

# 視線ベースの一筆書き文字入力におけるリアルタイムな単語絞り込みによる高精度化の検討

## Examination of Improving Accuracy for Gaze-Based Single-Stroke Text Input by Real-Time Word Refinement

藤井 寿紀<sup>1</sup>勝間 亮<sup>1</sup>

Hisanori Fujii

Ryo Katsuma

### 概要

視線ベースの文字入力は、身体障害を持つ人にとって重要なコミュニケーションツールとなる。文字を一つずつ入力する方法では視線を文字ごとに停止させる必要があり、入力の遅さが問題となる。先行研究で提案された EyeSwype ではユーザが入力したい単語を構成するすべての文字を視線で追跡することで単語を予測する入力手法であり、最初と最後の文字以外では視線の停止を必要としないため、文字入力の高速化が期待される。しかしながら、EyeSwype は誤って単語を予測する可能性があり、予測の高精度化は必須である。本稿では、予測の高精度化のために、視線先の近傍文字情報を基に、候補とする単語をリアルタイムに絞り込む手法を提案する。

### 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一般的なものである。これらの入力方法は手を使って入力するものであり、身体障害を持つ人にとって適した文字入力方法ではない場合がある。手を使わない入力方法である視線入力は身体障害を持つ人のコンピュータの利用を、さらには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ユーザの視線を取得し、視線情報から得られた視線先の位置をスクリーン上の座標とする入力である。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、滞留時間が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。身体障害を持つ人のコミュニケーション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。

視線ベースの文字入力手法の先行研究として EyeSwype がある [1]。EyeSwype とは、ユーザが入力したい単語を構成するすべての文字を視線で追跡することでその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 EyeSwype は短時間での入力が期待される。EyeSwype を提案した論文によると、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ことが示されている。また EyeSwype と滞留タイピングと比較実験したところ、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 となったことが示されている [1]。

しかしながら、我々は EyeSwype に3つの課題が存在すると考える。まず1つ目は、文字の配置を完全に覚えている必要があることである。ユーザは文字の配置をひとつでも忘れて、文字入力中に文字の位置を探す行為をされると考えられる。このとき、目的の文字以外を誤って見ることによって、単語の予測精度は著しく下がる。2つ目は、視線を素早く動かす必要があることである。EyeSwype の単語予測方式では、他の文字と比べ、単語に含まれる文字上に視線が長時間、位置していることが重要視される。視線をゆっくりと移動させると、単語に含まれない文字上に視線が長時間、位置することとなり、意図しない単語が予測される可能性が高くなる。もしくは、これを避けるために、素早く視線を目的の文字に送ろうとして失敗する可能性もある。どちらにせよ、素早く入力する必要があることは単語の予測精度を下げる要因であると考えられる。3つ目は、探索する辞書の単語数によ

<sup>1</sup> 大阪公立大学, Osaka Prefecture University, Sakai, Osaka 599-8531, Japan

り、単語予測の計算量と精度が大きく異なることである。EyeSwype では視線入力を基に、辞書に含まれる単語の中から最も適したものを取得することで希望する単語を入力する。すなわち、辞書の単語数が多ければ多いほど、計算に時間がかかる上に、予測精度は下がると考えられる。

そこで本稿では、3つ目の課題に焦点を当て、その課題を解決する手法として、視線入力の際の近傍文字情報を基に単語の候補を大幅に絞り込むことを提案する。候補を絞り込むことは、希望しない単語が候補となる可能性を減らし、結果として希望する単語の予測精度を向上させることにつながる。加えて、計算する必要がある単語数を減らすことができる。また、視線入力の際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、目的の文字が候補から除外される可能性は低い。

## 2. EyeSwype の概要

滞留タイピングのデメリットを考慮し、滞留時間を使わない視線ベースの文字入力の先行研究として、EyeSwype が提案されている [1]。EyeSwype の入力の様子を模式的に表したものを図 1 に示す。

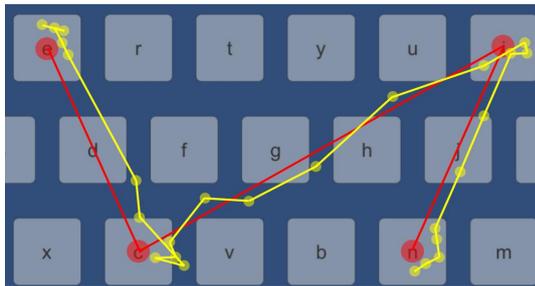


図 1: EyeSwype の入力の様子。赤線は「nice」の理想経路、黄線は視線経路を表している。

EyeSwype はスマートフォンの文字入力方法 Swype 入力を視線ベースの入力に適応させたものである。元となる Swype 入力はタッチスクリーンキーボード上で、希望する単語を構成する各文字を指でなぞることで入力する方法である。例えばユーザは「nice」と入力したければ、「n」「i」「c」「e」の近傍を順番になぞることになる。EyeSwype は Swype 入力におけるなぞる行為を視線を動かす行為に置き換えている。すなわち、ユーザは EyeSwype で「nice」と入力したければ、「n」「i」「c」「e」の近傍を順番に目で追跡することになる。EyeSwype により希望する単語を得るまでの大まかなプロセスを以下に示す。

- (a) 特徴的な視線ジェスチャにより最初の文字を決定し、単語の入力を開始する。
- (b) 単語の各文字を順番に視線で追跡しつつ、随時ユーザの視線経路と辞書に含まれる各単語の類似度を計算する。ただし、類似度を計算する単語は、最初と最後の文字を限定することで減らしている。
- (c) 特徴的な視線ジェスチャにより単語の入力を終了させ、類似度が一番高い単語を目的の単語とする。
- (d) (c) で表示された単語が希望する単語でない場合、スクリーンに表示された、類似度が降順となるように並べられた 5 つの単語から希望する単語を選択する。

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。EyeSwype では、一方のデータは視線経路であり、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。図 1 において、黄線は視線経路を表し、赤線は「nice」の理想経路を表している。なお、(b) で類似度の計算に使われる辞書はあらかじめ記憶領域に格納されている。

上記のプロセスを経て、ユーザは希望する単語を入力することが出来る。

## 3. 関連研究

有名な視線ベースの文字入力手法として Dasher がある [2]。Dasher は次々と現れる文字のボックスを移動していくことで文字を入力するインターフェースである。Dasher の評価実験を行った研究 [3] によると滞留タイピングが 6.0-7.0 wpm であったのに対して、Dasher は 12.6-14.2 wpm と約 2 倍ほど入力速度に差があったことが示されている。

Mott らの研究 [4] では滞留タイピングを改善した手法が提案されている。スクリーンキーボード上の文字が次に選択される可能性と、キーボード上の文字の位置に基づいて、文字ごとの滞留時間を動的に調整することで、滞留時間の最適化を測っている。静的な滞留時間による入力速度が平均して 10.62 wpm であるのに対して、彼らの手法は平均して 12.39 wpm となったことが示されている。

視線ジェスチャをベースとした文字を入力するアプローチがある。Wobbrock らの研究 [5] でキーボードを使わず、各アルファベットに対応した視線ジェスチャを用いて文字を入力する手法が提案されている。入力速度の比較実験を行ったところ、視線タイピングに劣ったものの、誤入力が少なくユーザが疲れにくいという利点、及びス

クリーン占有率が小さくて済むという利点があることが示されている。

他には、文字の入力を完了したことを表す視線領域をスクリーン上に作ることで、入力的高速化を図るアプローチがある。Bee と Andr'e[6] は、円周上に文字を配置し、円の中心に文字の入力を完了したことを表す視線領域を配置されたインターフェースを提案している。Morimoto と Amir[7] は二つのスクリーンキーボードを、文字の入力を完了したことを表す視線領域を挟むように配置されたインターフェースを提案している。この2つのキーボードを交互に入力する手法は、実験により入力速度は平均して 12 wpm であったことが示されている。

以上の様々な入力手法は、いずれも平均入力速度は 15 wpm 以下であり他の入力手法を凌駕するものではない。しかしながら、Kristensson と Vertanen[8] は、EyeSwype のような入力手法の可能性を示している。彼らは、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 に達したことが示され、EyeSwype に改善の余地がある可能性が示されている。平均入力速度は単語予測の失敗による入力のやり直しがあると遅くなるため、単語予測の精度を上げることで改善可能である。そこで本稿では EyeSwype の単語予測の高精度化を検討する。

## 4. 提案手法

### 4.1 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、EyeSwype 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。

### 4.2 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。時刻  $t$  までの視線経路を  $\{x_1, x_2, \dots, x_t\}$  とおく。なお、視線経路はノイズが多い視線入力を処理しやすくするために、平均フィルタリングされている。スクリーンキーボード上にある各文字の座標を  $\{\lambda_a, \lambda_b, \dots, \lambda_z\}$  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  $c_{\text{start}}$  とする。あらかじめ用意された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  $D_{\text{sub}}$  とする。例えば  $D = \{\text{word, eye}\}$  で、 $D_{\text{sub}} = \{\text{w, wo, wor, word, e, ey, eye}\}$  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  $x_t$  が得られたとき、 $x_t$  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  $C_t = \{c_{1,t}, c_{2,t}, \dots, c_{k,t}\}$  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  $c'_t \in (C_t \cup \text{NULL})$  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。

$$w_t = c_{\text{start}} + c'_1 + c'_2 + \dots + c'_t \quad (1)$$

(1) 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。(1) 式を満たす単語のうち、 $D_{\text{sub}}$  に含まれる辞書を  $W_{\text{sub}}(t)$  とし、 $W_{\text{sub}}(t)$  もトライ木で実装する。実際の入力では辞書  $W_{\text{sub}}(t)$  を更新し、 $W_{\text{sub}}(t)$  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 2 と図 3 を用いて示す。図 2 の黄線はある視線経路を表しており、図 2 を基に  $W_{\text{sub}}(t)$  が更新する様子を表したものが図 3 である。ただし、 $k = 3$  で最初の文字を  $c_{\text{start}} = 'n'$  としている。図 3 が示すように、 $W_{\text{sub}}(t)$  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  $W_{\text{sub}}(t)$  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表示される。

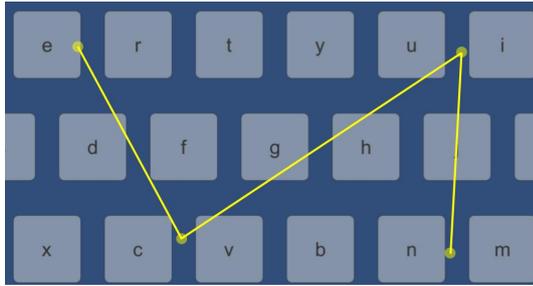


図 2: 簡単な視線経路. 黄線が視線経路を表している.

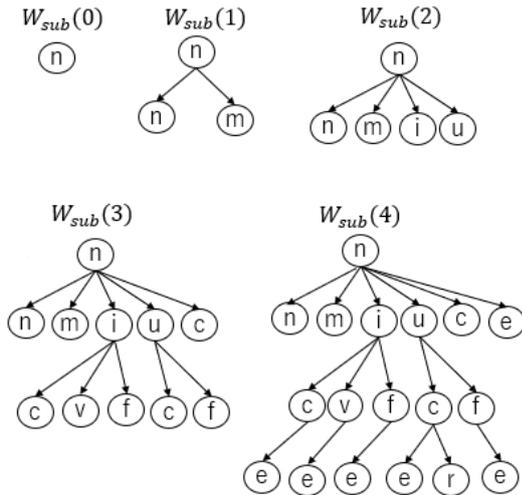


図 3:  $W_{sub}(t)$  の更新

## 5. 実験

### 5.1 実験環境

本実験では、アイトラッカとして Tobii Pro Nano を使用した。Tobii Pro Nano の視線入力のスAMPLINGレートは 60 Hz である。Tobii Pro Nano を Windows10 を搭載したノートパソコン (15.6 インチ, フルHD) に貼り付け、視線入力を取得した。また入力に使うスクリーンキーボードは QWERTY 配置であり、これは被験者が慣れているキーボード配置である。提案手法で使う辞書は、Google Web Trillion Word Corpus から Kaufman が抽出した最頻出の 1 万語を採用している [9]。

### 5.2 実験方法

実験では、単語の候補を絞り込む条件がない場合と、ある場合とで比較した。いくつかの単語に対して、被験者が視線で追跡することで、得られた視線入力を基にそれぞれの場合における類似度を計算した。なお、被験者はメガネを常時装着して視力を矯正しているが、アイトラッカが視線を取得しやすくするために、メガネを掛けずに実験を行った。文字入力の開始と終了は、マウスを

右クリックすることをトリガとすることで決定した。また提案手法における最初の文字  $c_{start}$  は、最初の視線入力座標から一番距離が近い文字とした。提案手法での近傍文字の数は  $k = 3$  とした。

入力する単語の長さによって、結果の傾向が異なることを考慮し、本実験では、短い単語として「is」、平均的な長さの単語として「this」、長い単語として「information」のそれぞれに対して比較した。

### 5.3 実験結果

「is」、「this」、「information」を入力したときの類似度の結果は表 1-3 のようになった。表 1 より、絞り込まな

表 1: 「is」入力時の類似度

順位	絞り込まない場合		絞り込む場合	
	候補単語	類似度	候補単語	類似度
1	its	1912.8	its	1912.8
2	irs	1950.3	irs	1950.3
3	ids	2084.8	ids	2087.1
4	ideas	2191.3	is	2320.5
5	hits	2200.7	isa	2458.8

表 2: 「this」入力時の類似度

順位	絞り込まない場合		絞り込む場合	
	候補単語	類似度	候補単語	類似度
1	things	5363.6	this	5514.1
2	this	5514.1	third	7578.0
3	ethics	5698.8	thus	8328.8
4	thincks	5812.7	those	8645.5
5	thickness	6222.2	tits	9762.9

い場合は、類似度が上位 5 つの中に「is」は現れなかった。一方で、絞り込む場合は、「is」が 4 番目に現れた。表 2 より、絞り込まない場合は、「this」の類似度は 2 番目であったのに対し、絞り込む場合は 1 番目であった。そして表 3 では、どちらのケースでも「information」が一番類似度が高かった。また表 2-3 の絞り込む場合を見ても、順位 1 の類似度と順位 2 以降の類似度に大きく差をつけた。

本実験では絞り込まない場合は辞書のすべて単語に対して、類似度を計算したが、絞り込む場合は、絞り込む条件にあった候補単語に対してのみ類似度の計算を行った。絞り込む場合における候補単語数は、「is」では 18 個、「this」では 53 個、「information」では 101 個得られた。

表 3: 「information」入力時の類似度

順位	絞り込まない場合		絞り込む場合	
	候補単語	類似度	候補単語	類似度
1	information	25958.3	information	25958.3
2	informational	27083.4	intention	44384.0
3	informative	31149.1	integration	44671.6
4	confirmation	31281.3	invitation	44746.0
5	instrumentation	37133.8	irrigation	45547.6

#### 5.4 考察

表 1-3 より、確かに提案手法により候補を絞り込んだ方が、目的の単語が類似度の高い候補として上位に現れる可能性が高い。これは、もし計算すれば目的の単語よりも類似度が高い単語が、提案手法の条件により除外されているからである。ただし、条件により目的の単語まで除外されると、目的の単語が候補として現れなくなってしまう。近傍文字の数  $k$  が大きいほど目的の単語が除外される可能性は下がるが、他の単語が候補として現れる可能性が高くなる。すなわち、目的の単語が候補に含まれる可能性を上げること、除外する他の単語を増やすことはトレードオフの関係にある。このことから、近傍文字の数  $k$  の値の大きさについて十分に検討する必要がある。

また、表 2-3 で目的の単語の類似度が他の単語の類似度と大きく差をつけたことから、目的の単語が 4 文字以上の長さであれば、安定して目的の単語が得られることが分かる。

次に候補単語数の観点から考察する。本実験では近傍文字の数を  $k = 3$  とし、絞り込みにより辞書にある単語 1 万語から 18-101 語にまで候補を減らすことが出来た。すなわち、類似度の計算もそれだけ少なく済んだ。辞書に含まれる単語の中で「i」から始まる単語は 386 個、及び「l」から始まる単語は 549 個である。これと本実験の候補単語数を比較すると、近傍文字情報は短い単語では候補を  $\frac{1}{20}$  倍程度に、長い単語では候補を  $\frac{1}{4}$  倍程度、減らすことが可能であることが分かる。

EyeSwype では単語の最初の文字だけでなく最後の文字も入力し、それらが正しいことを前提として単語を絞り込んでいた。そのために、最初と最後の文字決定時、一定の視線ジェスチャをする必要があった。一方で提案手法は最後の文字を限定せずに、候補単語を十分に減らすことが可能である。すなわち、単語入力の終了を表す一定の操作は、最後の文字を決定する必要がないため、提案手法を用いれば、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。あ

#### 6. まとめ

本稿では、先行研究の EyeSwype を改善し、視線入力との近傍文字による単語の候補を絞り込むことで、単語予測を高精度化する手法を提案した。実験により、提案手法は目的の単語を候補から除外することなく、十分に候補を絞り込むことが確認できた。今後は、先行研究との比較実験を行うことで、提案手法の有効性を検証する予定である。また EyeSwype の課題の 1 つである視線を素早く動かす必要があることを考慮した手法について検討していきたい。

#### 参考文献

- [1] Andrew Kurauchi, Wenxin Feng, Aijen 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 Systems May 2016*, pp.1952–1956 (2016)
- [2] Ward, DJ, Blackwell AF, MacKay DJ: "Dasher: a data entry interface using continuous gestures and language models", *In: ACM symposium on user interface software and technology (UIST)*, pp.129–137 (2000)
- [3] Daniel Rough, Keith Vertanen, and Per Ola Kristensson: "An Evaluation of Dasher with a High-performance Language Model As a Gaze Communication Method" *In Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces (AVI '14)*. ACM, New York, NY, USA, pp.69–176 (2014)
- [4] Martez E. Mott, Shane Williams, Jacob O. Wobbrock, Meredith Ringel Morris: "Improving Dwell-Based Gaze Typing with Dynamic, Cascading Dwell Times" *CHI '17: Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems May 2017*, pp.2558–2570 (2017)

- [5] Jacob O. Wobbrock, James Rubinstein, Michael W. Sawyer, Andrew T. Duchowski: "Longitudinal evaluation of discrete consecutive gaze gestures for text entry" *ETRA '08: Proceedings of the 2008 symposium on Eye tracking research & applications March 2008*, pp.11–18 (2008)
- [6] Nikolaus Bee, Elisabeth André: "Writing with Your Eye: A Dwell Time Free Writing System Adapted to the Nature of Human Eye Gaze" *In Perception in Multimodal Dialogue Systems, Elisabeth André, Laila Dybkjær, Wolfgang Minker, Heiko Neumann, Roberto Pieraccini, and Michael Weber (Eds.). Lecture Notes in Computer Science, Vol. 5078. Springer Berlin Heidelberg*, pp.111–122 (2008)
- [7] Carlos H. Morimoto, Arnon Amir: "Context Switching for Fast Key Selection in Text Entry Applications" *In Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications (ETRA '10). ACM, New York, NY, USA*, pp.271–274 (2010)
- [8] Per Ola Kristensson, Keith Vertanen: "The potential of dwell-free eye-typing for fast assistive gaze communication" *ETRA '12: Proceedings of the Symposium on Eye Tracking Research and Applications-March 2012*, pp. 241–244 (2012)
- [9] Josh Kaufman: Google 10000 English. <https://github.com/first20hours/google-10000-english>. (2022) Accessed: 2022-07-14.