

スマートフォンにおける傾きを利用した文字列操作手法

安藤 宗孝^{1,a)} 磯本 俊弥^{1,b)} 志築 文太郎^{2,c)} 高橋 伸^{2,d)}

概要：スマートフォンの片手操作時に文字列選択および選択した文字列を利用した操作（以降、文字列操作）を行うためには親指を画面上部に伸ばす必要があるが、親指の届く範囲は限られているためこれらの操作を自由に行う事は難しい。我々は、片手操作時に画面上部に親指を伸ばす必要なく、文字列操作が可能となる手法を開発した。我々の手法はソフトウェアキーボードのいずれかのキーを押下しつつスマートフォンを傾ける事によって傾きに応じたキャレット移動およびそれに伴う文字列の選択を行う。さらに、キーから指を離れた際に押下していたキーの種類に対応したカット、コピーなどの操作を選択した文字列に対して行う。我々の手法はソフトウェアキーボード部のみを用いて文字列選択が可能であるため、親指を上部に伸ばす必要がなく、さらに文字列操作を1タッチ内にて行う事が可能である。

1. はじめに

スマートフォン、タブレット端末などのタッチスクリーンを備えた端末において、文字列の選択、およびカット、コピーなどの文字列操作を行うためには、ユーザは文字列を直接ロングタップして選択し、文字列の周辺に表示されるメニューから目的の操作（カット、コピーなど）を選択する。この操作手法は、文字列操作を行いたい文字列を直接タッチすること、選択範囲の周辺にメニューが表示されることから理解しやすい。しかし、同時にいくつかの問題を抱えている。まず、操作を行う指が選択対象を隠す（オクルージョン）という問題がある。選択対象が隠れた場合、ユーザは対象を正確に選択する事は難しい。また、文字列操作では、選択対象である文字がボタンおよびチェックボックスなどのGUI部品よりも小さいため、オクルージョンが発生しやすい。次に、指よりも小さい選択対象が密集している場合に正確な選択が難しいという問題（Fat finger問題 [16]）がある。文字列操作では、操作対象が小さい事からこの問題が発生しやすい。最後に、スマートフォンの画面上部には片手持持時に指が届きにくい領域が存在する [1] ため、この領域に存在する文字列を直接タッチす

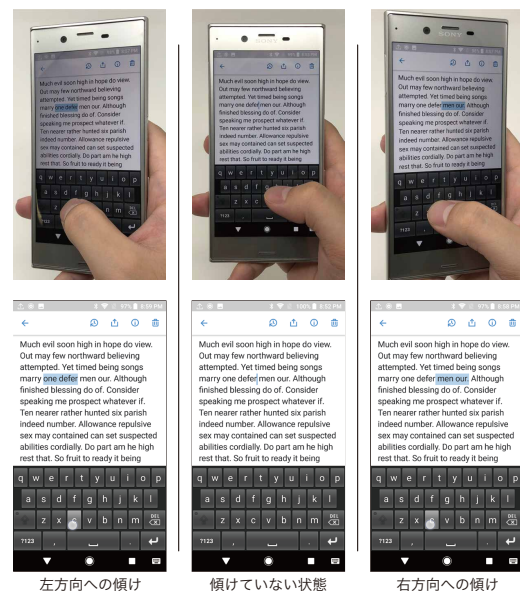


図1 スマートフォンの傾きによる選択範囲の変化。上段：スマートフォン傾き状態。下段：選択範囲。

る事は困難であるという問題がある。この領域を直接タッチするためにはユーザはスマートフォンの把持方法を変える必要がある。しかし、その動作は煩雑かつ不安定なものであるため、誤操作およびスマートフォン自体の落下の原因となる。

さらに、文字列操作の場合、コピー、カットなどのコマンドを選択する際にも、ユーザは指を伸ばす必要がある。また、近年多くの情報量が表示できることから大型のタッチスクリーンを持つ端末が市場に登場してきており、さらに片手での操作が難しくなっている。

我々はこれらの問題を解決するスマートフォン向けの文

¹ 筑波大学大学院システム情報工学研究科
Graduate School of Systems and Information Engineering,
University of Tsukuba, Tsukuba, Ibaraki 305-8573, Japan
² 筑波大学システム情報系
Faculty of Engineering, Information and Systems, University
of Tsukuba, Tsukuba, Ibaraki 305-8573, Japan
a) ando@iplab.cs.tsukuba.ac.jp
b) isomoto@iplab.cs.tsukuba.ac.jp
c) shizuki@iplab.cs.tsukuba.ac.jp
d) shin@iplab.cs.tsukuba.ac.jp

字列操作手法を示す。ユーザは、文字列選択および文字列操作を、1) ソフトウェアキーボードのキーを押下し、2) 押下中にスマートフォンを傾け (図 1)、3) その後にキーから指を離す事によって行う。キーを変えれば実行される文字列操作も変わる。この設計によって、ユーザは指を画面上部に伸ばす必要なくキャレット操作や文字列選択を行えるため、オクルージョンおよび Fat finger 問題が発生しない。また我々の手法は文字列選択および文字列操作を 1 回のタッチにて行う事を可能とする。さらに、大型のタッチスクリーンを持つ端末においても、ユーザはスマートフォンの把持方法を変える必要なく、これらの操作を行える。

本稿において、提案手法の概要、設計のために実施したスマートフォンにおける傾け角度の調査、実装、提案手法の性能を評価するために実施した Android OS を搭載したスマートフォンにおける標準の文字列操作手法 (以降、既存手法) との比較実験を述べる。

2. 関連研究

これまでに、スマートフォンの片手把持時に指が届きにくい範囲に対する操作手法、スマートフォンにおけるキャレット操作手法、文字列選択手法、および傾きを用いた操作手法の研究が多数行われてきている。これらの中の特に我々の手法に関連する研究について概説する。

2.1 指が届きにくい範囲に対する操作手法

スマートフォンにおいて指が届きにくい範囲に対する操作を可能とする手法がこれまでに数多く提案されている。TouchOver [22] は画面下部におけるタッチイベントを画面上部へと転送することによって、指が届かない画面上部に対する間接操作を可能とする。なお、スマートフォンを振るという操作によって画面下部への直接操作と画面上部への間接操作を切り替える。MagStick [13] は親指の動きと逆方向に移動するカーソルを提供する。iPhone には画面の上部にある項目を画面の下半分に移動し、画面上部の項目に対するタッチを行いやすくする簡易アクセス機能がある [6]。これらの手法は我々の手法と同じく、指が届きにくい範囲における操作を容易にする。しかし、文字列選択や文字列操作は文字入力に付随して行われることがあり、この場合、これらの手法においては頻繁に通常の操作との切り替えが発生する。これに対して我々の手法は主に文字入力中における画面上部における操作を容易にする。

2.2 キャレット操作、文字列選択、および文字列操作手法

キャレット操作および文字列選択のための手法がこれまでに提案されてきている。Scheibel ら [14] はタッチスクリーン上に表示したジョイスティックに基づくキャレット操作手法を示した。テキストを編集する実験からフォントサイズが大きい場合には既存手法よりも遅い事が示された

が、フォントサイズが小さい場合には既存手法との差はなかった。Fix and Slide [17] はドラッグによってキャレットを移動するのではなく、画面上のキャレットの位置を固定した状態にて、文字列全体をドラッグによって移動する事にてキャレットの位置を移動する手法である。iPhone には圧力を測定する事ができるタッチスクリーン上にて、ソフトウェアキーボードを強くタッチしドラッグすることによってキャレット移動を行う事ができる機能がある。また、この機能はタッチする圧力を変えることによりキャレット移動に伴う文字列選択も可能である [7]。これらの手法は我々の手法と同じく、キャレット操作または文字列選択が可能な手法である。我々の手法は、傾きを用いて文字列の選択を行う点、および文字列選択後の文字列操作を容易にする点がこれらの手法と異なる。

さらに、文字列選択後の文字列操作を容易にする手法も研究されている。Gestures and Widgets [4] はソフトウェアキーボード上にて左右スワイプ、円を描くなどのジェスチャを行う事によってキャレットの移動、文字列選択、またはコピーなどを行う手法である。この手法は我々の手法と同じく、指を伸ばす必要なく文字列操作を行う事ができる。この手法においてはジェスチャを用いて文字列操作を行うが、我々の手法はソフトウェアキーボードのキーに対応した操作を行うため、より多くの文字列操作を容易に行うことが可能である。PalmTouch [8] は画面上部をタッチしようとする際に手のひら部分がタッチスクリーンに触れることから、親指または手のひらによるタッチを識別し、この組み合わせを用いて操作を拡張する手法である。この手法の使用例の 1 つとして、手のひらの位置を起点としてコピー、ペーストなどのメニューを親指の届きやすい範囲に表示した。この手法は我々の手法と同じく、文字列操作をより容易にする手法である。しかし、直接タッチする手法であるため、オクルージョンおよび Fat finger 問題の発生を抑えることはできない。

2.3 傾きを用いた操作手法

これまでに多数の端末の傾きを用いた手法が研究されている。Rekimoto [12] は端末自体を傾ける事を端末への入力に使用する手法を示した。この手法は傾きとボタンを組み合わせる事によって縦型および円形のメニュー操作を可能とする。KeyTilt [5] はソフトウェアキーボードにおいて、1 つのキーに 5 つの文字を割り当て、キーをタッチしつつ端末を傾ける事によって 5 つの文字から 1 つを選択する手法である。この手法では、8 つのキーを用いてアルファベットおよび特殊文字を入力できるため、キー 1 つあたりを大きく表示することが可能である。TiltType [10] は手首に装着可能な端末において、8 方向に配置した文字それぞれの方向に傾ける事によって文字入力を行う。また、ユーザは端末の上下に配置されたボタンを押すことによって配

置される文字を変更することが可能である。Yeo ら [21] はジェスチャキーボードにおいて、キーボード上に傾きに応じて移動するポインタを表示し、指を用いてキーをなぞる動きを傾きによるポインタの移動に置き換えた入力手法を示した。Tilt & Touch [2] は傾け操作を 3 次元映像を見る際の視点操作に用いる手法である。TiltText [20] はフィーチャーフォンにおける文字入力の際、傾け操作を用いる事によってボタンを押す回数を減らす手法である。Tsandilas ら [19] はスマートフォンの傾きに応じて、タッチスクリーン上の表示を傾ける事により、画面上の指から遠い位置に対するタッチを容易にする手法を示している。提案手法はこれらの手法と同じく、傾きを用いて操作を行う手法であるが、提案手法は文字列操作を目的としている点がこれらの手法と異なる。

また、傾きを用いた操作手法の性能評価も行われている。Rahman ら [11] はフィーチャーフォンを用いて傾け可能な範囲およびその範囲における傾け操作の精度を調査し、調査結果を用いた傾け操作のための設計指針を示した。Fitchett ら [3] は iPod touch を用いた実験を行い、傾け操作を用いたスクロールとフリックを用いたスクロールを比較した。この結果、座位において傾け操作を用いたスクロールの方がフリックを用いたスクロールよりも素早く操作可能であった。Teather ら [18] はタブレット端末における傾け操作において、傾きから画面上の位置を決定する実装および傾きからカーソルの移動速度を決定する実装の比較を行った。この結果、傾きから画面上の位置を決定する実装がもう一方の実装と比較して約 2 倍のパフォーマンスを示した。Shima ら [15] はスマートウォッチにおける傾け操作の精度調査を行っている。これらの性能評価を参考に実装を行う。

3. 提案手法

我々はキーを押下しながらスマートフォンを傾けるといふ操作をキャレット操作および文字列選択に用いる。ユーザは、キャレット操作および文字列選択を、1) ソフトウェアキーボードのキーを押下し、2) 押下中にスマートフォンを傾け (図 1)、3) その後キーから指を離す事によって行う。キーを変えれば実行される文字操作も変わる。例として、提案手法を用いて文字列をコピーする手順を図 2 に示す。まず、ユーザは *m* キーを押下した状態にてスマートフォンを傾ける事によりキャレットを移動させる (図 2a-c)。次に、*c* キーを押下した状態にてスマートフォンを傾ける事によりコピーしたい文字列を選択し、指を離す (図 2d および e)。これらの手順によって文字列をコピーすることができる。

以下に、キャレット操作と文字列選択、および文字列操作それぞれを説明する。

表 1 文字列操作のキーへの割り当て例。

キー	文字列操作
<i>c</i>	コピー
<i>v</i>	選択範囲を上書きしてペースト
<i>Shift</i>	大文字と小文字を変換
<i>b</i>	文字列をブラウザ検索
<i>s</i>	文章内を選択した文字列で検索
<i>l</i>	文字列を用いたアプリケーションの起動
<i>m</i>	キャレット操作

3.1 キャレット操作および文字列選択

我々の手法を用いてキャレット操作を行うためには、ソフトウェアキーボードの *m* キーを押した状態にてスマートフォンを傾ける。この際のスマートフォン傾きに応じてキャレットが移動する。アルファベットを入力するキーとして最も右下に近い位置に存在する事、*m* キーから move を連想することが可能である事から *m* キーをキャレット操作を行うキーとして設計した。

文字列の選択を行うためには、ソフトウェアキーボードの *m* 以外のキーを押した状態にてスマートフォンを傾ける。この際のスマートフォンの傾きに応じて選択範囲が変化する。現状における、キャレット操作および文字列の選択は横方向のみに利用可能である。

3.2 文字列操作

選択した文字列を用いて文字列操作を行う。文字列操作は文字列選択時に押下したキーを離すことによって実行できる。キーを離した際に選択していた文字列を用いて文字列操作を行う。なお、傾けた後にキーを離した場合は通常の文字入力は行わない。押下していたキーによって異なる文字列操作を行う事ができる。表 1 にキーへの文字列操作の割り当て例を示す。表 1 に示すように様々な文字列操作を割り当てることが可能である。また、*m* キーを用いたキャレット操作は *m* キーを押した際の文字列操作として捉える事が可能であり、手法としての一貫線を持たせている。ソフトウェアキーボードのキーそれぞれに文字列操作を割り当てることが可能であるため、PC におけるキーボードショートカットの様に多くの機能を実現することができる。また、キートップの文字に関連する文字列操作を割り当てる事により、キートップから文字列操作を想起することが可能である。表 1 において *l* キーに割り当てられている、「文字列を用いたアプリケーションの起動」とは、選択されている文字列に関連するアプリケーションを起動するものである (例、選択された文字列: wifi, 起動するアプリケーション: wifi の設定)。文字列を用いたアプリケーションの起動を用いる事によってソフトウェアキーボードをアプリケーションランチャの様に表示する事が可能である。

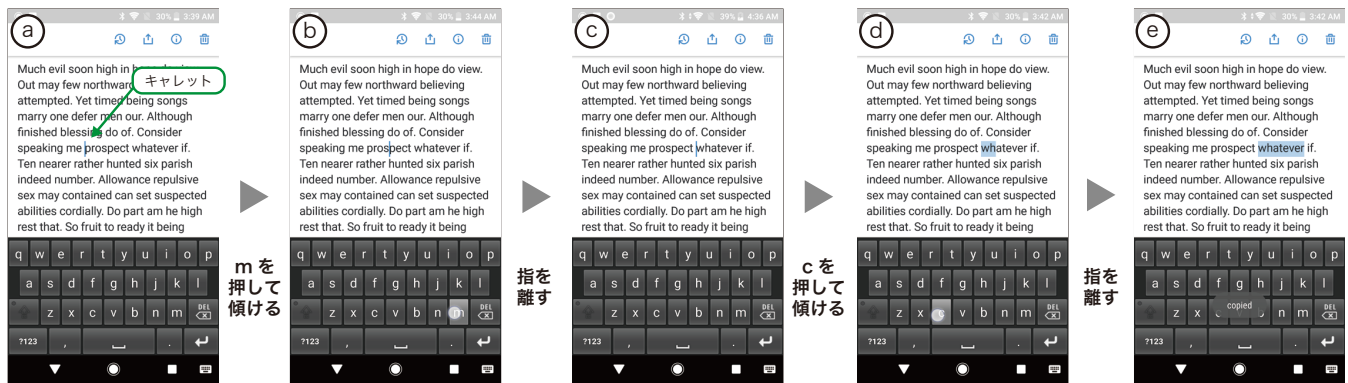


図 2 提案手法を用いた文字列操作の流れ. a) 傾けていない状態. b) *m* を押下しつつ右に傾けている状態. キャレットが右に移動している. c) *c* キーから指を離した状態. d) *c* キーを押下しつつ右に傾けている状態. 選択範囲が右方向に拡張されている. e) *c* キーから指を離した状態. 選択されている文字列 (whatever) がコピーされた.

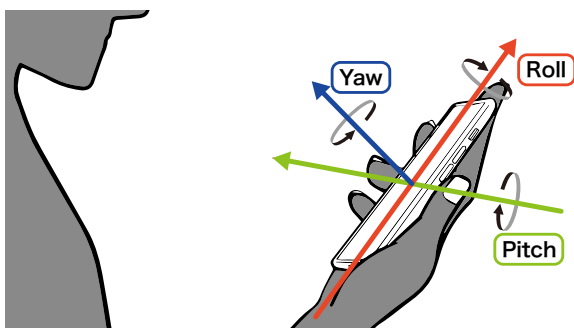


図 3 スマートフォンの傾け軸.

4. 傾け角度の調査実験

我々の手法を実現するために、スマートフォンを傾けた際の角度の調査実験を行った。本実験の目的はスマートフォンを左方向、右方向、上方向、および下方向（以降、左右上下方向）に傾けた際の *Roll*、*Pitch*、および *Yaw* (図 3) の変化を調査する事である。本実験においてはスマートフォンを片手持ちし、左右上下方向に傾けた際のスマートフォンの角度を計測した。本実験にて計測した角度を我々の手法の実装に利用する。

4.1 実験協力者および実験機器

大学生および大学院生のボランティア 12 名（男性 9 名、女性 3 名、平均年齢 23.7 歳）を実験協力者とした。すべての実験協力者が日常的にスマートフォンを使用していた。また、すべての実験協力者が普段スマートフォンを右手にて把持していた。実験開始時に実験協力者の手長を計測した。この結果、平均 18.4 cm（最長 20.1 cm、最短 16.3 cm）であった。

実験にて使用するスマートフォンとして Xperia XZ（端末サイズ：高さ 146 mm × 幅 72 mm × 厚さ 8.1 mm、画面サイズ：5.2 インチ、OS：Android 8.0.0）を用いた。

4.2 実験内容

実験は椅子に座った状態にて行った。実験協力者には普段スマートフォンを持つ持ち方にて、普段画面を見ている角度でスマートフォンを右手を用いて把持してもらった（以降、基本姿勢）。その後、基本姿勢から左方向、右方向、上方向、および下方向の順に傾けてもらった。なお、それぞれの方向に傾ける前に基本姿勢に戻すよう指示した。実験協力者間にて左右または上下が逆になる事を防ぐために、実験協力者に 4 方向の傾け方向を指定した。本実験におけるスマートフォンを傾ける範囲の指示として、実験協力者にはスマートフォンを傾けた際に画面上の文字列とキャレットの位置が無理なく確認できる範囲において傾けるように指示した。これは、我々の手法はスマートフォンの画面を見て操作を行うため、キャレットおよび文字列を確認不可能な範囲まで傾けた場合における角度は利用できないためである。傾け範囲の指示と同様に、本実験は我々の手法に利用する事を目的としているため、実験協力者には *m* キーを押下した状態にてスマートフォンを傾けた後、前述した傾け範囲内にて最も傾けた状態で指を離すよう指示した。指を離した際の角度を傾けた際の最大の角度として記録した。すなわち指を画面に押下した状態における傾き角度を取得した。

左方向、右方向、上方向、下方向の順にそれぞれ傾ける事を 1 セッションとし、1 人あたり 5 セッション行った。傾け 1 回を 1 試行とすると、合計 240 試行（4 方向 × 5 セッション × 12 人）を行った。

4.3 実験結果および考察

実験結果を表 2 に示す。今回は左右上下方向の中から、左右方向に傾けた場合に注目する。表 2 から、左右に傾けた場合において、*Roll* だけでなく、*Pitch* および *Yaw* が変化している事がわかる。左右に傾けた場合において、*Roll* のみが傾け角度の正負が逆転しており、*Pitch* および

表 2 傾け調査の結果. 上下左右: 基本姿勢と比較した際の傾きの変化量. 基本姿勢: 基本姿勢におけるスマートフォン把持角度.

傾け方向	Roll	Pitch	Yaw
左	-61.34°	-24.80°	25.30°
右	56.78°	-22.19°	15.35°
上	121.85°	2.39°	34.06°
下	-10.33°	-31.81°	9.53°
基本姿勢	13.03°	67.60°	-37.44°

Yaw においては傾け角度の正負は同一である事が分かった。また、左右に傾けた際、左方向に傾ける方がすべての回転角において変化量が大きくなる事が分かった。これは、すべての実験協力者が右手把持であったことが原因であると考えられる。画面上の文字およびキャラットの位置が無理なく確認できる範囲にて傾ける事を指示していたため、右手把持の場合、右手と頭の位置関係から左側に傾けた場合の方が右側に傾けた場合よりも画面を確認できる範囲が広がった可能性がある。このため、スマートフォンの操作に主に左手を使う実験協力者を対象とした調査を行った場合、異なる結果が得られると考えられる。

5. 実装

傾け角度の調査実験にて得た左右方向への傾け時の角度を用いて、傾きを用いたキャラット操作および選択範囲の変更の実装を行う。

5.1 傾きによる速度決定と位置決定の比較

傾きを用いてキャラット操作および選択範囲の変更を行う際、2つのキャラット位置および選択範囲の決定方法がある。1つは、傾け操作をキャラットや選択範囲の移動速度決定に用いる手法である。もう1つは、傾け操作をキャラットや選択範囲の位置決定に用いる手法である。これまでに、これらの比較を行った研究が複数存在する。Theatherら [18] はタブレット端末におけるポインティングタスクを用いてこれらの手法を比較している。結果として、傾きを位置決定に用いる手法が移動速度の決定に用いる手法の約2倍の速度であった。Oakleyら [9] は手のひらサイズのコンピュータにおけるメニュー操作タスクを用いてこれらの手法を比較した。結果、傾きを位置決定に用いる手法の方が移動速度の決定に用いる手法に比べて素早く操作可能である事を示した。これらの小型端末を用いた先行研究における実験の結果を参考とし、我々の手法においては傾きをキャラット位置および選択範囲の位置決定に用いる。

5.2 キャラットおよび選択範囲の位置決定

傾け角度の調査実験の結果を考慮してキャラットおよび選択範囲の位置決定手法を実装する。傾きを用いた画面上の位置の決定方法として、三角関数を用いて位置決定を行

う手法が先行研究にて示されている [15]。この手法では傾け角度 θ の場合のキャラット位置 (CP) は式 (1) を用いて求める事ができる。

$$CP = \sin \theta \times cd_ratio \quad (1)$$

cd_ratio は定数であり、ディスプレイ上のキャラット移動距離に対する傾け量の比である。しかし、この方法を用いてキャラット位置を決定した場合、アルファベットの文字幅が一定でない事から選択しやすい文字と選択しにくい文字が発生する。このため、今回の実装においては、1文字あたりの傾け量が等しくなるように実装する必要がある。また、傾け角度の調査実験から、左右において傾ける事ができる角度の大きさが異なる事が分かったため、これも考慮して実装する。

このため、今回の実装においては以下の手順にてキャラット位置を決定する。

- (1) 左右どちらへ移動するかは *Roll* を用いて決定する。
- (2) 左右ごとにキャラット位置を決定する。

まず、左右の決定においては、左右方向に傾けた際 *Roll* のみが左右対照的に変化することが分かっているため、これを利用する。つまり、*Roll* の値が正負のどちらに変化したかを用いてキャラット位置の変化方向を決める。次にキャラット位置は、式 (2) を用いて求める。

$$CP = \sqrt{\left(\frac{nR}{mR}\right)^2 + \left(\frac{nP}{mP}\right)^2 + \left(\frac{nY}{mY}\right)^2} \times cc_ratio \quad (2)$$

nR , nP , および nY はそれぞれ、現在の *Roll*, *Pitch*, および *Yaw* であり、 mR , mP , mY はそれぞれ傾き角度の調査実験にて得たそれぞれの最大角度である。 cc_ratio は定数であり、キャラットの移動文字数に対する傾け量の比である。今回の実装においては経験的に $cc_ratio = 20$ とした。

6. 文字列操作実験

提案手法と既存手法における文字列選択および文字列操作にかかる時間を比較する実験を行った。本実験は、キャラット操作、文字列選択、および文字列操作を含めた一連の操作にかかる時間を計測し、比較した。

6.1 実験協力者および実験端末

実験協力者として大学院生6名(男性4名, 女性2名, 平均年齢23.3歳, P1-P6)を雇用した。このうち4節にて述べた傾け角度の調査実験に参加していたのは4名(P1-P3, P5)であった。すべての実験協力者が日常的にスマートフォンを使用していた。また、すべての実験協力者が普段スマートフォンを右手にて把持していた。実験開始時に実験協力者の手長を計測した。この結果、平均18.3cm(最長19.2cm, 最短16.3cm)であった。実験にて使用するスマートフォンは4節と同じくXperia XZを用いた。



図 4 実験アプリケーション。

6.2 実験内容

実験協力者には着座姿勢にてこの実験を行ってもらった。実験協力者には2種類の手法を用いて文字列操作を行うよう指示した。図4に実験において使用したアプリケーションの画面を示す。図4における文字列Aが選択対象文字列である。図4において、文字列Aは文字列Bの部分文字列となっており、実験協力者は文字列B中の文字列Aの部分に対して指示された手法を使用して文字列操作を行う。以下に、2種類の文字列操作手法の実験における操作内容を示す。

提案手法 実験協力者は、まず、文字列選択の開始点を変更するために、選択対象文字列の端にカーレットを移動させる。これは m キーを押下しながらスマートフォンを傾ける事により行う。次に、文字列の選択を行う。これは m キー以外のキーを押下しながらスマートフォンを傾ける事により行う。傾け角度を調整することによって選択範囲を調整する。最後に、文字列を選択した状態にて指を離すことにより、選択した文字列に対して文字列操作が行われる。

既存手法 実験協力者は、まず、文字列を直接ロングタップする事によって文字列を選択状態にする。この状態にて、選択範囲を示すアイコンをドラッグし選択範囲を変更する。次に、文字列を選択した状態にて指を離すことにより、選択した文字列の上部にメニューが表示される。メニューの中から *Next* をタップする事によって選択した文字列に対して文字列操作が行われる。

文字列操作を行った際に、選択対象文字列に対して正しく文字列操作が行われた場合には、文字列が更新され、次の文字列が表示される。選択されている文字列が選択対象文字列と異なっている場合には、文字列は更新されない。文字列を正しく1回選択する事を1試行とし、10試行を1セッションとする。セッションは実験協力者が最初の文字

列に対し、文字列操作を正しく行った直後から開始される。実験協力者には手法ごとに連続して10セッション行ってもらった。よって各実験協力者は合計200試行(=10試行×10セッション×2手法)を行った。また、手法間の順序効果を打ち消すために、実験協力者を2つのグループに分け、片方のグループは提案手法を先に行い、もう片方のグループは既存手法を先に行った。疲労による影響を軽減するために、実験協力者にはセッション間に1分以上の休憩を取ってもらった。本実験は1人あたり約75分の時間を要した。実験に使用した文字列はランダムに生成されたものであり*1、1文あたり平均8.8単語である。また、文字列更新後における選択対象文字列の探索時間を減らすために、文字列B中の選択対象文字列は文字色を青色に設定した。なお、既存手法においては自然言語処理を用いた機能により単語の端に選択範囲が吸着するが、設定の変更は行わず、既定の仕様にて使用している。

6.3 実験結果および考察

図5に各手法ごとの各セッションにおける全実験協力者の文字列操作にかかった時間を示す。また、合わせて手法ごとの累乗近似曲線を示す。10セッションのデータを用いて対応のある t 検定を行った場合、提案手法と比較して既存手法間の方が有意にかかった時間が短かった ($p = .00 < .05$)。今回の実験においては、練習セッションなしに本番セッションを行っているため、セッション数を重ねるごとに操作時間が短くなっている。特に、提案手法において顕著にセッション数による操作時間の変化している。このため、セッション数の経過による影響を検証するため、10セッションを5分割し、それぞれにおいて提案手法と既存手法を比較した。結果として、1-2セッションは有意差あり ($p = .016 < .050$)、3-4セッションは有意差無し ($p = .072 > .050$)、5-6セッションは有意差あり ($p = .022 < .050$)、7-8セッションは有意差無し ($p = .215 > .050$)、9-10セッションは有意差無し ($p = .072 > .050$) となった。この結果から、ばらつきはあるが、セッション数を重ねるごとに提案手法と既存手法との差が減少すると考えられる。

7. 議論と今後の課題

提案手法に関する議論および今後の課題について述べる。

7.1 文字による影響

提案手法および既存手法は、文字の大きさや幅といった文字の違いによって操作速度および操作のしやすさなどに影響を受けると考えられる。

7.1.1 文字の大きさ

本稿にて行った文字列操作実験は、文字の大きさを実験

*1 <http://randomtextgenerator.com>

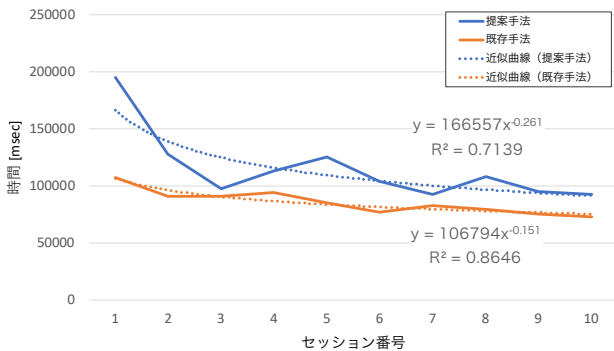


図 5 各セッションにおける全実験協力者の平均試行時間。

にて使用したスマートフォンの標準に設定して行った。標準以外の文字の大きさの場合、提案手法および既存手法の両方において、文字列選択の行いやすさ、またはキャラットの見やすさなどが変化する可能性が高いため異なる実験結果が得られる可能性がある。先行研究 [14], [17] においても文字の大きさが異なる条件にて実験が行われており、今後、我々の手法を用いた場合においても文字の大きさの影響を検証する必要がある。

7.1.2 文字幅

フォントによって文字幅が異なるため、これも考慮する必要がある。特に等幅フォントはすべての文字幅が同じであるため、提案手法にて考慮しているアルファベットごとの文字幅の差を考慮する必要が無い。このため、今回の実装にて採用しなかった先行研究にて使用されている手法 [15] を利用できる。これらの事から、文字幅を考慮した実装を行い、検証する必要がある。

7.2 ユーザの状況および周囲の環境による影響

ユーザが歩いているまたは走っているなどの状況においては、スマートフォンのセンサ値の変化によって、キャラット操作および文字列選択において予期せぬ動作が発生する可能性がある。また、ユーザが仰向けに寝ている状態においては、基本姿勢からの角度の変化方向が異なるため、現在の実装では提案手法が動作しないと考えられる。これと同様に、エレベータまたは電車などユーザの置かれている環境によってセンサ値が変化する場合においても予期せぬ動作が発生する。例として、著者がエレベータの中にて提案手法を用いたキャラット操作を行った場合、キャラットの位置が不用意に移動する事を確認した。このため、ユーザの状況および周囲の環境からの提案手法への影響を調査し、それらの状況において使用可能な設計を検討する。

7.3 実装の改善

提案手法を用いてより素早く正確に文字列操作を行うため、実装について改良が必要である。実験において実験協力者から複数のコメントを得たため、得られたコメントを

参考として実装を改善する。

7.3.1 上下方向のキャラット操作および文字列選択

現状の実装においてはキャラット移動、および文字列選択が左右方向のみであるため、これを上下方向に関しても実装する。4 節にて、上下方向に傾けた際の角度も取得しており、これを用いて実装する事が可能である。この実装により縦方向のキャラット移動および文字列選択が可能となれば、2 行以上の文字列を対象とした操作の場合に操作速度の向上が見込める。

7.3.2 傾きを用いた文字列選択の改善

実験協力者からキャラット移動に関して「1 文字ずれる事が多い」、「動きすぎる」などのキャラットの動作に関するコメントを得た。これらのコメントからキャラットの移動速度を最適化する必要がある。このため、今回の実装においては経験的に決定したキャラットの位置決定のための定数 (*cc_ratio*) に関して最適値を調査する必要がある。具体的には、傾け操作に対する慣れの度合いによって動的に定数を変化させる事が考えられる。また、文字列選択に関して「空白を選択しないようにしてほしい」、「単語単位の選択がよい」など、文字列選択の効率化に関するコメントも得られた。このため、自然言語処理を行い、不必要な空白の削除や、まず単語単位の選択を行い詳細な選択を必要とする場合のみ文字単位の選択を行う段階的な選択機能を実装し、より効率的な文字列選択を実現する。

7.4 キャラット操作および文字列選択手法の変更

本稿における実装においては、キャラット操作および文字列選択をスマートフォンの傾きを利用して行った。しかし、我々の手法の利点の 1 つであるソフトウェアキーボードのキーを利用した文字列操作は異なる方法にて文字列選択を行った際にも利用できる。例として、ソフトウェアキーボードのキーを押下した後、押下したままキーボード上にてジェスチャを行い、指を離すと最初に押したキーに割り当てられた文字列操作が行われる設計、もしくは、キーを押下したまま端末背面を用いて選択範囲を決定し、キーを離した際に文字列操作が行われる設計などが考えられる。複数の設計を考慮し、最適な設計を検討する。

8. おわりに

本稿にて、スマートフォンの傾きおよびソフトウェアキーボードを用いて、文字列選択および文字列操作を行う手法を示した。提案手法は指を画面上部に伸ばすことなく文字列選択および文字列操作を行えるため、オクルージョンおよび Fat finger 問題が発生しない。また、提案手法は 1 タッチ内にて文字列選択および文字列操作を行う事を可能とする。我々は提案手法を実現するために、スマートフォン片手持持時における傾け角度の調査実験を行った。結果として、左右方向に傾けた際、Roll, Pitch, Yaw の内、

Roll のみが傾きの変化量が対照的に変化する事, 左右方向への傾けであっても Pitch, Yaw も変化する事が分かった. この実験の結果を用いて, 提案手法を実装した. さらに実装した提案手法と既存手法の文字列選択および文字列操作に要する時間を比較した結果, 全体としては既存手法の方が有意に早かったが, 提案手法に慣れるに従って提案手法と既存手法間に差が無くなる事が分かった.

参考文献

- [1] Joanna Bergstrom-Lehtovirta and Antti Oulasvirta. Modeling the functional area of the thumb on mobile touchscreen surfaces. In *Proceedings of the 32nd Annual ACM Conference on Human Factors in Computing Systems*, CHI '14, pp. 1991–2000, New York, NY, USA, 2014. ACM.
- [2] Yuan Du, Haoyi Ren, Gang Pan, and Shjian Li. Tilt & Touch: Mobile phone for 3D interaction. In *Proceedings of the 13th International Conference on Ubiquitous Computing*, UbiComp '11, pp. 485–486, New York, NY, USA, 2011. ACM.
- [3] Stephen Fitchett and Andy Cockburn. Evaluating reading and analysis tasks on mobile devices: A case study of tilt and flick scrolling. In *Proceedings of the 21st Annual Conference of the Australian Computer-Human Interaction Special Interest Group*, OzCHI '09, pp. 225–232, New York, NY, USA, 2009. ACM.
- [4] Vittorio Fucella, Poika Isokoski, and Benoit Martin. Gestures and Widgets: Performance in text editing on multi-touch capable mobile devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pp. 2785–2794, New York, NY, USA, 2013. ACM.
- [5] Guillaume Homberg, Alassane Sangott, and Benoit Martin. KeyTilt: Un clavier logiciel par inclinaison. In *Proceedings of the 22nd Conference on L'Interaction Homme-Machine*, IHM '10, pp. 137–140, New York, NY, USA, 2010. ACM.
- [6] Apple Inc. iPhone ユーザガイド - 簡易アクセス, 2014. <https://help.apple.com/iphone/11/?lang=ja#/iph66e10a71c> (2018年7月25日閲覧).
- [7] Apple Inc. 3D Touch - iOS - Apple Developer, 2015. <https://developer.apple.com/ios/3d-touch/> (2018年7月25日閲覧).
- [8] Huy Viet Le, Thomas Kosch, Patrick Bader, Sven Mayer, and Niels Henze. PalmTouch: Using the palm as an additional input modality on commodity smartphones. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, pp. 360:1–360:13, New York, NY, USA, 2018. ACM.
- [9] Ian Oakley and Sile O'Modhrain. Tilt to Scroll: Evaluating a motion based vibrotactile mobile interface. In *Proceedings of the First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, World Haptics 2005, pp. 40–49. IEEE, 2005.
- [10] Kurt Partridge, Saurav Chatterjee, Vibha Sazawal, Gaetano Borriello, and Roy Want. TiltType: Accelerometer-supported text entry for very small devices. In *Proceedings of the 15th Annual ACM Symposium on User Interface Software and Technology*, UIST '02, pp. 201–204, New York, NY, USA, 2002. ACM.
- [11] Mahfuz Rahman, Sean Gustafson, Pourang Irani, and Sriram Subramanian. Tilt Techniques: Investigating the dexterity of wrist-based input. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pp. 1943–1952, New York, NY, USA, 2009. ACM.
- [12] Jun Rekimoto. Tilting operations for small screen interfaces. In *Proceedings of the 9th Annual ACM Symposium on User Interface Software and Technology*, UIST '96, pp. 167–168, New York, NY, USA, 1996. ACM.
- [13] Anne Roudaut, Stéphane Huot, and Eric Lecolinet. Tap-Tap and MagStick: Improving one-handed target acquisition on small touch-screens. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, AVI '08, pp. 146–153, New York, NY, USA, 2008. ACM.
- [14] Jean-Baptiste Scheibel, Cyril Pierson, Benoît Martin, Nathan Godard, Vittorio Fucella, and Poika Isokoski. Virtual stick in caret positioning on touch screens. In *Proceedings of the 25th Conference on L'Interaction Homme-Machine*, IHM '13, pp. 107:107–107:114, New York, NY, USA, 2013. ACM.
- [15] Keigo Shima, Kazusa Onishi, Ryosuke Takada, Takuya Adachi, Buntarou Shizuki, and Jiro Tanaka. Investigating accuracy of tilting operation on wrist-worn devices with touchscreens. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, CHI EA '16, pp. 2705–2711, New York, NY, USA, 2016. ACM.
- [16] Katie A. Siek, Yvonne Rogers, and Kay H. Connelly. Fat finger worries: How older and younger users physically interact with pdas. In *Proceedings of IFIP Conference on Human-Computer Interaction*, INTERACT '05, pp. 267–280, 2005.
- [17] Kenji Suzuki, Kazumasa Okabe, Ryuuki Sakamoto, and Daisuke Sakamoto. Fix and Slide: Caret navigation with movable background. In *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services*, MobileHCI '16, pp. 478–482, New York, NY, USA, 2016. ACM.
- [18] Robert J. Teather and I. Scott MacKenzie. Position vs. velocity control for tilt-based interaction. In *Proceedings of Graphics Interface 2014*, GI '14, pp. 51–58. Canadian Information Processing Society, 2014.
- [19] Theophanis Tsandilas, Caroline Appert, Anastasia Bezerianos, and David Bonnet. Coordination of tilt and touch in one- and two-handed use. In *Proceedings of the 32nd Annual ACM Conference on Human Factors in Computing Systems*, CHI '14, pp. 2001–2004, New York, NY, USA, 2014. ACM.
- [20] Daniel Wigdor and Ravin Balakrishnan. TiltText: Using tilt for text input to mobile phones. In *Proceedings of the 16th Annual ACM Symposium on User Interface Software and Technology*, UIST '03, pp. 81–90, New York, NY, USA, 2003. ACM.
- [21] Hui-Shyong Yeo, Xiao-Shen Phang, Steven J. Castellucci, Per Ola Kristensson, and Aaron Quigley. Investigating tilt-based gesture keyboard entry for single-handed text entry on large devices. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, pp. 4194–4202, New York, NY, USA, 2017. ACM.
- [22] 大西主紗, 志築文太郎, 田中二郎. Touchover:大画面を備える携帯情報端末を楽に操作するための片手親指操作手法. 第22回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論文集, pp. 85–90. 日本ソフトウェア科学会, 2014.