

Fix and Slide: 指による遮蔽を回避する 相対的なポインティング手法

鈴木健司^{†1} 岡部和昌^{†1} 坂本竜基^{†1} 坂本大介^{†2}

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おこなう際、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できなくなる問題、いわゆる Fat Finger Problem に対して、ポインタそのものの位置を指示するのではなく、ポインタが存在する画面を操作してポインタの位置指定を相対的にこなう手法について述べる。この手法は Fat Finger Problem が起こる多くの場面に有用であると考えられるが、本研究では特にスマートフォンにおける文字列選択がしにくい課題へのアプローチを目標とした。評価は、iOS における文章の一部文字列を本手法、および OS が標準で提供している手法それぞれで選択できるプロトタイプを作成したうえで、被験者実験をおこなった。その結果、提案手法をはじめて操作したユーザであっても従来手法と変わらない速度で作業が完了できた、もしくは早い場合があったことが確認され、また通常的手法よりも理解し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、可動式テキストフィールド、文字選択、モバイル端末

Fix and Slide: Caret Navigation with Movable Background

KENJI SUZUKI^{†1} KAZUMASA OKABE^{†1} RYUUKI SAKAMOTO^{†1}
DAISUKE SAKAMOTO^{†2}

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 not different or 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, movable background, text selection, mobile device

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のアプリへとシフトしてきた[1]。それらの端末が受信するメディアは旧来の文字や画像はいうに及ばず、よりリッチな映像にまで及び始めているが、ニュースや Web メールなど文字を主体とするサービスでの利用もいまだ根強い。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も大きく、いざ正確なポインティングをしようする際には「指が操作対象を隠してしまう」いわゆる“Fat Finger Problem”が発生するからである[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 OS において文字列選択をする場合、ユーザはまずキャ

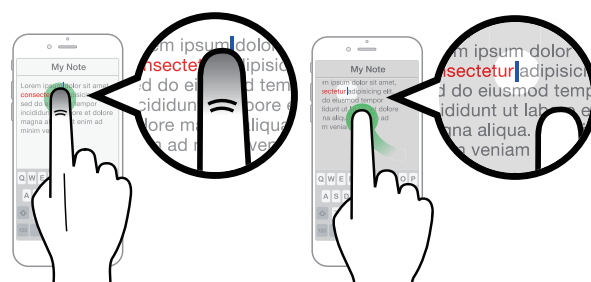
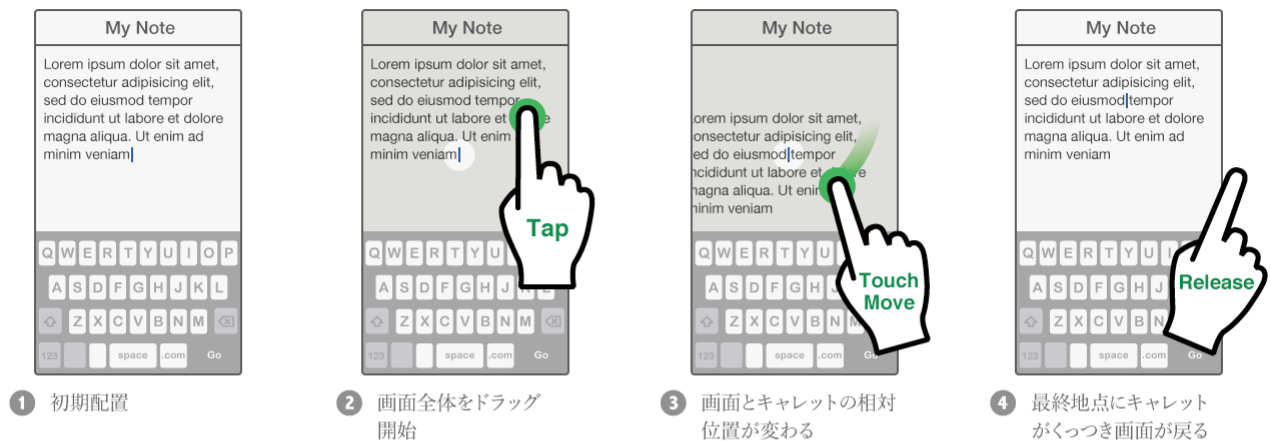


図 1 一般的なキャレット操作手法(左)と提案手法(右): ユーザは背景となる文字列をドラッグすることで任意の場所にキャレットを移動することができる。

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、長押し等で文字選択モードにしたのち、文字選択の終了地点にキャレットを移動させる(図 1 左)。しかし、Fat Finger Problem が一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、文字選択の終了地点までキャレットを移動させる場合も一度で正確な操作が難しく、何度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

^{†1} ヤフー株式会社
Yahoo Japan Corporation
^{†2} 東京大学
The University of Tokyo

キャラクタ移動



文字列選択

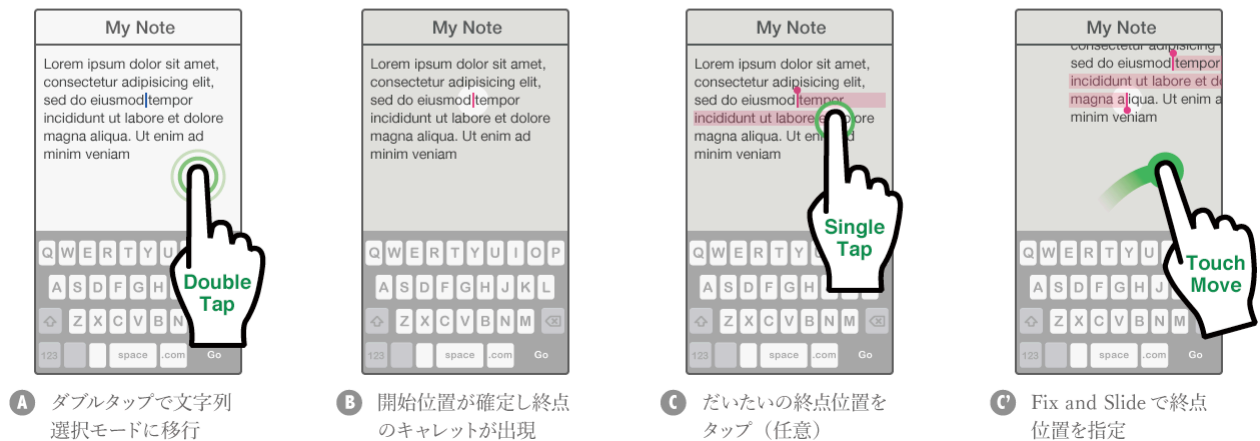


図 2 提案手法の概要. キャラクタ移動 (上段) と文字列選択 (下段)

はじめてキャラクタを固定(Fix)し、その後背景文字列を移動(Slide)することで文字列選択ができる。

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、直接対象を触って操作する直接操作手法ではなく、その背景となる文字列全体を操作することによるキャラクタ操作および文字選択手法をおこなう Fix and Slide とよぶポインティング手法を提案する (図 1 右)。Fix and Slide では、キャラクタは画面上の点 p で固定されており、基本的にそれを直接操作することはない (p が対象からあまりにも遠い場合はタップにより再配置できる)。その代わりに、画面のどこかをドラッグしても文字列全体を移動させることができ、ドラッグ操作が終わり指を画面から離れた時、p に位置する文字と文字の間にキャラクタがくつき、文字列は元の位置に戻る挙動をする。このドラッグをしている最中は、キャラクタが指で隠れないため Fat Finger Problem はおこらず、適切な位置にキャラクタ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おこなった。実験のうち、量的な比較検討は日常的に iOS を使用している成人に対しておこない、質的な観察は小型携帯端末上での文字へのポインティングをおこなったことがない年少者に対して実施した。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領域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実施されてきている。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によって隠されてしまう領域が「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[3].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遅かった[4].これまで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[5]. Hasan らのモバイル端末の裏面を用いた研究を行っている[6].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、キャラ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概観する.これまでに高精度にタッチ操作をセンシングする技術[7]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[8], 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[9], 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[10]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ラット操作に関する手法について提案[11]があったが、端末の周囲に曲げセンサを取り付ける必要があるなど、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[12]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.またモバイル端末上での仮想スティック[13]やジェスチャーを用いた文字列選択手法[14]も提案されている.これらはモバイル端末でのキャラット操作や文字列選択の手法であるが、我々の手法とは明らかに異なる.

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ラットの移動や文字列選択方法として文字列を含む背景を操作するということである(図2).以下でそれぞれの操作について説明する.

3.1 キャラット操作

小型携帯端末上でキャラット操作を行う際に問題となるのが、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正確にキャラットを一度のタップで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ってしまうようなインタ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.ユーザはまず直接操作手法と同様にスクリーンをタップしてキャラットを配置する(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、背景を動かすことができる.このとき、キャラットは最初にタップされた場所に固定されているため、キャラットは固定されたまま、背景が移動することになる.ユーザはキャラットを文字列中で移動させるのではなく、文字列(背景)を移動させることによってキャラ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

ユーザが最初にキャラット操作を行った後、文字列選択操作に移行する.画面をダブルタップ(2度画面を素早く触る)することで、文字列選択モードに変更することができる(図2下段).これ以降はキャラット操作手法と同様で、最初にスクリーンをタップし、スクリーンを動かすことで背景および文字列全体を移動させる.これにより、文字列選択の点を定めることができる.この操作はダブルタップを挟んでキャラット操作手法を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 上での文字列選択手法を既存手法と呼び^a、Fix and Slide のことを提案手法とよぶ.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 1,334×750 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.

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、日本語は単語間がスペースで区切られておらず、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

a 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、実験では全て利用可能であった. https://support.apple.com/kb/PH3577?locale=ja_JP

い単語の途中を切り出すこと」と見做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^b。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[15]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”

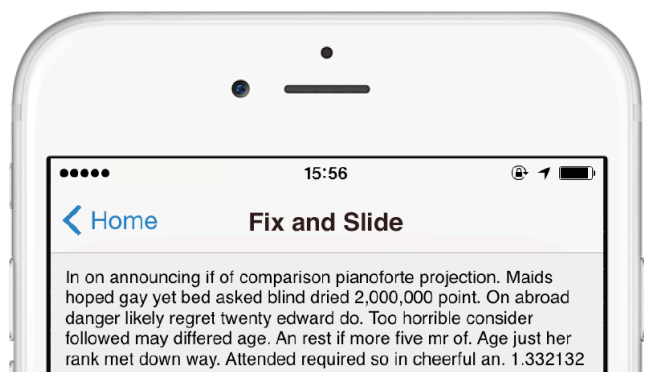
とされており、12、14、16ポイントは妥当なフォントサイズの選択であると考えている。

4.3.2 文字列選択の位置

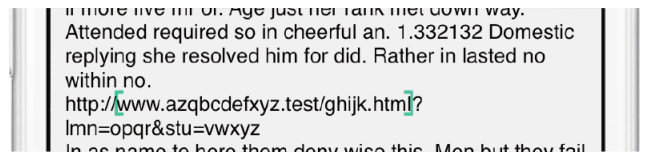
フォントサイズと同様に、文字列中のどの部分を選択するのも重要な要素となることが考えられる。本実験では「文字列の部分選択」と「文字列の外側選択」の2条件を設定する。

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用である。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。例えば、英語の場合には数字を部分的に選択したり、長いURLの場合、ホスト名だけを抜き出したい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スナップ（吸着）する性質があるため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。

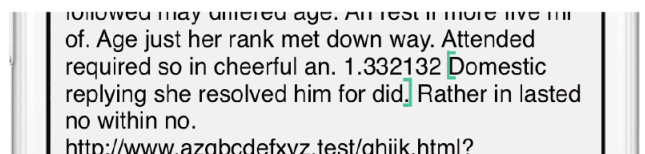
文字列の外側選択 現代的なOSに搭載されている賢い機能（単語の最初と最後を認識し、キャレットがジャンプしたりする機能）は、一般的にユーザは文字列の最初から



12pt（文字列の一般例）



14pt×部分選択の例



16pt×外側選択の例

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。

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に緑色の括弧が表示される。これらの文字列はランダムに生成されたものである。

表1 文字列選択の位置の例

部分選択	Admitted add peculiar get 83 566 joy doubtful. http://www.Koqiksmkis.test/a.php?i=gaod=dplzk http://www.hksllaz.test/test.html?x=ddsa
外側選択	Maids hoped gay yet bed asked blind dried 2,000,000 point. Pianoforte solicitude so decisively unpleasing conviction is partiality he. Or particular 278,061 so diminution entrea-ties oh do.

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）。全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名を除いて全ての参加者がiPhone

^b <http://randomtextgenerator.com/>

を所有していた。また、全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取った。一方で、文字列選択位置2種類については学習効果がないため、固定とした。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。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はランダムではなく、統制を取るために予め決められており、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。この時使われ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標準のユーザインタフェースは下に挙げる文字列選択方法を持つ[16].

- (1) 単語を2回タップすると単語全体が選択される
- (2) 単語を4回タップすると段落全体が選択される
- (3) 二本指でタップすると文章が選択される
- (4) 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

上に挙げた4つの手法に加え、我々はもう1つ手法を追加した。

- (5) 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

文字列選択のやり方について指示せず、実験参加者へiPhoneを普段通りに使うよう伝えた。

提案手法

実験での提案手法は文字列選択モードに変更するためにダブルタップを除き、既存手法に搭載されている幾つかの選択機能をサポートしない。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。

4.6.1 作業実行時間

作業実行時間[秒]の平均について、フォントサイズ毎にまとめたものを(図4)に示す。既存手法及び提案手法の操作手法毎に、部分選択、外部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、およびキャレットの位置。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×キャレットの位置[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×キャレットの位置[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。キャレット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]。

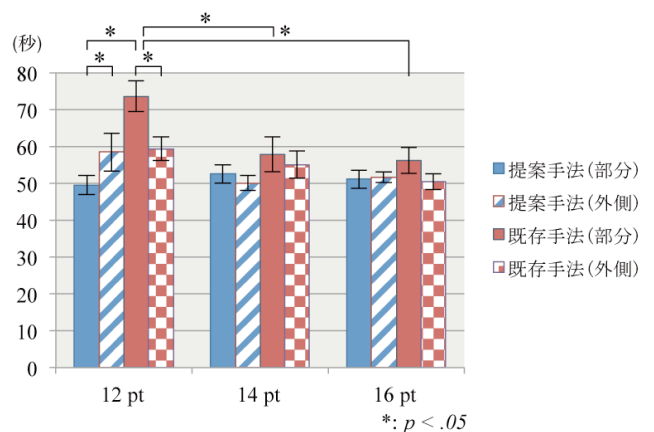


図4 要因分散分析の結果と標準誤差

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毎に実施した。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および図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(Fix and Slide)[F(1,29)=6.68, p<.05]、文字列選択の位置×操作手法(Default)[F(1,29)=16.69, p<.01]について有意な差が確認された。

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、Default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（ $MSe=206.2621, p<.05$ ）。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遅い結果であった（ $MSe=179.5455, p<.05$ ）。

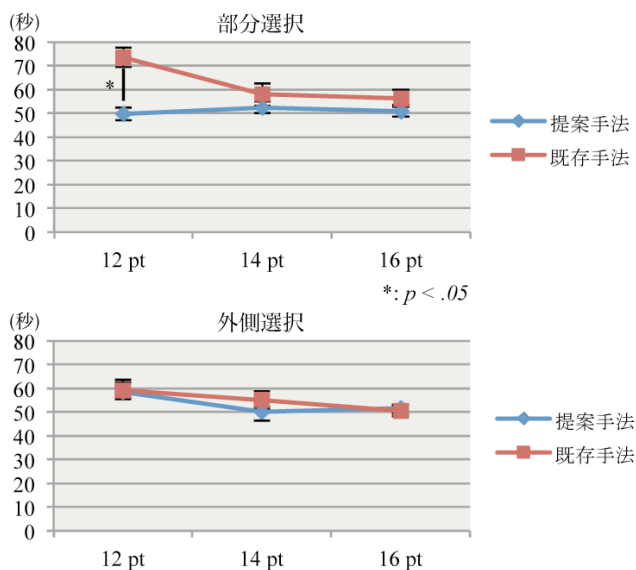


図 5 操作手法×文字列選択の位置の結果と標準誤差

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) [17]をもとに、本実験用に微修正したアンケートを実施した^c。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: “頻繁に使いたいと思った” $[F(1,29)=50.27, p<.01]$ 、Q3: “使いやすいと思った” $[F(1,29)=28.29, p<.01]$ 、Q9: “自身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併せて、提案手法は既存手法と比較し

^c 誌面の都合上、それぞれのアンケート項目は[17]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。

て使いやすく、またユーザはスマートフォンのインター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。

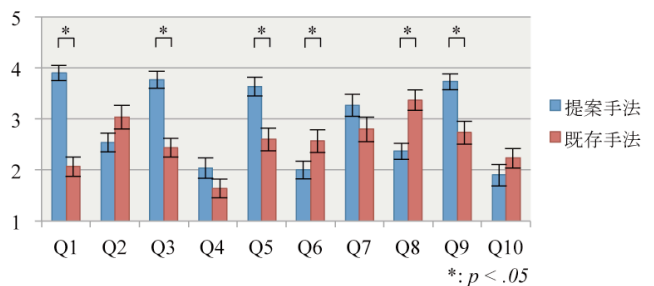


図 6 SUSの結果と標準誤差

5. 初学者の利用の観察実験

本実験では小型携帯端末における操作に不慣れである、児童を対象とした観察実験を行う。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、文字列操作を全く行ったことが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ていない。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を用い、タスクと条件を簡略化した内容で実施する。

5.1 タスク: 文字列選択

観察実験で実施する実験では、文字列選択タスクを採用する。使用する言語は日本語とする。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^dとした。

5.2 条件

観察実験では2つの文字列ポインティング・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。

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[15]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づらさによるタスク難易度の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難易度上昇を防ぐため、文字列選択の対象は文単位とした。すなわち、一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

^d 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

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箇所を選択する。

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人で、7, 9歳の児童は2人で実験を行った。

全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歳の男児は時々保護者のスマートフォンを用いゲームを行った経験があった。全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なかった。検索行為はサービス上に初期表示されるレコメンドから辿る手法を取っており、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つの文字列ポインティング・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位置を指示した紙を渡し、そこにマークされた箇所をプロトタイプ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分程の時間内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つの文字列ポインティング・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、慣れるにつれスムーズに操作した。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、慣れるにつれポインティングはスムーズに操作した。

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おり移動不能となる、いわゆる Fat Finger Problem と思われるケースがあった。これは3人中3人で発生し、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すことが多かった。これは回数を経るに従って発生しなくなった。

また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ている指以外の部分が画面に当たることにより起因する誤動作を頻発した。9歳の実験参加者も時々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も操作の習得ができ、また Fat Finger Problem を回避する有用性も一部見られた。

実験後フィードバックにて、ポインティングに関しては既存手法のほう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 の原因にもなり得るが、提案手法は既存手法を排他しないため、既存手法と融合す

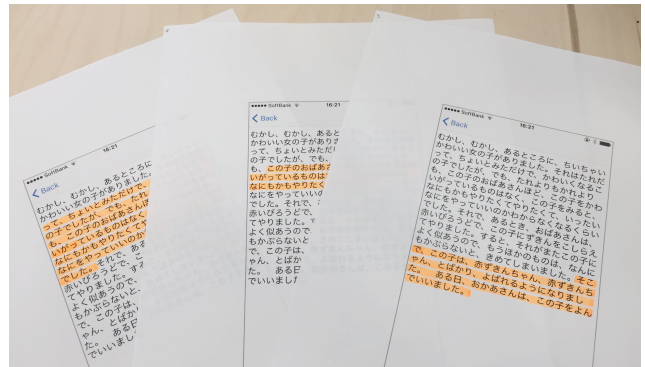


図7 文字列選択指示シートの例

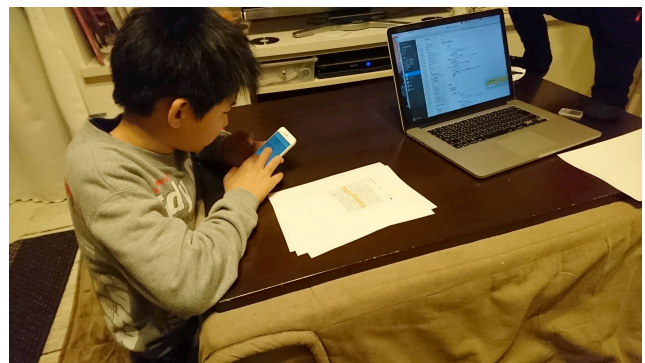


図8 実験参加者と実験の様子

ることでユーザビリティをより向上させられる可能性が示唆された。

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はじ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変わらない結果であった。ただし、フォントサイズが小さい場合には提案手法のほうが有意に作業時間が短い例もあった。
- 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 が観察されていたが、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。
- タッチ操作初心者の児童の利用観察の結果から、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。

上記の結果は概念実証 (Proof of concept) には十分な結

果であると考えられるが、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、例えば 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、一方で他の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面を想定しておらず、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、例えば一本指のドラッグ操作で選択、二本指のドラッグ操作でスクロール操作を行い、長い文章の選択を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全く異なり、キャレット操作及び文字列選択に背景を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批判が考えられるが、しかし比較実験においては 1 名を除いて全員が iPhone を所有し日常的に使用しているユーザが参加しており、この条件においてもはじめて触る Fix and Slide の操作速度のほうが統計的に有意な差で早く、その他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合わせて考えると、十分に受け入れられやすいものであると考えられる。観察実験における初心者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、その背景となる文字列全体を操作することによるキャレ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はじめて使うにも関わらず、フォントサイズが小さい場合は提案手法のほうが有意に作業時間が短く、またその他の場合では標準のインタフェースとほぼ同等のパフォーマンスであった。さらに、ユーザビリティの観点では提案手法のほうが高い評価を得た。

参考文献

- [1] 総務省, 平成 26 年通信利用動向調査,
<http://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h27/html/nc372110.html>, 2015.
- [2] D.Vogel and P.Baudisch. Shift: A Technique for Operating Pen-based Interfaces Using Touch. In Proc. CHI 2007, pp. 657–666. 2007.
- [3] A. Roudaut, S. Huot, and E. Lecolinet. TapTap and MagStick: Improving One-handed Target Acquisition on Small Touch-screens.

- In Proc. AVI 2008, pp. 146–153. 2008.
- [4] A. Sears and B. Shneiderman. High Precision Touchscreens: Design Strategies and Comparisons with a Mouse. Int. J. Man-Mach. Stud., Volume 34 Issue 4. pp. 593–613, 1991.
 - [5] D. Wigdor, C. Forlines, P. Baudisch, J. Barnwell, and C. Shen. Lucid Touch: A See-through Mobile Device. In Proc. UIST 2007, pp. 269–278. 2007.
 - [6] Khalad Hasan, Xing-Dong Yang, Hai-Ning Liang, and Pourang Irani. 2012. How to position the cursor?: an exploration of absolute and relative cursor positioning for back-of-device input. In Proc. MobileHCI 2012, pp. 103–112. 2012.
 - [7] C. Holz and P. Baudisch. The Generalized Perceived Input Point Model and How to Double Touch Accuracy by Extracting Fingerprints. In Proc. CHI 2010, pp. 581–590. 2010.
 - [8] X. Bi and S. Zhai. Bayesian Touch: A Statistical Criterion of Target Selection with Finger Touch. In Proc. UIST 2013, pp. 51–60. 2013.
 - [9] H. Benko, A. D. Wilson, and P. Baudisch. Precise Selection Techniques for Multi-touch Screens. In Proc. CHI 2006, pp. 1263–1272. 2006.
 - [10] Katrin Wolf and Niels Henze. 2014. Comparing pointing techniques for grasping hands on tablets. In Proc. MobileHCI 2014, pp. 53–62. 2014.
 - [11] A. K. Eady and A. Girouard. Caret Manipulation Using Deformable Input in Mobile Devices. In Proc. TEI 2015, pp. 587–591. 2015.
 - [12] 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. In Proc. WISS 2014, pp. 121–122, 2014.
 - [13] Jean-Baptiste Scheibel, Cyril Pierson, Benoît Martin, Nathan Godard, Vittorio Fuccella, and Poika Isokoski. 2013. Virtual Stick in Caret Positioning on Touch Screens. In Proc. IHM 2013, pp. 107–114. 2013.
 - [14] Vittorio Fuccella, Poika Isokoski, and Benoît Martin. 2013. Gestures and widgets: performance in text editing on multi-touch capable mobile devices. In Proc. CHI 2013, pp. 2785–2794. 2013.
 - [15] Apple. iOS Human Interface Guidelines: 色とタイポグラフィ. <https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/MobileHIG/ColorImagesText.html>
 - [16] Lifehacker, Five Shortcuts for Faster Text Selection in iOS. <http://lifehacker.com/five-shortcuts-for-faster-text-selection-in-ios-1641490708>. 2015.
 - [17] J. Brooke. SUS-A quick and dirty usability scale. Usability evaluation in industry, London: Taylor and Francis. pp. 189–194, 1996.