DStyleKeyboard：キーが動的に変化するソフトウェアキーボードの開発

LI MIAO* 高橋 伸†

概要. 本研究では入力効率を向上させるソフトウェアキーボードの設計を検討する. 標準的な QWERTY 配列のキー配置をもとに, キーのレイアウトとキーのスタイルを動的に変化させる新しいソフトウェアキーボードを提案する. 提案手法では, 入力が予想されるキーのスタイルを変更し, それに応じて他のキーのスタイルを調整する. さらに, 拡大するキーの数や拡大率のキー入力速度への影響を調査するために, 予備実験を行った.

1 はじめに

物理的なキーボードはデスクトップ環境では非常に適した構成で，両手で操作するにはとても効率的で操作が行いやすい。しかし、モバイルデバイスなどの小さい画面では，キーが密集した QWERTY 配列キーボードは,「ファットフィンガー」[4] の問題がある．同時に，Gunawardana らの研究 [2] によれば，ソフトキーボードには触覚フィードバックがないため，ユーザがキーに触れたり、クリックしたり、キーから離れたりしたことを知ることができない。キーの位置を繰り返し確認しなければ，入力効率が低くなる可能性がある。

そこで本研究では，入力効率を向上させるソフトウェアキーボードの設計を検討し，標準的な QWERTY 配列のキー配置を基に，キーのスタイルを動的に変化させる新しいソフトウェアキーボードを提案する．このキーボードは，QWERTY レイアウトを保ちつつ，ユーザが入力したいキーを予測し，キーのスタイル（色やサイズなど）を動的に変化させる．これにより，ユーザがキーの位置を確認しやすくなる．また「ファットフィンガー」[4] の問題が軽減することで入力効率が向上すると期待される．

2 関連研究

スマートフォンで入力効率を向上させるために，さまざまな方法が提案されている．Bhatti ら [1] は，スマートフォンの QWERTY 配列キーボードに基づく入力手法を提案した．このキーボードでは，ユーザがキーをクリックすると，次に入力する可能性が高いアルファベットを予測し，そのアルファベットに対応するキーの幅とその枠の色を変更する．具体的な方法としては，キーの幅を標準のキーの 2 倍に拡大することである．この方法の入力速度は，3 週間の使用で平均 8.7%向上した．しかし，この手法ではキーの高さとキー全体の色を変更することは考えられていなかった．また，標準的な QWERTY 配列キーボードとの比較はされていなかった．Rodrigues ら [3] は，タブレット端末上での QWERTY 配列キーボードに基づく入力手法を提案した．提案手法としては，次の 4 つの最も可能性の高いキーの幅や色を変えることである．この手法による入力速度は，標準的な QWERTY キーボードより低かったと報告された．しかし，この手法ではキーのサイズと色を両方変更することは考えられていなかった．また，結論はタブレットを用いた場合の結論であるため，スマートフォン上では違う結論が出る可能性がある．

そこで，本研究ではスマートフォン上で，キーの幅を変更することに限らず，キーのサイズ（幅と高さ）と色の両方が動的に変更されるキーボードを提案する．

3 提案手法

3.1 キーのスタイルの調整方法

キーのスタイルを調整する方法としては，図 1 に示すように，予測されたキーのサイズおよび色を変更する。予測されたキーのサイズを拡大するために，他のキーのサイズは縮小して調整される．

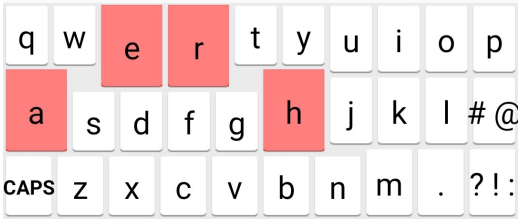

図 1. キーのサイズと色の変更

* 筑波大学　情報理工学位プログラム
† 筑波大学　システム情報系

3.2 キーのサイズの調整方法

アルゴリズム 1 はキーを拡大しながら, キーの位置を調整するアルゴリズムである.

Algorithm 1 ScaleKey

Require: K: 拡大するキーの集合, w: 拡大する幅, h: 拡大する高さ

function ScaleKey(K, w, h)
  for each key $k \in K$ **do**
    $K' \leftarrow k$ と同じ行のキー
    for each key $k' \in K'$ **do**
      if $k' \neq k$ **then**
        k' の幅を $w/(|K'|-1)$ 減らす
    k の幅を w 足す
    MoveCol(k, K')
    $K' \leftarrow k$ と同じ列のキー
    for each key $k' \in K'$ **do**
      if $k' \neq k$ **then**
        k' の高さを $h/2$ 減らす
    k の高さを w 足す
    MoveRow(k, K')
function MoveCol(k, K)
  for each key $k' \in K$ **do**
    if k' は k の左にある $\vee\, k' = k$ **then**
      k' は左の何かと隣接するまで左に移す
    else
      k' は右の何かと隣接するまで右に移す
function MoveRow(k, K)
  for each key $k' \in K$ **do**
    if k' は k の下にある $\vee\, k' = k$ **then**
      k' は上の何かと隣接するまで上に移す
    else
      k' は下の何かと隣接するまで下に移す

拡大しないキーの幅と高さを削り，拡大するキーに削った分を分配する．その上で，キーのサイズを調整することにより，他のキーまたはキーボードのボーダーと被らないように，各キーの位置を調整する（MoveCol と MoveRow を用いる）．1 つのキーを拡大する場合，拡大したキーを固定し，各キーの位置を調整する．複数のキーを同時に拡大する場合，まず，1 つのキーを拡大し，各キーの位置を調整し，そして次に 2 つ目のキーを拡大し，各キーの位置を調整する．ただし、ScaleKey でキーのサイズと位置情報を調整する途中では、キーボードのレンダリングは行わない．キーボードのレンダリングは ScaleKey が終わってから行う．

4 予備実験

提案手法において，キーのサイズが入力効率に与える影響を調査するために，予備実験を行った．本実験では，拡大するキーの数を 4 個に固定した．また，アルファベット予測が必ずユーザが入力したいアルファベットを含んで予測できるように実装した．実験のタスクとしてはスマートフォンの画面に表示された文章を入力することである．ソフトウェアキーボードのキーは，N,S,M,L の 4 つのキーのサイズ拡大方法を用意した．

今回の実験では参加者が 9 名であった．参加者は，それぞれのサイズに拡大する方法を使用し，ソフトウェアキーボードに email から抽出した短い文章 [5] を 16 個を選んだを入力した．

評価対象は，入力速度（WPM），エラー率（Error Rates) である．予備実験の結果としては，提案したキーボード（S, M, L）は標準的な QWERTY キーボード（N）と比べて入力速度（WPM）に対して有意差が検出されなかった．これは入力速度の高いキーのサイズが参加者によって違いがあったためと考えられる．

表 1. 平均入力速度

Size	N	S	M	L
WPM	21.29	23.04	22.03	21.23

平均入力速度結果（表 1）により，S サイズの平均入力速度が最も高く，L サイズの平均入力速度が低かった．つまり，キーのサイズをある程度大きくすると入力速度が上がるが，キーのサイズが大きすぎると入力速度が下がる傾向が見られた．その理由は，キーが大きすぎるとキーの位置が変わってしまい，参加者はキーを探し続けなければならないので，入力時間が長くなってしまうからだと推測される．また，エラー率に対してもキーのサイズに関して有意差が検出されなかった．

5 おわりに

本研究では，キーのスタイルを動的に変化させる方法を提案し，それに基づく QWERTY 配列のキーボードを実装し，予備実験を行った．今後は、サイズの変更だけでなく、適切な色の変更方法を検討する。また，変更するキーのサイズと色の組み合わせにより，入力効率に与える影響を調査する予定である．

参考文献

[1] M. S. Bhatti, M. A. Ahmed, S. S. Kajiya, A. Qayum, I. Latif, A. K. Shahid, R. Qureshi, and S. I. Hashmi. Mistype resistant keyboard (NexKey). In *2018 13th Iberian Conference on*

Information Systems and Technologies (CISTI), pp. 1–7. IEEE, 2018.

[2] A. Gunawardana, T. Paek, and C. Meek. Usability guided key-target resizing for soft keyboards. In *Proceedings of the 15th international conference on Intelligent user interfaces*, pp. 111–118, 2010.

[3] E. Rodrigues, M. Carreira, and D. Gonçalves. Improving text entry performance on tablet devices. *Interação*, 2013.

[4] K. A. Siek, Y. Rogers, and K. H. Connelly. Fat finger worries: how older and younger users physically interact with PDAs. In *IFIP Conference on Human-Computer Interaction*, pp. 267–280. Springer, 2005.

[5] K. Vertanen and P. O. Kristensson. A versatile dataset for text entry evaluations based on genuine mobile emails. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*, pp. 295–298, 2011.