

## 2段階の視線移動を用いたコマンド実行手法

儀本 俊弥<sup>1,a)</sup> 山中 祥太<sup>2,b)</sup> 志築 文太郎<sup>3,c)</sup>

**概要:** 我々は、操作対象を一定時間見つめた（凝視）後に、右を見た後に上を見るというような2段階の視線移動を行うことによるコマンド実行手法を提案する。提案手法は、コマンド実行に要する時間が短かつ意図しないコマンド実行が生じにくい手法である。さらに、アイコンなどのGUI部品を対象としたコマンド（例えば「あるファイルのコピー」）だけではなく、画面内のGUI部品がない場所におけるコマンド（例えば「複数デスクトップの切り替え」）も実行可能である。提案手法において、コマンド実行に要する時間を短くし、かつ意図しない実行を防ぐために、ユーザが選択のために対象を凝視する必要がある時間や、システムが2段階の視線移動を認識するためのパラメータを実験を通じて決定した。

### 1. はじめに

視線に基づく入力手法として、凝視（ある対象を一定時間見つめること）に基づく選択が用いられてきた [1, 2]。すなわちユーザが画面に表示されている対象（例えばアイコン）を凝視するとその対象が選択される。この入力手法において、凝視が認識されるまでユーザがある対象を見つめ続けるべき時間は凝視時間と呼ばれる。ただし、この凝視時間を短くするとミダタッチ（ユーザが意図していない対象の選択） [1] が生じやすくなる。この結果、例えば、移動中の視線がアイコンの中にたまたま入るだけでアイコンがユーザの意図に関係なく選択されるようになる。ミダタッチを防ぐには、凝視時間を長くする、もしくは選択可能な対象同士を離して配置する必要がある。しかしながらこれらに伴い、凝視に基づく選択を用いたコマンド実行（以降、凝視に基づくコマンド実行）を行う際に要する時間が長くなるという問題や、対象の配置を考慮するために UI 設計に制限が生じる。

この課題に対して、視線の動き（以降、ジェスチャ）に基づくコマンド実行手法が開発されている [3-6]。このコマンド実行手法には2種類ある。1つ目は、視線をある方向へ動かすというジェスチャを用いる手法（以降、ストロー

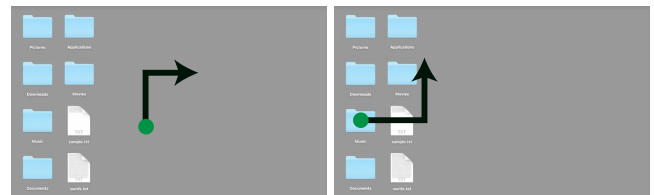


図 1 2段階の視線移動を用いたコマンド実行手法。グローバルコマンド（左）とオブジェクトコマンド（右）ともに実行できる。左：複数のデスクトップを切り替える。右：フォルダを選択してコピーする。

ク手法) [3,4,7,8] である。2つ目は、動く対象（以降、ガイド）を見ている時、そのガイドの動きにつられて視線が動くこと（円滑性追跡眼球運動）をジェスチャとして用いる手法（以降、ガイド追跡手法） [5,6,9,10] である。ガイド追跡手法におけるコマンド実行は、例えばコマンドを実行したい対象の周りに表示されているガイドを見続けるといったジェスチャ [5] や、凝視を用いて対象を選択した後に表示されるガイドを見続けるといったジェスチャ [6] により行われる。

これら2つの手法は、それぞれに利点を持つが、改善すべき点も持つ。ストローク手法によるコマンド実行は凝視によるコマンド実行より早い、多くのストローク手法ではコマンド実行の対象を選択するための手段が提供されていない。グローバルコマンドのみが実行可能である（例：Webブラウザにおける「前のページに戻る」、画面における「複数デスクトップを切り替える」）。一方、ガイド追跡手法では、GUI部品に対するコマンド（オブジェクトコマンド）が実行可能である（例：あるフォルダを「コピーする」）。しかし、ガイドが対象の周りに常に表示されている手法 [5] では、ガイドのために、UI設計への制限が生じる。

<sup>1</sup> 筑波大学コンピュータサイエンス専攻  
Department of Computer Science, University of Tsukuba  
<sup>2</sup> ヤフー株式会社  
Yahoo Japan Corporation  
<sup>3</sup> 筑波大学システム情報系  
Faculty of Engineering, Information and Systems, University of Tsukuba  
a) isomoto@iplab.cs.tsukuba.ac.jp  
b) syamanak@yahoo-corp.jp  
c) shizuki@cs.tsukuba.ac.jp

また、凝視後にガイドが表示される手法 [6] では、ミダスタッチによってガイドがユーザの意図に反し頻繁に表示されることを防ぐために凝視時間を長くする必要があり、これに伴ってコマンド実行に要する時間（凝視時間 + ジェスチャを行うための時間）が長くなる。

本稿において我々は、グローバルコマンドおよびオブジェクトコマンドを両方とも実行可能とするストローク手法を示す（図 1）。ユーザは、対象の選択を短時間の凝視により行い、コマンドの実行を、右を見た後に上を見るような 'L' 字の視線移動（以降、2 段階の視線移動）により行う。凝視時間を短くした場合においても、2 段階の視線移動が認識されない限りユーザの意図しないコマンドは実行されない。そのため、ミダスタッチを防ぐために、凝視時間を長くするもしくは対象を離して配置する必要がない。

## 2. 関連研究

本節において、まず凝視に基づくコマンド実行手法の課題、およびジェスチャを用いた手法がコマンド実行手法として有望である理由を述べる。その後、提案手法に関連するストローク手法およびガイド追跡手法を述べる。

### 2.1 凝視に基づくコマンド実行手法

凝視に基づくコマンド実行手法を用いる場合、マウスを用いる場合と同様に、ユーザは画面に表示されている GUI 部品の中からコマンドを選択する必要がある。ただし、例えば、プルダウンメニューを用いる場合は、実行したいコマンドが含まれるメニューを表示するために少なくとも 1 度、コマンドの実行に 1 度の合計 2 回以上の選択を行う必要がある。さらに、これらの GUI 部品は、密集して配置されている。そのため、凝視に基づくコマンド実行手法では、ミダスタッチによりユーザの意図しないコマンドが実行される。また、メニューの表示がミダスタッチによって行われることにより、オクルージョンが発生する可能性がある。したがって、凝視に基づくコマンド実行手法を採用して GUI 部品を操作させる場合には、ミダスタッチを防ぐために凝視時間を長くする必要があり、この問題は、短い凝視時間においてミダスタッチの発生率を減らそうとしている研究を用いても生じる。例えば、Isomoto ら [11] の手法は、選択対象が密集している場合はミダスタッチが生じるため、密集して配置される事が多い GUI には不向きである。そのため、凝視に基づくコマンド実行手法を行う場合は、GUI 部品がそれぞれ離れて配置されているような専用のインタフェースが必要となる。さらに、コマンドを実行する手法として凝視に基づく手法より、ジェスチャを用いた手法の方が素早くかつ正確である事が知られている [12,13]。

これらの理由より、視線に基づく入力手法を用いてコマンドを実行する際は、ジェスチャに基づく手法が有望であ

る。ジェスチャに基づく手法において、凝視に基づく選択を対象の選択に用いるが、ミダスタッチによりユーザの意図しない選択が生じて、ジェスチャが行われなくなる限り、ユーザの意図しないコマンドは実行されない。我々の手法においても、凝視に基づく選択は対象の選択にのみ用いられる。また、コマンド実行のために画面にメニューなどを表示する必要がなく、インタフェース設計の点において自由度が高い。

### 2.2 ジェスチャを用いたコマンド実行手法

提案手法が、先行研究におけるストローク手法およびガイド追跡手法それぞれとどのように異なるかを述べる。

#### 2.2.1 ストローク手法

Istance ら [7] は、画面に半透明な領域を常に表示し、その領域を通るような 2 または 3 段階の視線移動を用いたコマンド実行手法を示した。Wobbrock ら [14] は、EdgeWrite [15] を視線に基づく操作に適応し、画面 4 隅を通る視線移動を用いた文字入力手法を示した。これらの手法は、画面にコマンド実行のための領域を表示する手法である。一方、画面に何も表示しない手法も提案されている。Drewes ら [3] は、画面の 4 隅を通る視線移動を用いたコマンド実行手法を提案した。例えば、画面右上端から左上端への移動を 3 往復するような視線移動である。Rajanna ら [4] は、画面内における 1 段階および U 字の視線移動を用いたコマンド実行手法を示した。Jungwirth ら [17] は、日常に存在する物理デバイスの縁をなぞるような視線移動を用いたコマンド実行手法を示した。例えば、ランプの縁が視線にてなぞられるとそのランプが点灯する。Möllenbach ら [8,18,19] は、画面内における上下もしくは左右の視線移動をコマンド実行手法として用いて実験を行った。実験の結果、水平方向の視線移動は垂直方向の視線移動より素早く行えること、および、画面にコマンド実行のための経路や領域を表示しない方が素早くコマンド実行が可能であることを示した。ただし、上記の手法は、グローバルコマンドを実行可能であるがオブジェクトコマンドを実行するための手段は提供されていない。

対して、オブジェクトコマンドを実行可能とする手法も提案されている。Urbina ら [20] は、パイメニュー [21] を用いたコマンド実行手法を示した。この手法において、GUI 部品の選択は対象を見ることにより行われ、対象を見た後に表示されるパイメニューの縁を横断することにより、コマンドが選択した対象に実行される。しかし、対象を見た後にコマンドの実行を止めること（コマンド実行の中止）ができないという問題がある。Ware および Mikaelian [2] は、ある対象を見た後、画面に常に表示されているボタンを凝視時間の短い凝視することにより対象を選択する手法を提案している。オブジェクトコマンドを実行するためには、対象およびコマンドへ 1 度ずつ選択を行う必要があ

る。これらの手法は、対象を見た後に視線をパイメニューの縁もしくはボタンへ移すという点においてジェスチャを用いた手法と言える。対象の選択を凝視時間が短い凝視に基づく選択にて行うため、ミダスタッチが生じるが、意図しないコマンドはジェスチャを行わない限り実行されない。しかし、対象を見た後の意図しないジェスチャにより、意図しないコマンドが実行される可能性がある。

提案手法はグローバルコマンドおよびオブジェクトコマンドの実行を可能とするストローク手法である。対象の選択を短い時間の凝視により行い、コマンドは対象の選択後に行われる2段階の視線移動により実行される。提案手法において、画面にコマンド実行のための領域などを表示しないためオクルージョンは発生せず、対象を選択した後にコマンド実行を中止することができる。1段階の視線移動を用いた手法において生じるような意図しないコマンド実行を防ぐために、提案手法における2段階の視線移動は、実験を通じて決定した視線移動の経路幅などを基に検出される。

### 2.2.2 ガイド追跡手法

Vidal ら [9] が円滑性追跡眼球運動をジェスチャに用いた手法を示して以来、多くのガイド追跡手法が提案されてきた。Esteves ら [5] はスマートウォッチ向けに拡張し、Velloso ら [10] は画面に表示もしくは物理デバイスに投影されたガイドを見続けることによるコマンド実行手法を示した。また、Špakov ら [22] は凝視に基づく選択とガイド追跡手法を比較し、凝視に基づく手法における選択に要する時間と選択の成功率がガイド追跡手法と同等もしくは、優れていることを示した。ガイドは一般的にある対象の周辺に複数表示され、それぞれにその対象へのコマンドが割り当てられている。ユーザは、ガイドに表示されている情報（コマンドを表すアイコンや文字など）をもとに任意のコマンドを実行できる。これらの手法において、ユーザはどのコマンドがどのジェスチャに割り当てられているかを覚える必要がないが、ガイドが常に表示されているためUI設計に制限が生じやすくなる。この問題を解決するために、凝視に基づく選択とガイド追跡手法を組み合わせた手法が Delamare ら [6] によって提案されている。この手法において、まずユーザは凝視に基づく選択を用いてコマンドを実行したい対象を選択する。その後、表示されるガイドのうち任意の1つを見続けることにより任意のコマンドを実行する。しかし、凝視に基づく選択を行うためにミダスタッチが生じてしまう。ミダスタッチを防ぐために、Delamare ら [6] は、凝視時間を2秒としている。結果として、凝視時間が長くなるに従いコマンドの実行により長い時間を要する。

提案手法は画面にコマンド実行に必要な領域やガイドなどを表示しないため、UI設計に制限が生じない。また、凝視に基づく選択後に2段階の視線移動を行うことによりは

じめてコマンド実行が行われるため、凝視時間の長さによる意図しないコマンド実行は行われない。しかし、凝視に基づく選択後に意図しない2段階の視線移動が認識されるとコマンド実行を引き起こす可能性がある。そこで、提案手法における2段階の視線移動の検出は、ユーザが意図せず2段階の視線移動が行われないよう、実験を通じて決定した視線移動の経路幅などを基に行われる。

## 3. 提案手法

提案手法に用いる2段階の視線移動は、'L'字の経路を通るような移動である。2段階の視線移動を行うためには、ユーザはまず水平方向もしくは垂直方向に視線を移動させ(1段階目)、その後、1段階目に視線を動かさなかった方向、つまり垂直方向もしくは水平方向に視線を移動させる(2段階目)。つまり提案手法において実行可能な2段階の視線移動は、右を見た後に上(右→上)や左を見た後に下(左→下)を見るようなジェスチャであり、この2つに加え、右→下、左→上、上→右、上→左、下→右、下→左の8種類である。また、これらの2段階の視線移動それぞれに、任意のコマンドを割り当てる事ができる。

また、提案手法における、コマンド実行は以下の3要素から構成される。

**対象の選択** GUI部品内のある点やGUI部品が配置されていないある点を  $T_{\text{dwell}}$  時間以上凝視することにより行われる。

**コマンドの実行** ある点への凝視後、凝視が行われた点(凝視点)を始点とした2段階の視線移動が行われると、その方向に対応したコマンドが実行される。凝視した点がGUI部品に含まれている場合は実行されるコマンドがオブジェクトコマンドとなり、そうでない場合はグローバルコマンドとなる。

**コマンド実行の中止** ある点への凝視後、2段階の視線移動中に視線が留まった(対象の選択が行われた)場合は、その凝視点を始点とした2段階の視線移動が行われない限りコマンドは実行されない。

### 3.1 2段階の視線移動の認識

提案手法における2段階の視線移動の検出方法を以下に示す(図2)。

- 凝視が行われた時、凝視点を中心に幅  $W_{\text{path}}$  の1段階目の経路を生成する。
- 1段階目の視線移動は、水平(垂直)方向にある閾値  $D_{\text{thld}}$  以上、視線が凝視点から移動することにより認識される。この時視線が1段階目の経路から外に出ると2段階の視線移動の認識は終了する。
- 1段階目の視線移動が認識された後、視線が1段階目の経路から外に出た点を中心に幅  $W_{\text{path}}$  の2段階目の経路が生成される。

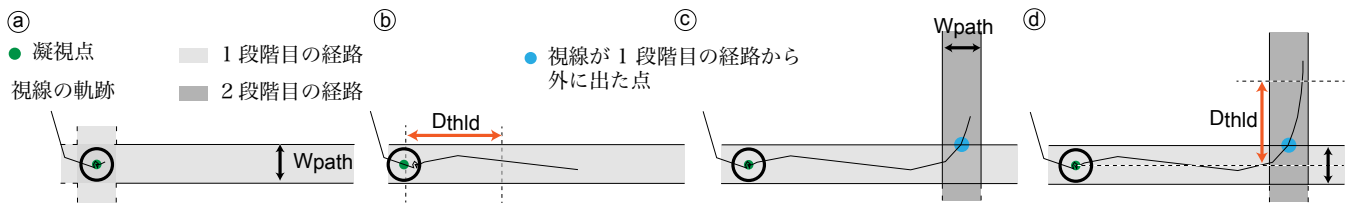


図2 提案手法における2段階の視線移動の認識方法.

- d) 2段階目の視線移動は、垂直（水平）方向に閾値  $D_{thld}$  以上、視線が1段階目の経路から外に出た点から移動した場合認識され、認識されるとコマンドが実行される。この時、凝視した点に対象がある場合はその対象へコマンドが実行され、対象はない場合は対象をとらないコマンドが実行される。

1段階目および2段階目の経路は画面に表示されない。凝視時間 ( $T_{dwell}$ )、経路の幅 ( $W_{path}$ )、および視線移動認識を認識するための閾値 ( $D_{thld}$ ) は実験的に決定される。

#### 4. 実験1：パラメータの決定

提案手法における、2段階の視線移動を検出するために用いるパラメータを決定するための実験を行った。実験1を通じて決定したパラメータは  $T_{dwell}$ 、 $W_{path}$ 、および  $D_{thld}$  の3つである。パラメータは、例えば画面中心から右上に視線を移動する時、斜めに視線が移動した場合は2段階の視線移動と認識せず、右→上のように視線が移動した場合は2段階の視線移動と認識するように決定する。パラメータを決定するために、実験1では1段階の視線移動および2段階の視線移動を行う際の視線データ（視線座標およびそのタイムスタンプ）を収集する。

##### 4.1 実験参加者

実験1において、16名（女性2名）を実験参加者として雇用した。実験参加者は20–24歳（平均22.0歳）であり、6人が眼鏡、2人がコンタクトを着用していた。11名が以前に視線認識機器を使用したことがあり、いずれも日常的には使用していなかった。我々は実験参加者をそれぞれ裸眼4名、眼鏡3名、およびコンタクト1名ずつのグループ（AおよびB）に分けた。

##### 4.2 実験機器

実験1において使用した視線認識機器はTobii EyeX<sup>\*1</sup>であり、画面における背景などの反射を防ぐためにノングレアタイプの画面（24インチ、解像度：1,920×1,080 pixels）を用いた。実験参加者には画面から約60cm離れるよう指示し、実験中の実験参加者の動きによる視線認識精度の低下を防ぐために、顎のせ台に顎をのせるよう指示した。

##### 4.3 実験手順

実験参加者に1段階の視線移動（条件1）および2段階の視線移動（条件2）を行うことを指示した。条件1では、実験参加者に「（ある方向）を見てください」と指示した。（ある方向）には右上、左上、右下、および左下の4方向が該当する。条件2では、実験参加者に「（1段階目の方向を見た後に2段階目の方向）を見てください」と指示した。（1段階目の方向を見て2段階目の方向）には、右→上、左→上、右→下、左→下、上→右、上→左、下→右、および下→左の8方向が該当する。なお、条件2における視線移動の方向を変える際は、視線を止める必要はないと指示した。また、各条件において、画面上のどこまでを見るかまでは指示しなかった。

本実験での1試行は各条件において1度指示に従い視線を移動させることである。各試行において視線移動の開始位置を固定するために、各試行前に画面の中心に表示されている黒塗りの円（直径28mm）を見るよう指示した。試行は事前に録音された各条件における指示が再生された後に開始される。指示は実験参加者の手元にあるキーボードのキーを押すことにより再生される。指示通りの視線移動を行った後に再度キーを押すと1試行が終了する。実験参加者が再度キーを押すことにより次の試行が開始する。疲れの影響を減らすために、試行間に任意の休憩、各条件32試行ごとに少なくとも5分間の休憩を設けた。各条件において指示される方向を条件1において4方向、条件2において8方向ずつをランダムとした。また、条件間の順序効果を相殺するために、グループAは条件1から実験を開始し、グループBは条件2から実験を開始した。各条件の開始前に視線認識機器のキャリブレーションを行った。視線認識結果と実際に見ている点の誤差による影響をなくすために、視線認識結果に関する情報を実験参加者に通知しなかった [23]。

実験1を通じて、条件1において1,024（4方向×16試行×16人）試行、条件2において2,048（8方向×16試行×16人）試行分のデータを収集した。各条件の終了後、実験参加者に各条件の視線移動に関するアンケートの記入を指示した。アンケートへの記入を終了するまでに要した時間は約52分であった。

\*1 <https://tobiigaming.com/product/tobii-eyex/>  
(2018年1月26日最終閲覧)

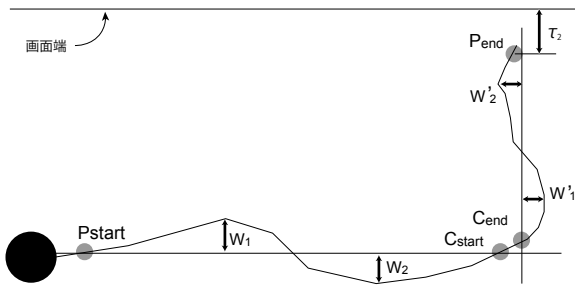


図 3 実験 1 の解析方法.

#### 4.4 解析

条件 2 の試行における視線移動の軌跡を観察したところ 2 つの特徴を観測した. 1 つは, 視線移動の方向を変える際 (例えば, 右 → 上において右を見た後に上に移動を開始するまで) に視線が留まっていることであり, もう 1 つは, 視線移動時に直線に視線が移動せず, ジグザグに移動している (視線移動時にブレていた) ことである. そのため, これらの特徴をパラメータを決定するための指標として用いる.

我々はまず, これらの特徴を求めるために, 収集した視線データに対して以下の処理を行った. まず, 視線座標のばらつきを除くために各試行の視線座標に対しローパスフィルタ ( $P_i = 0.25p_i + 0.75P_{i-1}$ :  $P_i$  はローパスフィルタ適用後の  $i$  番目の視線座標,  $p_i$  は  $i$  番目に収集された視線座標) を適用した. そして, 指示と異なる方向へ視線が移動していた試行 (例えば, 右 → 上と指示した際に, 上 → 右のように視線が移動していた試行) を取り除いた. この結果, 全試行の内 11.8% の試行が取り除かれた. 取り除かれなかった 88.2% の試行 (以降, 正解試行) を用いて, 視線移動の開始座標 ( $P_{start}$ ), 視線移動の方向が変わる座標 ( $C_{start}$ ), 視線が  $C_{start}$  から移動し始めた座標 ( $C_{end}$ ), および視線移動が終了した座標 ( $P_{end}$ ) を以下のように決定した (図 3).

$P_{start}$  1 段階目の方向が右の場合, 視線が図 3 に示す黒色の円 (画面表示した黒い円と同じく直径 28 mm) から外に出ている, かつ  $d_x$  がある  $\tau_1$  以上移動した座標. 1 段階目の方向が上の場合,  $d_x$  ではなく  $d_y$  が  $-\tau_1$  移動した座標.

$C_{start}$   $P_{start}$  が検出された後に  $d_x$  が  $\tau_1$  より小さくなった座標.  $d_x$  が  $\tau_1$  より大きくなった場合は  $C_{start}$  の検出を再度行う.

$C_{end}$  視線座標が  $C_{start}$  から垂直方向に  $\tau_1$  より移動し,  $d_y$  が  $d_x$  より大きい座標.

$P_{end}$   $C_{end}$  が検出された後に  $y_i$  が画面上端から  $\tau_2$  (もしくは画面下端から  $\tau_2$ ) 以内まで移動した座標.

本稿において, ある点を見ている際の視線座標は約 5 mm のばらつきにて取得されること [24] から, 視線がある点に留まる際の閾値として  $\tau_1$  を 5 mm とした. また, 正解試行

における視線移動の軌跡の多くが画面端まで移動していたことから, 各正解試行において  $P_{end}$  を決定できるように  $\tau_2$  を 55.5 mm とした.

##### 4.4.1 視線移動の方向を変える際に視線が留まった時間

視線移動の方向を変える際に視線が留まった時間を, 正解試行において  $C_{end}$  と  $C_{start}$  におけるタイムスタンプの差とした. 計算した結果, 条件 2 における, 視線移動時に視線が留まった時間の平均時間は 133.2 ms (標準偏差: 86.4) であった.

##### 4.4.2 視線移動時のブレ

視線移動時のブレを, 正解試行において  $P_{start}-C_{start}$  間および  $C_{end}-P_{end}$  間にどれだけ視線移動の方向と直交する方向に視線が移動したかを用いて決定した. 例えば図 3 では, 1 段階目の視線移動の方向が水平方向であるため, 上方向へのブレの最大 ( $W_1$ ) と下方向へのブレの最大 ( $W_2$ ) 値の和を計算し, 2 段階目の視線移動の方向が垂直方向であるため, 右方向へのブレの最大 ( $W_1'$ ) と左方向へのブレの最大 ( $W_2'$ ) 値の和を計算する. 計算した結果, 視線移動時のブレの平均は,  $P_{start}-C_{end}$  において 17.0 mm (標準偏差: 25.9) であり,  $C_{end}-P_{end}$  間において 17.8 mm (標準偏差: 24.1) であった.

#### 4.5 実験結果

実験結果をもとに各パラメータを決定する.

##### 4.5.1 凝視時間

提案手法は, 対象選択に凝視に基づく選択手法を採用したため, 視線移動中においても新たな対象が選択される可能性がある. 4.4.1 節に示した通り, 視線移動中に視線が留まることがわかったため,  $T_{dwell}$  として適当な時間を決定する必要がある. そこで,  $T_{dwell}$  を視線移動中に視線が留まった時間の平均と標準偏差の和とした. したがって,  $T_{dwell}$  は 219.6 (133.2 + 86.4) ms である. これは, 条件 2 における視線移動時に視線が留まった時間の 85.9 パーセント値である.

##### 4.5.2 経路の幅

提案手法において,  $W_{path}$  を大きくすぎると意図しないコマンドが発生しやすくなり, 小さくすぎるとコマンド実行が難しくなるため, 適当な  $W_{path}$  を決定する必要がある. 4.4.2 節に示したとおり, 視線移動時は視線の移動方向に対し直線的に移動するのではなく, ジグザグに移動していることがわかった. そこで, 計算した視線移動時のブレの平均と標準偏差の和を用いて,  $W_{path}$  を決定した.  $P_{start}-C_{end}$  間および  $C_{end}-P_{end}$  における視線移動時のブレの平均と標準偏差の和は, それぞれ 42.9 mm (17.0 + 25.9) および 41.9 mm (17.8 + 24.1) であった. また, ウィルコクソンの符号順位検定 ( $\alpha = 0.05$ ) の結果,  $P_{start}-C_{end}$  間および  $C_{end}-P_{end}$  間のブレの平均と標準偏差の和に有意差はなかった. そこで, 値が大きかった 42.9 mm を  $W_{path}$

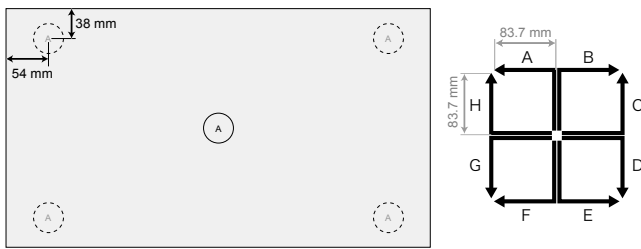


図 4 実験 2 に用いる画面。左：円はターゲットの位置を示し、円中の文字は視線移動の方向を示す。右：円の中の文字に対応する方向を示す。

とした。これは、条件 2 における視線移動時のブレの 93.2 パーセント値である。

#### 4.5.3 視線移動認識を認識するための閾値

$D_{thld}$  も  $W_{path}$  と同様な理由から、適当な値とする必要がある。既に決定した  $T_{dwell}$  および  $W_{path}$  を用いて、 $D_{thld}$  の値を変更したときに、どれだけ条件 1 の試行が 2 段階の視線移動と認識されるかをもとに決定する。このとき、条件 1 が 2 段階の視線移動と認識された場合、意図しない 2 段階の視線移動が認識され意図しないコマンド実行が行われることになる。条件 1 および条件 2 が 2 段階の視線移動と認識された確率は、 $D_{thld}$  が 27.9 mm のときそれぞれ 9.9% および 74.5%、 $D_{thld}$  が 41.9 mm のとき 0.5% および 15.8%、および  $D_{thld}$  が 55.8 mm のとき 0.0% および 5.4% であった。この結果から、意図しないコマンド実行を防ぐために、条件 1 が 2 段階の視線移動と認識された確率が 0.0% であった 55.8 mm を  $D_{thld}$  として用いることにする。なお、条件 2 が 2 段階の視線移動と認識された確率が 5.4% と低い、これは、実験時にどのように視線を移動させるかまで指示していなかったことが原因であり、ユーザにどのように視線を移動させると良いかを指示することにより、条件 2 の認識率が向上すると考えられる。

## 5. 実験 2：提案手法の性能評価

実験 1 にて決定したパラメータを用いた際の提案手法の性能評価を調査する実験を行った。実験 2 において求める提案手法の性能は、2 段階の視線移動が画面中央および画面 4 角から始まる場合における、2 段階の視線移動の認識率および視線移動に要する時間である。実験に用いる機材および実験環境は実験 1 と同じである。また、実験 2 でも、視線データに対して実験 1 と同じローパスフィルタを適用した。

### 5.1 実験参加者

男性 16 名を実験参加者として雇用した。実験参加者は 21-24 歳（平均 23.0 歳）であり、2 人が眼鏡を着用していた。13 名が以前に視線認識機器を使用したことがあり、いずれも日常的には使用していなかった。

### 5.2 実験手順

実験 2 には図 4 に示す画面を用いた。図 4 左に示すように、コマンド実行の対象を右上、左上、右下、左下、および中心の 5 箇所いずれかに表示した。また、図 4 右に示すように、8 方向それぞれを 'A' から 'H' までの文字に対応づけた。実験参加者へ凝視が認識されたことを通知するために、対象への凝視が認識されると文字を赤色にした。視線認識機器は 84 mm 程度画面外の視線座標を認識することが可能であるため、対象の位置が画面の角である場合においても 2 段階の視線移動の認識は可能である。

本実験では、実験参加者が対象を凝視した上で、対象内に書かれている文字に対応する方向へ視線を移動することを 1 試行とした。実験は休憩、実行、説明の 3 状態から構成される。実験は休憩から始まる。このとき画面には何も表示されていない。実験参加者が任意の休憩後に 'Enter' を押すと実行に状態が遷移する。実行に遷移すると、画面に対象と文字が表示される。実行において、文字と方向の対応がわからない場合は 'Space' を押すよう指示した。'Space' が押されると図 4 右が画面に表示され、再度 'Space' を押すと表示が消える。実験参加者が 1 試行を終えた後に 'Enter' を押すと説明に状態が遷移する。説明に遷移すると、その試行が指示通りに 2 段階の視線移動が認識されたかを通知するとともに、その試行において収集された視線座標を画面に表示する。このとき、実験参加者はどのように視線が移動したため認識された（されなかった）かを確認する。その後、'Enter' を押すと休憩に状態が遷移する。

40 (8 方向 × 5 箇所) 試行を行うことを 1 セッションとし、6 セッション行った。前半 3 セッションを練習セッションとし後半 3 セッションを本番セッションとした。セッションごとに対象の表示位置および視線移動の方向の指示をランダムとした。練習セッションのみ、実験監督者は説明中になぜ認識されなかったのかを実験参加者に説明した。最初のセッションの開始前に視線認識機器へのキャリブレーションを行った。視線認識結果に関する情報を実験 1 と同様に実験参加者に通知しなかった [23]。練習セッションおよび本番セッションそれぞれにおいて、合計 1,920 (40 試行 × 3 セッション × 16 人) 試行の視線データを収集した。全セッション終了後、実験参加者に 2 段階の視線移動に関するアンケートの記入を指示した。アンケートへの記入を終了するまでに要した時間は約 85 分であった。

### 5.3 実験結果

データを用いて、まず画面中心における性能に関する結果を述べ、実験 1 にて求めたパラメータの正当性を検討する。その後、図 4 左に示す画面 4 角における性能に関する結果を述べ、実験 1 にて求めたパラメータの汎用性を検討する。求める性能は 2 段階の視線移動の認識率およびコマンド実行に要する時間とした。認識率は全試行のうち 2 段

階の視線移動が認識され指示と視線移動の方向があっている試行がどれだけあるかをもとに計算し、時間は対象を凝視し始める時間から2段階の視線移動が認識されるまでの時間を用いた。以降、検定にはウィルコクソンの符号順位検定を用いた ( $\alpha = 0.05$ )。

### 5.3.1 画面中心における性能評価

認識率の平均は、練習セッションにおいて74.7%、本番セッションにおいて77.6%、および全セッションにおいて76.2%であった。練習および本番セッションにおいて、検定の結果、練習セッションの1-3および2-3セッション間に有意差があり、提案手法への経験が認識率に影響を与えることがわかった。練習セッションの説明において、実験参加者へ視線移動の方向を変えるときに視線を止めないよう頻りに指示していたこと、実験参加者が「視線移動中に視線を留めないようにすることが難しい」とアンケートに回答したことから219.6msは凝視時間として短すぎるということがわかった。そこで、 $T_{\text{dwell}}$ を306.0ms、392.4ms、および478.8ms(それぞれ実験1における平均と2, 3, 4倍の標準偏差の和)とした際の認識率も調査した。これらは実験1の条件2における視線移動時に視線が留まった時間の95.5, 98.6, 99.5パーセント値であった。収集した視線データを用いて、 $T_{\text{dwell}}$ を変更した際の全セッションにおける認識率の平均はそれぞれ、82.6%、84.2%、および84.7%であり、検定の結果、306.0-392.4ms間および392.4-478.8ms間に有意差がないことがわかった。この結果より、 $T_{\text{dwell}}$ を306.0msとしたときの認識率に関してさらに調査した。練習および本番セッションにおける認識率の平均は、それぞれ80.2%および84.9%であり、本番および練習セッションにおける3セッション目はともに89.8%であった。

コマンド実行に要する時間の平均は、練習セッションにおいて822ms、本番セッションにおいて813ms、および全セッションにおいて817msであった。検定の結果、練習および本番セッションの各セッション間に有意差はなかった。認識率と同様に、3つの $T_{\text{dwell}}$ を用いた際の時間も調査した。全セッションの時間の平均はそれぞれ、918ms、1013ms、および1103msであり、検定の結果、392.4ms-478.8ms間のみに有意差がなかった。また、 $T_{\text{dwell}}$ を306.0msとしたときの練習および本番セッションにおける時間の平均は、それぞれ925msおよび912msであり、本番および練習セッションにおける3セッション目はともに900msであった。

### 5.3.2 画面4角における性能評価

画面4角における認識率の平均は、練習セッションにおいて57.8%、本番セッションにおいて62.5%、および全セッションにおいて60.2%であった。検定の結果、練習および本番セッションの各セッション間に有意差はなかった。画面右上、左上、右下、および左下における練習セッション

の認識率の平均は、それぞれ52.1%、55.0%、62.0%、および62.2%であった。本番セッションの認識率の平均は、それぞれ50.8%、61.7%、70.1%、および67.5%であった。検定の結果、左下-中心の間にのみ有意差がなかった。 $T_{\text{dwell}}$ を306.0msとしたときの認識率についても調査したところ、画面4角における本番セッションの認識率の平均は、67.2%であった。また、右上、左上、右下、および左下における本番セッションの認識率の平均は、それぞれ56.2%、65.1%、74.7%、および72.7%であった。検定の結果、各場所と中心の間に有意差があった。

練習セッションにおける時間の平均は、練習セッションにおいて865ms、本番セッションにおいて855ms、および全セッションにおいて860msであった。検定の結果、練習および本番セッションの各セッション間に有意差はなかった。画面右上、左上、右下、および左下における本番セッションの時間の平均は、それぞれ905ms、872ms、847ms、および833msであった。検定の結果、左下と右下および中心の間に有意差がなかった。 $T_{\text{dwell}}$ を306.0msとしたときの時間についても調査したところ、右上、左上、右下、および左下における本番セッションの時間の平均は、それぞれ1012ms、990ms、967ms、および958msであった。検定の結果、左下と右下および中心の間に有意差がなかった。

## 5.4 考察

実験結果をもとに、提案手法の性能および2段階の視線移動を検出するためのパラメータの汎用性に関して述べる。

### 5.4.1 性能

$T_{\text{dwell}}$ を306.0msとした際の提案手法における認識率およびコマンド実行に要する時間は、それぞれ82.6%および1013msであった。実験2の結果および提案手法と同様にコマンド実行の対象をとることができる手法との比較を行う。例えば、Delamareらの手法[6]において、画面中央での認識率は75.0%および時間は4.48秒(対象選択のための凝視時間:2秒、ジェスチャに要した時間:2.48秒)であった。また、Estevesらの手法[5]において、認識率は83.0%および時間は1.0秒であった。このとき、意図しないコマンドは実行されなかった。これらの研究における実験と実験設計が異なるため適切ではないが、提案手法における2段階の視線移動は高い認識率かつ短い時間にてコマンド実行ができるといえる。

また、認識率についても再調整したパラメータを用いることにより向上するか調査する必要がある。すなわち、解析において、 $T_{\text{dwell}}$ を変更した際の認識率は、 $T_{\text{dwell}}$ として219.6msを用いた実験時の視線データを用いて求めている。そのため、今後は実際に $T_{\text{dwell}}$ として306.0msを用いた際の視線データを収集し、認識率を調査する必要がある。なお、 $T_{\text{dwell}}$ を306.0msとした際に、誤った試行は画面中央において0.7%であった。このことから、コマンド

を実行しようとして視線を移動した際に、異なるコマンドが実行される可能性が低いことがわかる。

#### 5.4.2 汎用性

$T_{\text{dwell}}$  を 306.0 ms とした際の画面 4 角における認識率の平均は 67.2% であった。このことから、実験 1 にて求めたパラメータを画面上の場所によらず用いることが難しいことがわかった。この原因としては、実験 2 のアンケートから、画面外へ視線を移動する（例えば、場所が右上において方向が右 → 上である）試行が特に難しかったことが挙げられる。また、画面の端において視線認識に認識率が低下すること [24] や対象の選択が難しいこと [11] も原因である可能性がある。この課題に対して我々は、画面の端にてコマンド実行が開始される場合には、 $D_{\text{thld}}$  を小さくするといったように、パラメータを緩めるという改善方法を考えている。

## 6. おわりに

本稿にて、操作対象を一定時間見つめた後に、右を見た後に上を見るような 2 段階の視線移動を行うことによるコマンド実行手法を提案した。提案手法は、アイコンなどの GUI 部品を対象としたコマンド（例えば「あるファイルのコピー」）および画面内の GUI 部品がない場所におけるコマンド（例えば「デスクトップの切り替え」）が実行可能である。我々は、意図しない 2 段階の視線移動の認識を防ぐために、認識に用いるパラメータを決定する実験を行った。提案手法の性能を評価した結果、82.6% の認識率および 1013 ms にてコマンドが実行可能であった。先行研究と比較した結果、コマンド実行の認識率は同等もしくは高くなる可能性、およびコマンド実行に要する時間は短くなる可能性が示された。

今後は、実環境下において、意図しない 2 段階の視線移動がどの程度生じるのかを調査する。また、今回決定したパラメータを用いた場合には画面の端において 2 段階の視線移動を認識することが難しいため、パラメータを緩め画面端における 2 段階の視線移動の認識率の向上を試みる。

## 参考文献

[1] Jacob, R. J. K.: The Use of Eye Movements in Human-computer Interaction Techniques: What You Look at is What You Get, *ACM Transaction on Information Systems*, Vol. 9, No. 2, pp. 152–169.  
[2] Ware, C. and Mikaelian, H. H.: An Evaluation of an Eye Tracker As a Device for Computer Input, *CHI '87*, pp. 183–188.  
[3] Drewes, H. and Schmidt, A.: Interacting with the Computer Using Gaze Gestures, *INTERACT' 07*, pp. 475–488.  
[4] Rajanna, V. and Hammond, T.: A Gaze Gesture-based Paradigm for Situational Impairments, Accessibility, and Rich Interactions, *ETRA '18*, pp. 102:1–102:3.  
[5] Esteves, A., Velloso, E., Bulling, A. and Gellersen,

H.: Orbits: Gaze Interaction for Smart Watches using Smooth Pursuit Eye Movements, *UIST '15*, pp. 457–466.  
[6] Delamare, W., Han, T. and Irani, P.: Designing a Gaze Gesture Guiding System, *MobileHCI '17*, pp. 26:1–26:13.  
[7] Istance, H., Hyrskykari, A., Immonen, L., Mansikkamaa, S. and Vickers, S.: Designing Gaze Gestures for Gaming: An Investigation of Performance, *ETRA '10*, pp. 323–330.  
[8] Møllenbach, E., Lillholm, M., Gail, A. and Hansen, J. P.: Single Gaze Gestures, *ETRA '10*, pp. 177–180.  
[9] Vidal, M., Bulling, A. and Gellersen, H.: Pursuits: Spontaneous Interaction with Displays Based on Smooth Pursuit Eye Movement and Moving Targets, *UbiComp '13*, pp. 439–448.  
[10] Velloso, E., Wirth, M., Weichel, C., Esteves, A. and Gellersen, H.: AmbiGaze: Direct Control of Ambient Devices by Gaze, *DIS '16*, pp. 812–817.  
[11] Isomoto, T., Ando, T., Shizuki, B. and Takahashi, S.: Dwell Time Reduction Technique Using Fitts' Law for Gaze-based Target Acquisition, *ETRA '18*, pp. 26:1–26:7.  
[12] Hyrskykari, A., Istance, H. and Vickers, S.: Gaze Gestures or Dwell-based Interaction?, *ETRA '12*, pp. 229–232.  
[13] Dybdal, M. L., Agustin, J. S. and Hansen, J. P.: Gaze Input for Mobile Devices by Dwell and Gestures, *ETRA '12*, pp. 225–228.  
[14] Wobbrock, J. O., Rubinstein, J., Sawyer, M. W. and Duchowski, A. T.: Longitudinal Evaluation of Discrete Consecutive Gaze Gestures for Text Entry, *ETRA '08*, pp. 11–18.  
[15] Wobbrock, J. O., Myers, B. A. and Kembel, J. A.: EdgeWrite: A Stylus-based Text Entry Method Designed for High Accuracy and Stability of Motion, *UIST '03*, pp. 61–70.  
[16] Wobbrock, J. O., Wilson, A. D. and Li, Y.: Gestures Without Libraries, Toolkits or Training: A \$1 Recognizer for User Interface Prototypes, *UIST '07*, pp. 159–168.  
[17] Jungwirth, F., Haslgrübler, M. and Ferscha, A.: Contour-guided Gaze Gestures: Eye-based Interaction with Everyday Objects and IoT Devices, *IoT '17*, pp. 26:1–26:2.  
[18] Møllenbach, E., Hansen, J. P., Lillholm, M. and Gale, A. G.: Single Stroke Gaze Gestures, *CHI EA '09*, pp. 4555–4560.  
[19] Møllenbach, E., Hansen, J. P. and Lillholm, M.: Eye Movements in Gaze Interaction, *Journal of Eye Movement Research*, Vol. 6, No. 2.  
[20] Urbina, M. H., Lorenz, M. and Huckauf, A.: Pies with EYES: The Limits of Hierarchical Pie Menus in Gaze Control, *ETRA '10*, pp. 93–96.  
[21] Callahan, J., Hopkins, D., Weiser, M. and Shneiderman, B.: An Empirical Comparison of Pie vs. Linear Menus, *CHI '88*, pp. 95–100.  
[22] Špakov, O., Isokoski, P., Kangas, J., Akkil, D. and Majaranta, P.: PursuitAdjuster: An Exploration into the Design Space of Smooth Pursuit-based Widgets, *ETRA '16*, pp. 287–290.  
[23] Jacob, R. J. K.: Eye Movement-Based Human-Computer Interaction Techniques: Toward Non-Command Interfaces, pp. 4: 151–190.  
[24] Feit, A. M., Williams, S., Toledo, A., Paradiso, A., Kulkarni, H., Kane, S. and Morris, M. R.: Toward Everyday Gaze Input: Accuracy and Precision of Eye Tracking and Implications for Design, *CHI '17*, pp. 1118–1130.