

モバイル端末のためのホバー機能を用いた 文字入力支援インタフェース

相吉澤 聖明^{1,a)} 小室 孝^{1,b)}

概要: 本稿は、モバイル端末のホバー機能を用いて指の位置に応じてソフトウェアキーボードの一部を拡大することで、文字入力におけるミスタイプを減らすことができる入力支援インタフェースを提案する。最初に指を検出した位置に拡大中心を固定することにより、指を動かしてもキーが動かないため、安定した選択操作が可能となる。拡大中心を固定した手法、拡大中心を固定しスクロール機能を追加した手法、拡大中心を固定しない手法、通常のソフトウェアキーボードの4つの手法で比較評価を行った。その結果、拡大中心を固定した手法は、入力速度は遅くなったが他の手法に比べエラー率が低くなった。これにより、拡大中心を固定することにより、ユーザは安定して選択操作を行うことができ正確にターゲットキーを入力することができることが示された。

A Text Input Support Interface for Mobile Devices Using a Hover Function

AIYOSHIKAWA TOSHIKI^{1,a)} KOMURO TAKASHI^{1,b)}

Abstract: In this paper, we propose a text input support interface for mobile devices such as smartphones to reduce mistypes, in which part of a software keyboard according to the position of the finger is enlarged by using a hover function. By fixing the center of enlargement to the position at which the finger is detected for the first time, keys do not move even when a user moves his/her finger, which makes it possible for him/her to select keys stably. we compared four methods: fixing the center of enlargement, not fixing the center of enlargement, adding a scrolling function, and using a standard software keyboard for reference. The method in which the center of enlargement was fixed showed the slowest text entry speed, but showed an error rate that was significantly lower than that of the method in which the center of enlargement was not fixed. This result shows that fixing the center of enlargement enables users to select keys stably and to input target keys accurately.

1. はじめに

近年、スマートフォンなどのタッチパネルを搭載したモバイル端末は持ち運びがしやすく多機能なため、様々な用途で用いられている。しかし、モバイル端末は表示画面が小さく、小さなオブジェクトを選択することが困難である。特に、ソフトウェアキーボードにおいては画面内に小さなキーが密集して配置されているためミスタイプが生じ

やすい。

この問題を解決する先行研究として、拡大を用いた入力支援がある。ユーザがモバイル端末の画面に対してタッチまたはスワイプ操作を行うことによりキーを拡大して、ミスタイプを減らすことができるインタフェースが提案されている。関連研究 [1,2] は、ユーザは文字を1文字入力するごとに画面に対してタッチまたはスワイプ操作を行い、キーを拡大する必要がある。そのため、これらのインタフェースはキーを拡大するための操作をユーザに強いることになる。一方、吹き出しを用いてタッチした領域の映像を拡大表示するインタフェースが提案されている [3,4]。しかし、これらのインタフェースは、ユーザが画面をタッ

¹ 埼玉大学 埼玉県さいたま市桜区下大久保 255
Saitama University 255 Shimo-okubo, Sakura-ku, Saitama
City, 338-8570, Japan

^{a)} aiyoshizawa@is.ics.saitama-u.ac.jp

^{b)} komuro@mail.saitama-u.ac.jp

チした後にタッチした領域が拡大表示される。拡大表示後に選択操作を行うため、ターゲットオブジェクトを選択するまでに時間が掛かる。また、1つのボタンに複数の文字を割り当て、ボタンの数を減らして1つあたりのボタンサイズを大きくした研究がある。関連研究 [5] は、1つのボタンに2つの文字を割り当て、人差し指と中指の2本の指を使用することで、それぞれの文字を入力することができる。しかし、ユーザは入力する文字ごとに指を切り替える必要があり、入力までに時間が掛かる。また、関連研究 [6] は、スワイプ操作を用いて1つのボタンに割り当てられた複数の文字を入力することができる。しかし、このインタフェースは、キーの配置が従来の QWERTY 配列と異なるため、ユーザは新しいキー配置を覚える必要がある。

静電容量方式のタッチパネルを使用してタッチスクリーン上方の指の位置を取得することができるモバイル端末のホバー機能がある。ホバー機能を使用して、ユーザが画面に直接タッチすることなく選択操作を支援するインタフェースの研究がある。ホバー機能を使用して、ユーザが拡大されたキーを選択することができるインタフェースが提案されている [7]。このインタフェースは、ホバー機能が指を検出している間、指直下周辺のキーを常に拡大する。しかし、拡大中心が指の動きに伴い移動するため、キーが指の動きと反対方向に動く。これにより、不安定な選択操作を招く可能性がある。一方で、拡大中心を固定することにより安定した選択操作を可能にしたインタフェースが提案されている [8,9]。これらのインタフェースは、指を画面に近づけたとき、ターゲット位置周辺のオブジェクトを拡大する。しかし、いくつかのオブジェクトが拡大されたとき、それ以外のオブジェクトが画面に表示されない。そのため、これらのインタフェースは、次のターゲットオブジェクトが画面に表示されていないとき、ターゲットオブジェクトを探すための余分な操作をユーザに強いることになる。一方で、指の動きに伴い移動するカーソルを画面に表示することにより正確にターゲットオブジェクトの選択を可能にするインタフェースが提案されている [10]。このインタフェースは、幅 2-3 mm 程度のオブジェクトを選択するときには効果的だが、ソフトウェアキーボードに対しても効果的かどうかは定かでない。

そこで本稿では、モバイル端末のホバー機能を用いて指直下周辺のソフトウェアキーボードの一部を拡大することで、文字入力におけるミスタイプを減らすことができる入力支援インタフェースを提案する。最初に指を検出した位置に拡大中心を固定することにより、指を動かしてもキーが動かないため、安定した選択操作が可能となる。拡大中心を固定することの有効性を調査するために、拡大中心を固定した手法、拡大中心を固定しスクロール機能を追加した手法、拡大中心が固定されていない手法、通常のソフトウェアキーボードの4つの手法の比較評価を行った。その

結果、拡大中心を固定することによって、ユーザは安定して選択操作を行うことができ正確にターゲットキーを選択できることがわかった。

2. 入力支援インタフェース

2.1 共通機能

図 1 のように、指を画面に近づけ、端末が指を検出したとき、指の位置に応じてソフトウェアキーボードの一部を拡大する。指を画面から引き、指が検出されないとき、ソフトウェアキーボードは通常表示になる。スマートフォンのソフトウェアキーボードのキーは縦長になっており、キーの横幅が指の幅より短いため、ユーザはターゲットキーと横方向に隣接するキーでよくミスタイプをする。そのため、3つの拡大手法では横方向にのみソフトウェアキーボードの一部を拡大し、キーの横幅を指の幅より長くするために拡大倍率を2倍に設定した。ソフトウェアキーボードが拡大される拡大領域は、幅は拡大されたキー2つ分であり、縦はソフトウェアキーボード全体である。領域内のソフトウェアキーボードが拡大されているとき、全てのキーを見えるようにするため、それ以外の領域は縮ませる。指を検出している間、指を左右に動かすことによって、拡大表示されるキーを変更することができる。



図 1: ソフトウェアキーボードの一部を拡大
Fig. 1 Part of the software keyboard is enlarged.

2.2 拡大中心

図 2 のように、端末が指を近づけたとき、拡大中心の位置から左右にキーが拡大される。拡大中心を固定する手法は、拡大中心の位置が移動しないため、拡大中心の位置に応じてキーの位置が決まる。拡大中心を固定しない手法は、指直下に拡大中心が位置するため、指の位置に応じてキーの位置が決まる。

2.3 NonFixedBoard

NonFixedBoard は、拡大中心を固定しない手法である。拡大中心は関連研究 [7] のように指直下に設定される。拡大中心が指と一緒に動くことで、キーは指の動きと反対方

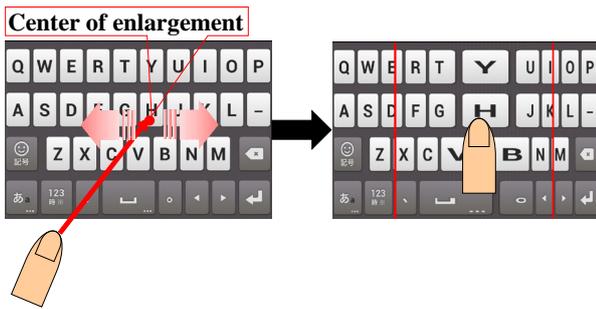


図 2: 拡大中心

Fig. 2 Center of enlargement.

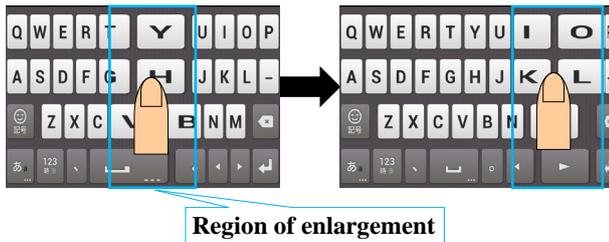


図 3: NonFixedBoard - 拡大中心を固定しない手法

Fig. 3 NonFixedBoard - not fixing the center of enlargement.

向に移動するため、安定したキー選択ができない可能性がある。図 3 は、拡大中心が 'H' 周辺から 'K' と 'L' の間に移動したことを示す。

2.4 FixedBoard

FixedBoard は、拡大中心を固定する手法である。拡大中心は最初に指を検出した位置に設定される。ソフトウェアキーボード上の 2 本の赤い線は拡大表示できる範囲を示している。図 4 のように、指を右端に移動させることによって、'J' や 'K' が拡大される。そのため、ユーザは拡大表示できる範囲内のキーを指を動かすことによって選択することができる。拡大表示できる範囲内にターゲットキーがないとき、一度画面から指を引き、ターゲットキーに指を近づけることによりターゲットキーを選択することができる。

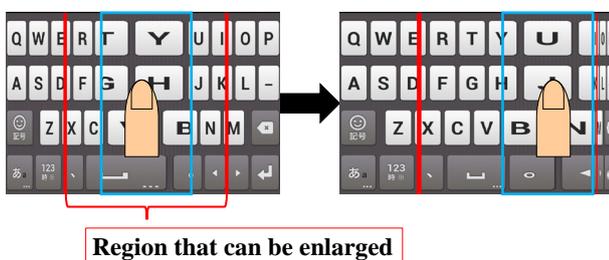


図 4: FixedBoard - 拡大中心を固定する手法

Fig. 4 FixedBoard - fixing the center of enlargement.

2.5 ScrollBoard

ScrollBoard は、スワイプ操作とフリック操作を Fixed-

Board に追加した手法である。スワイプ操作やフリック操作を行うことによって拡大中心を変更することができる。ターゲットキーが拡大できる領域内にないとき、スワイプ操作やフリック操作を行うことにより領域内に、ターゲットキーを移動させることができる。スワイプ操作では、拡大中心を指の動きと反対方向に指と同じ移動量だけ動かすことができる。図 5(a) は左方向にスワイプ操作を行っており、拡大領域も指と一緒に移動していることを示している。フリック操作では、拡大中心を指の動きと同じ方向に一定量動かすことができる。そのため、図 5(b) のように、ユーザは速くに位置するターゲットキーをすばやく選択することができる。

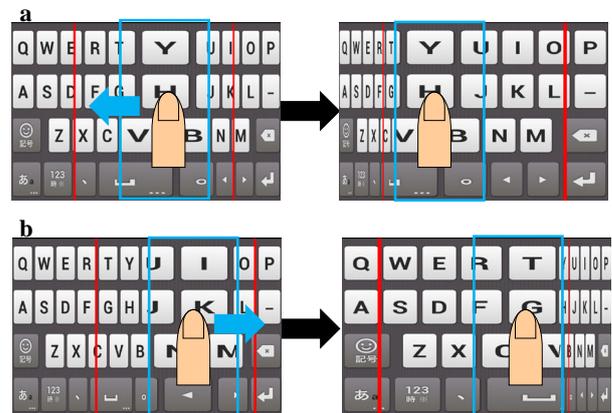


図 5: ScrollBoard - (a) スワイプ操作 (b) フリック操作

Fig. 5 ScrollBoard - (a)Swipe operation. (b)Flick operation.

3. 評価実験

4 つの手法を比較するためにテキストコピータスクを用いて被験者実験を行った。比較する手法として、前節で示した 3 つの拡大手法と通常のソフトウェアキーボードを用いた。

3.1 装置

手法は、全て Android アプリケーションとして実装した。Samsung Galaxy S4 スマートフォンを使用した。スマートフォンは 1920 × 1080 ピクセルのタッチスクリーンディスプレイを有する。モバイル端末のホバー機能を使用することによって、タッチスクリーン上方約 2 cm 以内の指の位置を取得することができる。キーボードは、Google 日本語入力アプリケーションの QWERTY 配列のソフトウェアキーボードを使用した。図 6 のように、入力フレーズとユーザが入力したフレーズはキーボード領域の上部に表示される。入力フレーズは、MacKenzie & Soukoreff のフレーズセット [11] から事前に選択した 30 フレーズ中からランダムに表示される。参加者がタイピングに集中できるように、全てのフレーズは句読点と記号を含まず、小文

字のみで構成される。

Phrase entered by a user

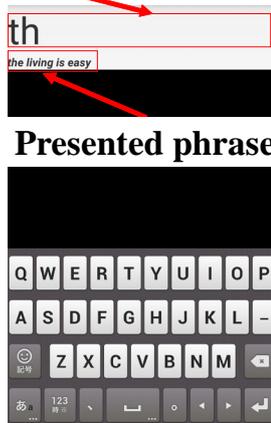


図 6: 実験時の画面

Fig. 6 Appearances on the screen in the user study.

3.2 設計

実験は 4 (手法) × 3 (ブロック) の被験者内計画である。独立変数は、手法とブロックである。手法による手順につりあいを持たせるために、ラテン方格を使用した [12]。入力速度として word per minute (WPM)、エラー率として total error rate (TER) を使用した。WPM は、1 単語をスペースを含む連続した 5 文字とした。TER は、参加者が転写中に修正したミスタイプと修正しなかったミスタイプをあわせたものをミスタイプ数とした [13]。入力時間は、1 文字目が入力されたときから最後の文字が入力されたときまでの時間とした。

3.3 参加者

参加者は 12 名 (全て男性)、年齢は 21-25 歳 (平均: 22.6 歳、標準偏差: 0.95) である。参加者は、全員英語のネイティブスピーカーではない。参加者は、全員定期的に PC のキーボードを使用しており、タッチパネルを搭載したスマートフォンを所持していた。11 名が右利き、1 名が左利きであった。

3.4 手順

実験中、参加者は椅子に座った状態で入力を行った。実験は、参加者 1 人あたり約 1 時間行った。初めに、各参加者に対して、簡単に実験の目的を述べた。各参加者は、1 つの手法につき 3 ブロック行う。1 ブロックは、5 フレーズで構成される。各参加者に対して、4 つの手法を短いデモンストレーションを通して説明した。また、参加者に対して以下のように指示した。

(1) 非利き手で端末を持ち、利き手の人差し指で入力を行うこと

- (2) 非利き手の指で端末を覆わないように持つこと
- (3) 入力フレーズを覚えてからタイピングを始めること
- (4) ミスタイプは修正すること

ただし、2 回以上連続で backspace キーを入力する必要がある場合は無視して、入力続けること

- (5) できるだけ速くかつ正確に入力を行うこと

また、ScrollBoard と FixedBoard を区別するために、ScrollBoard を使用するときには、指を引いて拡大中心を変更する操作を禁止した。各ブロックを始める前に、1 分間練習時間を設けた。また、参加者が疲労を感じたときは、ブロック間で最大 3 分間休憩を取ることができるようにした。実験終了後、参加者に対してアンケートを実施した。

3.5 結果

ミスタイプによるエラーのみを計測するために、単語の入力し忘れやフレーズと明らかに異なる単語が入力されたものは取り除いてエラー率を算出した。図 7 は、テキストコピータスクの入力速度とエラー率を示したものである。入力速度に関して、ANOVA による分析の結果、手法 ($F(3, 33) = 93.89, p < .001$) とブロック ($F(3, 33) = 12.38, p < .001$) に有意差があり、交互作用 ($F(6, 66) = .369, p > .05$) には有意差はなかった。Holm-Bonferroni 法による事後比較の結果、全ての手法間に有意差があった。図 7 から通常のソフトウェアキーボードが最も速く、続いて NonFixedBoard, ScrollBoard, FixedBoard の順であった。

また、エラー率に関しては、ANOVA による分析の結果、手法 ($F(3, 33) = 6.173, p < .001$) に有意差があったが、ブロック ($F(3, 33) = 2.602, p > .05$) と交互作用 ($F(3, 33) = .644, p > .05$) には有意差はなかった。Holm-Bonferroni 法による事後比較の結果、通常のソフトウェアキーボードと FixedBoard, 通常のソフトウェアキーボードと ScrollBoard, NonFixedBoard と ScrollBoard の間に有意差があった。図 7 から ScrollBoard は通常のソフトウェアキーボードと NonFixedBoard より低いエラー率であった。また、FixedBoard は通常のソフトウェアキーボードより低いエラー率であった。

4. 考察

4.1 入力速度

NonFixedBoard は通常のソフトウェアキーボードより入力速度が遅かった。NonFixedBoard は、ユーザに拡大中心を変更する操作を強いることがないため、通常のソフトウェアキーボードと同様に操作することができる。しかし、拡大中心が固定されていないためユーザは安定してキーを選択することができない。そのため、ユーザは慎重にターゲットキーを選択する必要がある、通常のソフトウェアキーボードより遅くなったと考えられる。ScrollBoard は

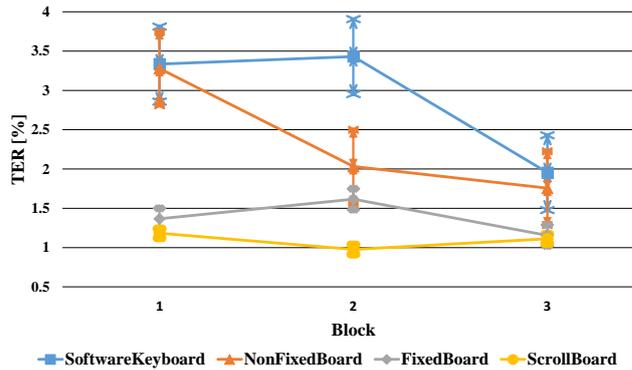
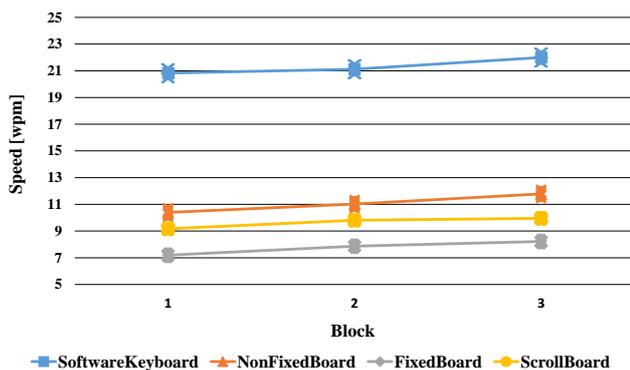


図 7: 各手法における入力速度とエラー率 (エラーバーは標準誤差を示す)

Fig. 7 Text entry speed and error rate in each method. (Error bars show standard errors.)

NonFixedBoard より入力速度が遅かった。これは、スワイプ操作やフリック操作により拡大中心を変更する必要があったためである。また、ソフトウェアキーボード上でスワイプ操作やフリック操作を行うことがユーザになじみがなかったためと考えられる。ScrollBoard は FixedBoard より入力速度が速かった。この結果から、拡大中心を変更するとき、画面から指を引くよりもスワイプ操作やフリック操作を行ったほうがすばやくキーを選択できることが示された。

4.2 エラー率

FixedBoard と ScrollBoard のエラー率が低いことから、拡大中心を固定することによりユーザは安定してキーを選択することができたといえる。また、ソフトウェアキーボードの一部を拡大することにより、ターゲットキーと横方向に隣接するキーでのミスタイプを減らすことを可能にした。拡大中心を固定しないことは、ミスタイプ数の増加を招いた。

5. まとめ

本稿では、モバイル端末のホバー機能を用いて指直下周辺のソフトウェアキーボードの一部を拡大することで、文字入力におけるミスタイプを減らすことができる入力支援インタフェースを提案した。実験結果から、ソフトウェアキーボードの一部を拡大し、拡大中心を固定することによりミスタイプが減少した。これにより、拡大中心を固定することによりユーザは安定して選択操作を行うことができ、正確にターゲットキーを入力することができることが示された。今回提案したホバー機能を使用して拡大する技術は、視力が悪い人や慎重に選択操作を行う必要があるアプリケーションに対して有益な技術であると考えられる。

今後の課題として、入力速度を改善するために拡大倍率や拡大領域を変更して、拡大中心を固定した手法のパラメータを最適化することが挙げられる。

参考文献

- [1] X. A. Chen, T. Grossman, and G. Fitzmaurice. 2014. Swipeboard: a text entry technique for ultra-small interfaces that supports novice to expert transitions, Proc. UIST 2014, 615-620.
- [2] S. Oney, C. Harrison, A. Ogan, and J. Wiese. 2013. ZoomBoard: a diminutive QWERTY soft keyboard using iterative zooming for ultra-small devices, Proc. CHI 2013, 2799-2802.
- [3] L. A. Leiva, A. Sahami, A. Catala, N. Henze, and A. Schmidt. 2015. Text Entry on Tiny QWERTY Soft Keyboards, Proc. CHI 2015, 669-678.
- [4] D. Vogel and P. Baudisch. 2007. Shift: a technique for operating pen-based interfaces using touch, Proc. CHI 2007, 657-666.
- [5] A. Gupta and R. Balakrishnan. 2016. DualKey: miniature screen text entry via finger identification, Proc. CHI 2016, 59-70.
- [6] Y. F. Shao, M. C. Ogimoto, R. Pointner, Y. C. Lin, C. T. Wu, and M. Chen. 2016. SwipeKey: a swipe-based keyboard design for smartwatches, Proc. MobileHCI 2016, 60-71.
- [7] F. Pollmann, D. Wenig, and R. Malaka. 2014. HoverZoom: making on-screen keyboards more accessible, Proc. CHI 2014, 1261-1266.
- [8] N. Koarai and T. Komuro. 2013. A zooming interface for accurate text input on mobile devices, Proc. CHI 2013, 1299-1304.
- [9] X. Yang, T. Grossman, P. Irani, and G. Fitzmaurice. 2011. TouchCuts and TouchZoom: enhanced target selection for touch displays using finger proximity sensing, Proc. CHI 2011, 2585-2594.
- [10] A. Ostberg and N. Matic. 2015. Hover Cursor: Improving Touchscreen Acquisition Of Small Targets With Hover-enabled Pre-selection, Proc. CHI 2015, 1723-1728.
- [11] I. S. MacKenzie and R. W. Soukoreff. 2003. Phrase sets for evaluating text entry techniques, Proc. CHI 2003, 2754-2755.
- [12] J. V. Bradley. 1958. Complete counterbalancing of immediate sequential effects in a Latin square design J. Amer. Statist. Ass. 53, 525-528.
- [13] R. W. Soukoreff and I. S. MacKenzie. Recent developments in text-entry error rate measurement. Proc. CHI 2004, 1425-1428.