を行った。

#### 5.4 観察結果

11 歳の実験参加者は2つの文字列選択手法（提案手法と既存手法）のどちらの操作も最初はエラーを繰り返したが、慣れるにつれ正確かつ迅速に操作するようになった。7, 9 歳の実験参加者は全体的にどちらの操作も時間がかかったが、慣れるにつれキャレットのポインティングを正確に操作するようになった。

頻繁に確認されたエラーとして、既存手法では選択範囲の終点を移動する際に、誤って始点を触ってしまい移動不能となる Fat Finger Problem に起因されたと思われるケースがあった。これは3名中3名で発生し、学習時間に関係なく頻発していた。提案手法では学習初期に、選択の始点指定をしないまま選択モードに変更してしまい、やり直しとなるケースが多かったが、これは試行回数が増えるに従って発生しなくなった。7 歳の実験参加者は、どちらの手法でもスマートフォンを固定する手や、操作しようとしている指以外の部分が画面に当たることに起因する誤動作を頻発した。9 歳の実験参加者もときどき同様の誤動作を起こした。

未習熟者にとってどちらの手法も初見では操作が困難であったが、短い学習時間を経ることでタスクを完了させられるようになった。また既存手法で起きた Fat Finger

Problem に起因したと思われるエラーは提案手法では発生しなかった。このように、提案手法は未習熟者でも短い学習時間で操作の習得ができ、また既存手法でよく見られた Fat Finger Problem によるエラーを回避する有用性も一部見られた。

実験後のインタビューで、キャレットのポインティングに関しては既存手法の方が直接触るから早くてやりやすいという意見が出た。これは Fat Finger Problem の原因にもなりうるが、提案手法は既存手法を排他しないため、既存手法も同時に機能させることでユーザビリティを向上させられる可能性がある。

## 6. 議論

本研究で実施した実験の結果は以下のようにまとめられる。

- 小型携帯端末を日常的に利用するユーザは、提案手法を初めて使用するにもかかわらず、作業時間は標準的な操作手法とは有意な差が確認されなかった。ただし、フォントサイズが小さい場合には提案手法の方が有意に短い作業時間となる例もあった。
- 本実験の結果からも、既存手法の小フォントサイズ条件において作業速度が遅くなる Fat Finger Problem が観察されたが、提案手法では観察されなかった。
- SUS の結果から、ユーザビリティの観点からは提案手法は標準的な操作手法よりも有意に使いやすいインタフェースであると評価された。
- 小型携帯端末未習熟の児童の観察実験の結果から、どちらの操作手法についてもタッチ操作の問題点は観察された。一方で提案手法においてはタッチ操作に慣れているユーザと同じように利用できている様子が観察された。

上記の結果より、既存手法と比較した際の提案手法の有用性、有効性を示すことができたと考えている。一方で提案手法についての制限や将来課題は残っている。

まず、本研究で実施した実験は提案手法を iOS で実装されている文字列選択手法と比較しただけであり、他の OS、たとえば Android 上で実装されている手法とは比較していない。今回の実験は概念実証には十分な実験であったと考えているが、他の環境で提供される手法との比較は必要であると考えている。次に、実験においては長い文章を選択する場合を想定しておらず、この場合の実験についても必要であると考えている。インタフェースの実装としては、たとえば1本指でのドラッグ操作で選択、2本指でのドラッグ操作で画面スクロールを行い、長い文章の選択することは可能であるが、実用化に向けては事前検証が必要であると考えている。

最後に、学習に関するコストについて触れる。本研究で提案したインタフェースはこれまでの概念とはまったく異



なり、キャラットの移動および文字列の選択を文字列全体を背景として移動することで操作するというものであった。このため、学習コストが大きいのではないかという批判が考えられるが、比較実験においては1名を除いた全員がiPhoneを所有し日常的に使用しているユーザが参加し、これら実験参加者が初めて触るFix and Slideの操作速度の方が統計的に有意な差で速く、そのほかについては統計的な差が確認されていない程度の速度であったことを考えると、学習コストは高くなく、ユーザビリティ評価の結果とあわせて考えると、十分に受け入れられやすいものであると考えられる(図4, 図6)。また、観察実験における小型携帯端末未習熟者の利用観察の結果からも、提案手法は既存手法と同様の学習コスト程度であることが示唆された。

## 7. むすび

本稿では、小型携帯端末における文字列選択操作について、従来の直接対象を触って操作する直接操作手法ではなく、文字列全体を背景として移動することによるキャラット操作および文字列選択手法の提案を行った。現在標準的に利用されているインタフェースとの比較実験を行った結果、ユーザは提案手法を初めて使うにもかかわらず、フォントサイズが小さい場合は提案手法の方が有意に短い作業時間となり、またその他の場合では標準のインタフェースとは有意な差が確認されなかった。さらに、ユーザビリティの観点では提案手法の方が高い評価を得た。

## 参考文献

- [1] 総務省：平成26年通信利用動向調査，入手先 (<http://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h27/html/nc372110.html>) (2015).
- [2] Vogel, D. and Baudisch, P.: Shift: A technique for operating pen-based interfaces using touch, *Proc. SIGCHI Conference on Human Factors in Computing Systems (CHI '07)*, pp.657-666, ACM (2007).
- [3] Suzuki, K., Okabe, K., Sakamoto, R. and Sakamoto, D.: Fix and slide: caret navigation with movable background, *Proc. 18th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '16)*, pp.478-482, ACM (2016).
- [4] Roudaut, A. Huot, S. and Lecolinet, E.: TapTap and MagStick: Improving One-handed Target Acquisition on Small Touch-screens, *Proc. Working Conference on Advanced Visual Interfaces (AVI '08)*, pp.146-153, ACM (2008).
- [5] Sears, A. and Shneiderman, B.: High Precision Touchscreens: Design Strategies and Comparisons with a Mouse, *Int. J. Man-Mach. Stud.*, Vol.34, No.4, pp.593-613, Elsevier (1991).
- [6] Wigdor, D. Forlines, C. Baudisch, P. Barnwell, J. and Shen, C.: Lucid Touch: A See-through Mobile Device, *Proc. 20th Annual ACM Symposium on User Interface Software and Technology (UIST '07)*, pp.269-278, ACM (2007).
- [7] Hasan, K., Yang, X.-D. Liang, H.-N. and Irani, P.: How to position the cursor?: An exploration of absolute and relative cursor positioning for back-of-device input, *Proc. 14th International Conference on Human-computer Interaction with Mobile Devices and Services (MobileHCI '12)*, pp.103-112, ACM (2012).
- [8] Holz, C. and Baudisch, P.: The Generalized Perceived Input Point Model and How to Double Touch Accuracy by Extracting Fingerprints, *Proc. SIGCHI Conference on Human Factors in Computing Systems (CHI '10)*, pp.581-590, ACM (2010).
- [9] Bi, X. and Zhai, S.: Bayesian Touch: A Statistical Criterion of Target Selection with Finger Touch, *Proc. 26th Annual ACM Symposium on User Interface Software and Technology (UIST '13)*, pp.51-60, ACM (2013).
- [10] Benko, H. Wilson, D.A. and Baudisch, P.: Precise Selection Techniques for Multi-touch Screens, *Proc. SIGCHI Conference on Human Factors in Computing Systems (CHI '06)*, pp.1263-1272, ACM (2006).
- [11] Wolf, K. and Henze, N.: Comparing pointing techniques for grasping hands on tablets, *Proc. 16th International Conference on Human-computer Interaction with Mobile Devices & Services (MobileHCI '14)*, pp.53-62, ACM (2014).
- [12] Eady, K.A. and Girouard, A.: Caret Manipulation Using Deformable Input in Mobile Devices, *Proc. 9th International Conference on Tangible, Embedded, and Embodied Interaction (TEI '15)*, pp.587-591, ACM (2015).
- [13] 三浦元喜, 最所賢至, 清弘祥太: タブレット端末における日本語形態素を考慮した文字列範囲選択手法, インタラクティブシステムとソフトウェアに関するワークショップ (WISS '14), pp.121-122, ソフトウェア学会 (2014).
- [14] Scheibel, J.-B., Pierson, C., Martin, B., Godard, N., Fuccella, V. and Isokoski, P.: Virtual Stick in Caret Positioning on Touch Screens, *Proc. 25th Conference on l'Interaction Homme-Machine (IHM '13)*, pp.107-114, ACM (2013).
- [15] Fuccella, V., Isokoski, P. and Martin, B.: Gestures and widgets: Performance in text editing on multi-touch capable mobile devices, *Proc. SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*, pp.2785-2794, ACM (2013).
- [16] Apple. iOS Human Interface Guidelines : 色とタイポグラフィ, 入手先 (<https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/MobileHIG/ColorImagesText.html>).
- [17] Lifehacker, Five Shortcuts for Faster Text Selection in iOS (2015), available from (<http://lifehacker.com/five-shortcuts-for-faster-text-selection-in-ios-1641490708>).
- [18] Brooke, J.: SUS-A quick and dirty usability scale, Usability Evaluation in Industry, London: Taylor and Francis, pp.189-194 (1996).



鈴木 健司

2006年多摩美術大学大学院美術研究科修士課程修了。同年ヤフー株式会社，現在に至る。2018年北海道大学大学院情報科学研究科博士課程進学。



岡部 和昌

2006年ヤフー株式会社に入社，現在に至る．同社にてWeb標準関連やフロントエンド開発業務に従事．



坂本 竜基 (正会員)

2003年北陸先端科学技術大学院大学知識科学研究科博士後期課程修了．同年ATR知能ロボティクス研究所研究員，2008年和歌山大学システム工学部講師，2012年Yahoo!JAPAN研究所上席研究員，先端技術応用室室長を経て，2017年より電駆ビジョン株式会社CEO．2004年度IPA未踏ソフトウェア天才プログラマー/スーパークリエータ，2014年度山下研究記念賞等を受賞．博士（知識科学）．



坂本 大介 (正会員)

2008年公立ほこだて未来大学大学院システム情報科学研究科博士（後期）課程修了．博士（システム情報科学）．国際電気通信基礎技術研究所（ATR）でインターン，東京大学で日本学術振興会特別研究員PD，JST ERATO五十嵐デザインインタフェースプロジェクト研究員，東京大学大学院情報理工学系研究科コンピュータ科学専攻助教，特任講師を経て，現在，北海道大学大学院情報科学研究科准教授．人とロボットを含む情報環境とのインタラクション設計に関する研究に従事．ACM会員．