近傍文字情報を用いた単語絞り込みを利用した視線スワイプによるテキスト入力法の改良

藤井 寿紀[1,a)]　勝間 亮[1,b)]

概要：視線ベースのテキスト入力は，手が汚れた状態，手の怪我，身体障碍などでキー入力デバイスが扱いにくいときの重要なコミュニケーションツールとなる．これまでに，画面上に表示された QWERTY 配列のキーボードに対して視線を用いてテキスト入力する方法が提案されている．その中でも視線スワイプはユーザが入力したい単語の文字列を順に視線で追跡することで単語を予測する入力手法であり，文字入力の高速化が期待されている．視線スワイプは単語予測の高精度化と計算量の低減の課題があるが，先行研究の EyeSwipe では最初と最後の文字を特殊な視線ジェスチャで確定させることで，実用に耐える計算量にし，単語予測を高精度化している．しかし，この視線ジェスチャは時間のかかる操作である．我々は視線先の近傍文字情報を基に単語を絞り込み，単語の最後の文字に対する特殊な視線アクションを用いずに高精度化および計算量の低減を行う手法を提案している．この提案手法を実装したアプリケーションを展示する．

1. 導入

現在コンピュータを利用する際，キーボードが文字を入力する方法として主流となっている．またスマートフォンにおいては，人によってさまざまではあるが，タッチスクリーンキーボードの他にトグル入力やフリック入力が一般的である．これらの入力方法は手を使って入力するものであり，手が汚れた状態，手の怪我，身体障碍などでキー入力デバイスを扱いにくい場合に適した入力方法ではない．手を使わない入力方法である視線入力はより様々な状況において，コンピュータを介した他者とのコミュニケーションを促進させる効果が期待できる．

視線入力とは，アイトラッカ等の機器を用いて取得したユーザの視線情報を入力信号にするものである．一般的な視線入力を用いた文字入力は，スクリーンキーボードを配置し，そのキーボード上の入力したい文字を注視することで達成される．この入力手法は，安定した入力を可能にするが，文字入力に時間がかかるデメリットがある．これは，注視したことを確認するために，1 文字の入力ごとに一定時間を要するためである．以降，この一定時間を滞留時間と呼び，滞留時間を用いたキーボード文字入力を滞留タイピングと呼ぶこととする．滞留タイピングでは，滞留時間をある程度長く設定しておく必要がある．滞留時間を短くすればするほど，意図せず文字を入力する可能性が上がるからである．そのような性質から，滞留タイピングでは入力速度に限界があると考えられる．テキスト入力を円滑にするためには，誤認が少なく短時間で入力できる他の手法の確立が必要である．

高速に入力できる視線ベースの文字入力手法として，視線スワイプが期待されている．視線スワイプとは，ユーザが入力したい単語の文字列を順に視線で追跡することで，その視線経路とあらかじめ用意された辞書上の単語の理想視線経路とを照合して，入力された単語を予測する入力方法である．この方法は従来の滞留時間を用いた文字入力のように，1 文字ずつ視線を停止させながらを入力する必要はない．そのため EyeSwipe は短時間での入力が期待される．そして，必ずしも各文字上に厳密に視線を動かす必要はないため，ノイズの大きい視線入力に対して堅牢である．しかしながら，入力された視線経路と辞書に含まれる単語の理想視線経路との照合において,，すべての単語に対して類似度を計算すると計算時間が実用に耐えられないため，計算量を減らす必要がある．Andrew ら [1] は，視線スワイプの先行研究として EyeSwipe を提案している．EyeSwipe の貢献は主に 2 つある．1 つ目は単語の最初の文字と最後の文字を，特殊な視線ジェスチャをユーザに行わせることによって確実に取得し，最初の文字と最後の文字が一致し

[1] 大阪公立大学
Osaka Metropolitan University, Sakai, Osaka 599-8531, Japan
[a)] sb22651s@st.omu.ac.jp
[b)] katsuma@omu.ac.jp

ている単語に対してのみ類似度計算を行うことで，実用レベルにまで計算量を減らし，高精度になることを示したことである．2 つ目は，視線ジェスチャを単語入力のトリガとすることは，ユーザーが間違えて単語入力を開始させることを防止する効果があることを示したことである．彼らによる EyeSwipe と滞留タイピングと比較実験では，一分間での入力単語数（wpm: words per minute）はそれぞれ 11.7 wpm，9.5 wpm となり，入力の高速化が実現できたことが示されている [1].

Kristensson らの研究 [2] は，視線スワイプの可能性を示している．この研究では，単語入力が単語の文字に対応したスクリーンキーボード上の文字近傍を見ることで完了する，完璧なインターフェースを仮定し，そのインターフェースを用いてシミュレーションを行った．40 分の練習後，被験者の平均入力速度は 46 wpm に達し，EyeSwipe に改善の余地がある可能性が高いことが示された．

EyeSwipe の入力速度が低くなっている要因の 1 つとして，単語の最初の文字と最後の文字を確定させるために必要な特殊な視線ジェスチャに時間がかかっていることが挙げられる．そこで我々は単語の最後の文字を確定させる代わりに，視線入力の近傍文字情報を基に単語の候補を絞り込むことを提案している．この手法では，単語の最後の文字を確定させるための視線ジェスチャを行う必要がないため，キーボードから視線を外す等の短時間でできる操作でも予測精度を落とすことなく入力することが可能となる．また視線入力の近傍文字によって絞り込むため，もし途中で意図しない文字に視線を移したとしても，一度は目的の文字に視線を移動させるため，目的の単語が候補から除外される可能性は低く，頑健性が高い．

2. 提案手法

2.1 前提条件

まずスクリーン上に英字アルファベット 26 文字が各 1 文字ずつ存在するキーボードが配置され，ユーザは英字アルファベットのみで構成される単語を入力すると仮定する．ユーザは単語を入力するために，スクリーン上で視線を動かし，その視線の動きはアイトラッカにより計測されているとする．単語入力は単語の最初の文字を一定の操作により決定することで開始される．一定の操作の例としては，EyeSwipe で採用されている視線ジェスチャや滞留タイピング等がある．その後，ユーザは単語を構成する各文字を視線でなぞる．単語の最後の文字までなぞり終えた後，最初の文字と同様，一定の操作により単語入力を終了させる．入力終了後，即座にシステムが入力された単語を推測し，可能性が高い順に単語を列挙して表示する．ユーザはその列挙された単語を注視することで単語の入力を決定する．即座にシステムが入力された単語を推測し，可能性が高い順に単語を列挙して表示する．ユーザはその列挙された単語を注視することで単語の入力を決定する．

2.2 単語の候補の絞り込み

最初に，単語の候補を絞り込むために必要なデータとその定義について説明する．アイトラッカはサンプリングレート f Hz でスクリーン上の視点の座標が取得され，その座標の時系列データを視線経路と呼ぶこととする．視線経路の生データを $\{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_t, ..., x_{\mathrm{t_{end}}}\}$ とおく．$\boldsymbol{x}_t$ は時刻 t における視線入力である．視線経路の生データをノイズが多い視線入力を処理しやすくために，平均フィルタリングする．平均フィルタリングの具体的な計算は以下のようになる．

$$\boldsymbol{x}'_t = \begin{cases} \boldsymbol{x}_t & t = 1, t_{\mathrm{end}} \\ \frac{\boldsymbol{x}_{t-1}+\boldsymbol{x}_t+\boldsymbol{x}_{t+1}}{3} & o.w \end{cases} \tag{1}$$

スクリーンキーボード上にある各文字の座標を $\{\boldsymbol{\lambda}_{\mathrm{a}}, \boldsymbol{\lambda}_{\mathrm{b}}, ..., \boldsymbol{\lambda}_{\mathrm{z}}\}$ とする．また一定の操作により選択された単語の最初と文字を c_{start} とする．視線スワイプでは，辞書の中から入力された可能性が高い単語が選択される．可能性の高さの基準は，視線入力と単語との類似度により決定される．類似度には DTW（Dynamic Time Warping）を使用している．DTW とは時系列データ同士の類似度を計算する手法である．視線スワイプにおける DTW の一方のデータは平均フィルタリングされた視線経路であり，もう一方のデータは単語を構成する各文字の中心座標を順に並べたもの（以降，理想経路と呼ぶ）である．例えば「nice」の理想経路は $\{\boldsymbol{\lambda}_{\mathrm{n}}, \boldsymbol{\lambda}_{\mathrm{i}}, \boldsymbol{\lambda}_{\mathrm{c}}, \boldsymbol{\lambda}_{\mathrm{e}}\}$ である．辞書に含まれる単語の集合を D とする．そして，D の任意の単語に対して先頭の文字から一部切り出した文字列の集合を D_{sub} とする．例えば，$D = \{\mathrm{word}, \mathrm{eye}\}$ で，$D_{\mathrm{sub}} = \{\mathrm{w}, \mathrm{wo}, \mathrm{wor}, \mathrm{word}, \mathrm{e}, \mathrm{ey}, \mathrm{eye}\}$ となる．

次に視線経路の近傍文字の取得について説明する．ある時刻 t において，視線座標 $\boldsymbol{x_t}$ が得られたとき，$\boldsymbol{x_t}$ と各文字の座標との距離を求め，ソートする．そして，その距離の中で，距離が短い順に k 個の要素を取得し，それらに対応する文字 $C_t = \{c_{1,t}, c_{2,t}, ..., c_{k,t}\}$ を得る．

候補とする単語であるための条件について説明する．以降，提案手法の条件に合う単語を候補単語と呼ぶこととする．ある時刻 t における採用する文字を $c'_t \in (C_t \cup \mathrm{NULL})$ とする．ここで NULL とは空文字列を表している．時刻 t における候補単語 w_t は以下の条件を満たすとする．

$$w_t \ = \ c_{\mathrm{start}} + c'_1 + c'_2 + ... + c'_t \tag{2}$$

(2) 式における $+$ 演算子は文字列の結合を表している．これに加えて，候補単語 w_t は辞書に含まれる必要がある．これらの条件を満たす候補単語を高速に抽出するために，まず辞書 D をトライ木で実装する．トライ木とは文字列を高速に検索することを可能にするデータ構造である．(2)

式を満たす単語のうち，D_{sub} に含まれる辞書を $W_{\mathrm{sub}}(t)$ とし，$W_{\mathrm{sub}}(t)$ もトライ木で実装する．実際の入力では辞書 $W_{\mathrm{sub}}(t)$ を更新し，$W_{\mathrm{sub}}(t)$ から得られるすべての単語の類似度を計算する．

簡単な具体例を図 1 と図 2 を用いて示す．

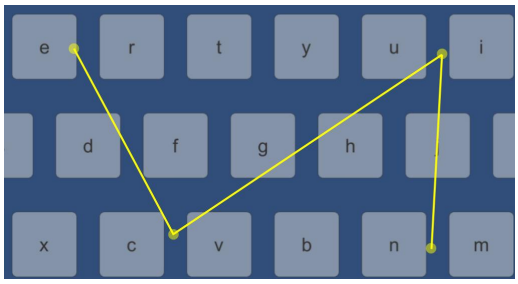

図 1 単語「nice」の入力視線経路の例

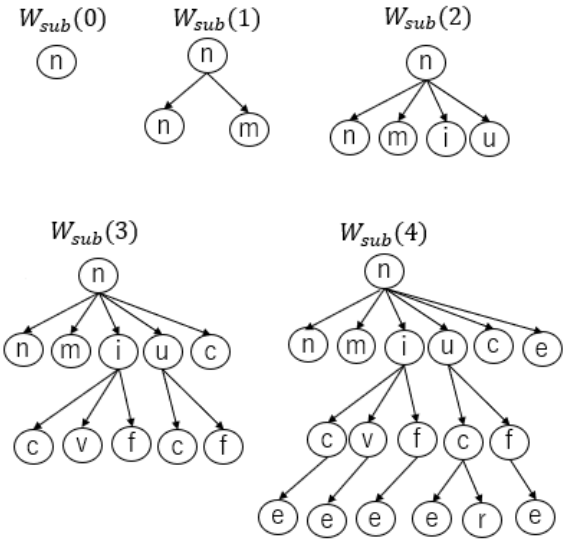

図 2 $W_{\mathrm{sub}}(t)$ の更新

図 1 の黄線は単語「nice」を入力したときの視線経路を表し，視線の軌跡が単語に含まれるキー上を通過していない場合の例を示す．図 1 を基に $W_{\mathrm{sub}}(t)$ が更新する様子を表したものが図 2 である．ただし，$k = 3$ で最初の文字を $c_{\mathrm{start}} = \mathrm{'n'}$ としている．図 2 が示すように，$W_{\mathrm{sub}}(t)$ は視線経路の近傍を通った文字列のみに限定されている．こうして得られた $W_{\mathrm{sub}}(t)$ から候補単語を構成する．その後，候補単語に対して類似度を計算し，類似度の高い単語がユーザの希望する単語としてが表示される．図 2 より，単語「nice」が含まれていることが分かる．

3. まとめ

本稿では，視線スワイプの先行研究である EyeSwipe を改善し，単語の最後の文字を確定させる代わりに，視線先の近傍文字情報を基に単語を絞り込み，高精度化する手法を提案している．この提案手法を実装したアプリケーションを展示する．会場にアイトラッカの装着したモニタとコンピュータを搬入し，実際に視線スワイプでの入力と単語予測を体験していただく．

参考文献

[1] Andrew Kurauchi, Wenxin Feng, Ajjen Joshi, Carlos Morimoto, Margrit Betke: ”EyeSwipe: Dwell-free Text Entry Using Gaze Paths”, *CHI ’16: Proceedings of the 2016 CHI Conference on Human Factors in Computing SystemsMay 2016*, pp.1952–1956 (2016)

[2] Per Ola Kristensson, Keith Vertaneny: ”The potential of dwell-free eye-typing for fast assistive gaze communication” *ETRA ’12: Proceedings of the Symposium on Eye Tracking Research and ApplicationsMarch 2012*, pp. 241–244 (2012)