

筑波大学大学院博士課程

システム情報工学研究科修士論文

2段階の視線移動に基づく操作手法

礒本 俊弥

修士（工学）

（コンピュータサイエンス専攻）

指導教員 志築 文太郎

2020年3月

概要

視線に基づく操作手法が普及することにより、マウスやキーボード、タッチパネルを用いた手に基づく操作手法にてコンピュータを操作できないユーザが、コンピュータを操作できるようになる。今までに視線に基づく操作手法に関する研究が多くされてきたが、依然として普及していない。これは、視線に基づく操作手法ではユーザがどこを見ているかという情報を用いて操作を行なうため、単に画面を見ているにも関わらず（つまり、ユーザの意図によらず）操作が生じる問題に直面しているためである。この意図しない操作は、手に基づく操作手法では生じず、視線に基づく操作特有の問題である。そこで、本論文において、意図しない操作が生じにくい視線に基づく操作手法を新たに示す。ユーザは、「右を見た後に上を見る」ような2段階の視線移動を行なうことにより、コピーやペーストといったコマンドを実行できる。2段階の視線移動は日常的に行われる視線移動であるため、ユーザは視線移動（つまり、コマンド実行）を容易にできると考えられるが、意図しない操作を誘発するとも考えられる。そこで、まず、3つの実験を通じてユーザが実際に行った視線移動の軌跡を収集し、ユーザがどのように視線を移動するかの特徴を見出す。そして、見出した特徴を用いて、ユーザが容易にコマンドを実行でき、意図しない操作が生じにくいシステムを実装する。実装したシステムを用いた実験から、272分間の視線に基づく操作時に意図しない操作は生じなかったこと、ユーザはコマンド実行を85.8%の精度にて成功でき、コマンド実行に要する時間は956msであることを示した。

目次

第1章	序論	1
1.1	背景	1
1.1.1	手に基づくコンピュータの操作手法	1
1.1.2	視線に基づくコンピュータの操作手法	2
1.2	本研究の目的およびアプローチ	3
1.3	本研究の貢献	3
1.4	本論文の構成	4
第2章	関連研究	5
2.1	凝視に基づく対象選択手法を用いたコマンド実行	5
2.1.1	ミダスタッチを防ぐことを目的とした手法	5
2.1.2	凝視に基づく対象選択手法を用いてコマンドを実行する際の問題点	6
2.2	ジェスチャに基づくコマンド実行手法	7
2.3	凝視に基づく対象選択とジェスチャに基づくコマンド実行を組み合わせた手法	7
2.4	関連研究と本研究の差分	9
第3章	2段階の視線移動を用いたコマンド実行手法	10
3.1	コマンド実行に用いるジェスチャ	10
3.2	コマンド実行の手順	10
3.3	2段階の視線移動の認識システムの設計指針	12
第4章	実験1：視線移動の軌跡の解析	13
4.1	実験参加者	13
4.2	実験機器および実験環境	13
4.3	実験内容	14
4.3.1	実験条件	14
4.3.2	実験手順	14
4.4	事後処理	15
4.5	解析	16
4.5.1	視線移動の方向を変える際に視線が留まった時間	19
4.5.2	視線移動時のブレ	19
第5章	2段階の視線移動の認識システムの実装	20

5.1	認識システムの概要	20
5.2	認識システムに用いるパラメータの決定	20
5.2.1	凝視時間 T_{dwell}	22
5.2.2	経路の幅 W_{path}	22
5.2.3	視線移動を認識するための閾値 D_{thld}	23
第 6 章	実験 2 : 提案手法の性能評価およびパラメータ調整	24
6.1	実験参加者	24
6.2	実験機器および実験環境	24
6.3	実験手順	26
6.4	実験結果	26
6.4.1	画面中央における性能	27
6.4.2	画面 4 角における性能評価	30
6.5	議論	30
6.5.1	パラメータ調整	32
6.5.2	パラメータの汎用性	32
第 7 章	実験 3 : 2 段階の視線移動の認識システムの改良	34
7.1	実験参加者	34
7.2	実験機器および実験環境	34
7.3	実験に用いたアプリケーション	35
7.3.1	ペイントアプリケーション	35
7.3.2	クイズアプリケーション	35
7.3.3	読書アプリケーション	37
7.3.4	タイピングアプリケーション	37
7.4	実験手順	38
7.4.1	ジェスチャアプリケーションを用いたタスク	38
7.4.2	凝視アプリケーションを用いたタスク	39
7.5	実験結果	39
7.5.1	意図しない 2 段階の視線移動の認識回数	40
7.5.2	2 段階の視線移動を用いたコマンド実行に対する印象	40
7.6	解析	41
7.6.1	パラメータを変更した際の意図しない操作の認識回数	41
7.6.2	意図しない 2 段階の視線移動に要した時間	43
7.6.3	解析のまとめ	44
7.7	認識システムの改良	45
第 8 章	実験 4 : システム改良後の提案手法の性能評価	46
8.1	実験参加者	46

8.2	実験機器および実験環境	46
8.3	実験手順	46
8.4	実験結果	47
第 9 章	議論	50
9.1	実験条件による 2 段階の視線移動の認識システムへの影響	50
9.1.1	実験参加者	50
9.1.2	実験アプリケーション	50
9.1.3	画面背景の影響	51
9.2	2 段階の視線移動の使用およびその認識システムの実装に関する議論	51
9.2.1	1 段階の視線移動の使用に関して	51
9.2.2	ジェスチャ認識システムの使用に関して	52
9.3	さらなるアプリケーション例	52
9.3.1	マーキングメニューへの応用	57
9.3.2	仮想現実空間への応用	57
9.3.3	パラメータの動的な調整	58
第 10 章	結論	59
	謝辞	60
	参考文献	61
	著者論文リスト	67

目次

3.1	提案手法において実行されるコマンド。a) あるアイコンを対象としたコマンド。画像アイコンを凝視した後に $R \rightarrow U$ の視線移動を行うことにより、画像アイコンを“開く”コマンドを実行する。b) アプリケーションやシステム全体を対象としたコマンド。アイコンが表示されていない場所を凝視した後に $U \rightarrow R$ の視線移動を行うことにより、「デスクトップの作業領域を切り替える」コマンドを実行する。	11
4.1	実験1の環境。	14
4.2	実験1に用いた画面。	14
4.3	実験1において取り除いた、各試行の指示と実際の視線移動の方向が異なる試行。	15
4.4	実験1において取り除いた、各条件の指示と異なる方向へ視線が移動していた試行。	16
4.5	実験1の条件2において収集した視線の軌跡。	17
4.6	実験1の解析方法。	18
5.1	2段階の視線移動の認識方法。	21
6.1	実験2の環境。	25
6.2	実験2に用いる画面。左：円はターゲットの位置を示し、円の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。	25
6.3	実験2における画面中央での各セッションの平均認識率。	27
6.4	実験2における画面中央での各セッションの平均時間。	28
6.5	実験2における場所毎の平均認識率。	28
6.6	実験2における場所毎の平均時間。	29
6.7	実験2の本番セッションにおける凝視時間 T_{dwell} を 220 ms–1000 ms まで 10 ms ずつ変更した際の認識率。	29
6.8	実験2の本番セッションにおける凝視時間 T_{dwell} を 219 ms、305 ms、391 ms、および 477 ms とした際の認識率。	30
6.9	実験2の本番セッションにおける凝視時間 T_{dwell} を 219 ms、305 ms、391 ms、および 477 ms とした際の時間。	30

6.10	実験2における、対象の場所と指示の方向の組み合わせごとの認識率。中央→角は画面中央における認識率である。	31
7.1	実験3の環境。	35
7.2	実験3に用いたアプリケーション。a) ペイントアプリケーション。b) クイズアプリケーション。c) およびd) 読書アプリケーション。d) タイピングアプリケーション。	36
7.3	ジェスチャアプリケーションを用いたタスク時に実施したアンケートの結果。	40
7.4	T_{dwell} ごとの凝視の認識回数と凝視が認識された際に意図しない認識が行われた確率。	41
7.5	意図しない2段階の視線移動が認識された際の視線移動の軌跡。	42
7.6	2段階の視線移動を認識に要した時間の頻度。上) 意図的に2段階の視線移動を行った際に要した時間の頻度。下) 意図しない2階の視線移動の認識が行われた際に要した時間。	44
8.1	実験4の環境。	47
8.2	2段階の視線移動を1度および2度行った際の、2段階の視線移動の認識率およびコマンド実行に要した時間。	48
8.3	2段階の視線移動を1回および2回行った際の、2段階の視線移動の方向ごとの認識率。	48
9.1	マウスを用いた操作における、マーキングメニューを表示する場合のコマンド実行の手順。	53
9.2	マウスを用いた操作における、マーキングメニューを表示しない場合のコマンド実行の手順。	54
9.3	提案手法における、マーキングメニューを表示しない場合のコマンド実行の手順。	55
9.4	提案手法における、マーキングメニューを表示する場合のコマンド実行の手順。	56

表目次

2.1	関連研究と本研究の位置づけ。	8
6.1	実験2における画面角 → 画面中央の組み合わせ。	31
6.2	実験2における画面角 → 画面外の組み合わせ。	31
7.1	実験3のタイピングアプリケーションに使用した課題文。	37
7.2	実験3のクイズアプリケーションに使用した問題文およびその回答。	38
7.3	実験3のジェスチャタスクにおける、2段階の視線移動の方向に対応するコマンド。	38
7.4	改良後の2段階の視線移動の認識システムに用いたパラメータ。	45

第1章 序論

本章では、本研究の背景と目的、および本論文の構成を述べる。

1.1 背景

ユーザの視線を認識することが可能な機器（以降、視線認識機器）から収集できる視線にまつわる情報のみを用いたコンピュータの操作手法（以降、視線に基づく操作手法）は、マウスとキーボードを組み合わせたコンピュータの操作手法、タッチパネルを用いたコンピュータの操作手法に続く、第3のコンピュータの操作手法となるだろう。

マウスとキーボードを組み合わせた操作、およびタッチパネルを用いた操作はいずれもユーザ自身の手に基づく操作である。そのため、上肢不自由者や手を操作に使うことができない状況におけるユーザは、これらの操作を行うことができない。そこで、ユーザ自身の手を用いずにコンピュータの操作が可能である、視線に基づく操作手法が多く研究されてきた。しかし、依然として、視線に基づく操作手法はコンピュータの操作手法として用いられていない。これは、視線に基づく操作には、手に基づく操作において生じない問題があるためである。なお、視線に基づく操作および手に基づく操作を組み合わせた操作手法も研究されている（例 [SD12, CXH15, TABG15, PG16]）が、本研究では、視線に基づく操作手法のみを用いたコンピュータの操作手法を研究対象とする。

本節では、まず、手に基づく操作手法において、どのような操作が行われているかを述べた後に、これらの操作が、視線に基づく操作手法ではどのように行われるかを述べる。そして、視線に基づく操作手法の問題点を述べる。

1.1.1 手に基づくコンピュータの操作手法

現在、コンピュータの操作には主にユーザ自身の手が用いられている。例えば、コンピュータの操作は、ユーザが手にてマウスもしくはタッチパネルを用いてカーソル操作すること、およびキーボードを用いた文字入力により操作している。また、スマートフォンの操作は、ユーザが手にてタッチパネルを直接接触れることにより行われている。これらの操作を用いて、コンピュータの操作、特に、ある対象を選択する「対象選択」およびコピーやペーストを行う「コマンド実行」は以下のように行われる。

マウスを用いた操作において対象選択は、マウスを移動することによりマウスポインタを移動し、選択したい対象の上にマウスポインタが乗っている状態にて、左クリックを行うこ

とにより可能である。また、タッチパネルを用いた操作においては、選択したい対象が表示されている場所を触れることにより可能である。コマンド実行は、インタフェース（アイコンやメニューなどの画面表示）への対象選択を複数回行うことにより可能である。例えばマウスを用いた操作においては、まず、コマンドを実行したい対象を右クリックする。その後に表示されるメニューの中から実行したいコマンドを選択することにより、そのコマンドを実行可能である。また、キーボードを用いた操作では例えば、‘Control’ キーと ‘C’ キーを押すとコピーが実行され、‘Control’ キーと ‘V’ キーを押すとコピーした対象のペーストが可能である。このように、事前に決まっている複数のキーを押すことにより、コマンドを実行することも可能である。さらに、ある特定の軌跡のようにマウスポインタを動かすこと、もしくは、ある特定の軌跡を指にて描くことにより、その軌跡（ジェスチャ）に紐づくコマンドを実行することも可能である。このジェスチャを用いてコマンドを実行する手法をジェスチャに基づく操作手法と呼ぶ。

1.1.2 視線に基づくコンピュータの操作手法

手に基づく操作手法に対して、視線に基づく操作手法では、ユーザの視線情報のみを操作に用いられる。視線情報は、ユーザが画面のどこを見ているかを示す視線座標およびそのタイムスタンプからなる。この視線情報を用いて、対象選択はユーザが対象を見続ける（対象の中に視線座標が一定時間入る）ことにより行われる。ユーザが対象を一定時間以上見続けることを凝視と呼び、この対象選択手法は、凝視に基づく対象選択手法と呼ばれる [Jac90, Jac91]。なお、凝視に基づく対象選択手法において、ユーザが対象を見続ける必要のある時間は凝視時間と呼ばれる。マウスを用いた操作と同様に、視線に基づく操作手法においても、複数回の対象選択を行うことによりコマンドを実行できる。具体的には、まずコマンドを実行したい対象を凝視し選択する。その後、表示されるメニューへ視線を移し、実行したいコマンドを凝視することにより実行可能である。また、視線に基づく操作手法においてもジェスチャに基づく操作手法を用いてコマンドを実行することも可能である（例 [ULH10, DHI17]）。

手に基づく操作手法と視線に基づく操作手法の大きな違いとして、意図しない操作の発生の有無がある。手に基づく操作では、ユーザが左クリックを行う、もしくはタッチパネルに触れるといった動作を行わない限り操作は行われない。対して、視線に基づく操作では、例えばユーザが画面をただ見ていたとしても、凝視が認識され、操作（つまり、対象選択）が行われることがある。このように、ユーザの意図とは別に凝視が認識され、意図しない対象選択が行われることはミダスタッチ問題 [Jac90] と呼ばれ、凝視に基づく操作手法において大きな問題とされている [JS03, IBHV08]。また、ジェスチャに基づく操作手法においても、同様な問題が生じる。例えば、ある特定の動きとして、視線を左から右へ移動するようなジェスチャがあったとする。このジェスチャは、ユーザが文章を読む際に生じ、文章を読む際に視線が左から右へ移動すると、ユーザは文章を読んでいるにも関わらず、システムがその視線移動を操作を行うための視線移動であると認識し、ジェスチャに紐づくコマンドが実行されてしまう。このように、視線に基づく操作手法を提案する際は、意図しない操作をどのように防ぐかを考慮する必要がある。

意図しない操作を防ぐことを目的とした研究はこれまでに多く行われている。凝視に基づく操作手法においては、長い凝視時間を用いることにより意図しない操作を防ぐことができるが、対象選択に要する時間が長くなり、ユーザビリティが低下する。そこで、短い凝視時間においても、意図しない操作を防ぐことを目的とした手法が提案されてきた（例 [BVH12, PLW12, NDA⁺17, MWW17, IAST18]）。しかし、これらの研究において提案されている手法は、その用途が制限されているという課題がある。同様に、ジェスチャに基づく操作においても、意図しない操作を防ぐことを目的とした手法が提案されてきた（例 [DS07, DHI17]）。これらの研究では、日常的には生じにくい複雑なジェスチャをコマンドに紐付けることにより、ユーザが意図的に視線を移動させたときのみ、その視線移動をジェスチャとして認識するようにしている。

1.2 本研究の目的およびアプローチ

本研究の目的は、視線に基づく操作の中でも特に「コマンド実行」に着目し、意図しない操作が生じにくいコマンド実行手法を示すことである。「コマンド実行」の観点では、ジェスチャに基づく操作手法は凝視に基づく操作手法に比べ、意図しない操作に対して堅牢であり、素早くコマンド実行が可能とされている [HIV12, DAH12]。そのため、ジェスチャに基づく操作手法に関して提案する。

本研究において提案する手法は、2段階の視線移動に基づくコマンド実行手法である。提案手法において用いるジェスチャは、例えば「右を見て上を見る」のような2段階（‘L’字）の視線移動である。これは、先行研究に用いられている視線移動と比べ単純である。この単純な視線移動において意図しない操作を防ぐために、本研究では、実際にユーザに視線移動を行ってもらい、その視線移動の軌跡を観察することにより、2段階の視線移動の認識システムを実装する。その後、さらにユーザの視線移動の軌跡を収集し、意図しない操作がより生じにくく、かつ意図した操作を実行しやすくなるように認識システムを改良する。そして、改良した認識システムを用いた場合、ユーザはどの程度の精度にてコマンドが実行できるか、およびコマンドを実行する際に要する時間を示す。

1.3 本研究の貢献

本研究の貢献を以下に列挙する。

- 2段階の視線移動を用いたコマンド実行手法の提案
- ユーザが視線を移動する際に、どのように視線を移動するかの調査
- 調査した視線移動の軌跡を用いた、2段階の視線移動の認識システムの設計および実装
- 実装した認識システムを用いた場合の、2段階の視線移動を用いたコマンド実行手法の性能評価

1.4 本論文の構成

第1章において、本論文の背景および研究目的を述べる。第2章では、関連研究に対する本研究の位置付けを述べる。第3章では、提案手法である2段階の視線移動に基づくコマンド実行手法に関して述べる。第4章では、2段階の視線移動をどのように認識するかを決めるための実験に関して述べ、第5章において、実験結果を元に、2段階の視線移動の認識システムを実装する。第6章、第7章では、第5章にて述べた認識システムを改良することを目的とした実験を行う。そして、改良したシステムを用いた場合の2段階の視線移動に基づくコマンド実行手法の性能評価を第8章にて述べる。第9章では、本研究に関する議論を示し、第10章にて、本研究のまとめを述べる。

第2章 関連研究

本研究は、視線に基づく操作において、意図しない操作が生じにくいコマンド実行手法を提案することを目的としている。本研究と関連している研究として、視線に基づく操作において、「凝視に基づく対象選択手法を用いたコマンド実行」および「ジェスチャに基づくコマンド実行手法」が挙げられる。本章にて、これらの研究に関して述べた後に、提案手法の位置づけを示す。

2.1 凝視に基づく対象選択手法を用いたコマンド実行

本節では、まず凝視に基づく対象選択手法において意図しない操作（つまり、ミダスタッチ）を防ぐことを目的とした研究を述べる。その後、凝視に基づく対象選択手法を用いてコマンドを実行する際の課題を述べる。

2.1.1 ミダスタッチを防ぐことを目的とした手法

凝視に基づく操作手法は、操作の簡単さおよび学習が容易であることから多くの研究がされており、コマンドを実行することも容易であると考えられるが、ミダスタッチ問題に直面している。これまでに、ミダスタッチ問題の解決を目的とした研究は多く行われてきている。これらの研究は、凝視時間を対象の種類や特徴ごとに変化させることにより、ミダスタッチの発生を防いでいる。Penkerら [PLW12] は、アイコンの大きさによって凝視時間を変更することにより、ミダスタッチの発生を防いでいる。具体的には、大きなアイコンには視線が入りやすいため、凝視時間を長くし、視線が入り続けたとしても凝視が認識されないようにしている。また、Nayyarら [NDA⁺17] は、ユーザの慣れやボタンの種類に応じて凝視時間を変更することにより、ミダスタッチの発生を防いでいる。さらに、Isomotoら [IAST18] は、ユーザの視線移動がフィッツの法則 [Fit54]（ユーザが対象選択を行う際の行動モデル）に従っている場合にのみ、対象選択をすることにより、ミダスタッチの発生を防いでいる。また、ソフトウェアキーボードを用いたタイピング（選択したいキーを凝視することにより、そのキーを選択可能）時のミダスタッチの発生を防ぐことを目的とした研究も行われている。Majarataら [MAv09] は、操作しやすくミダスタッチが生じにくい凝視時間をユーザが自ら選択できるインタフェースを構築した。加えて、Panwarら [PSS12] は、文字の出現頻度を考慮して設計した専用のインタフェースを構築した。

さらに、ミダスタッチが生じにくい手法として、ある移動するアイコンを見ているとき、そのアイコンの動きにつられて視線が動くこと（円滑性追跡眼球運動）を用いた手法も研究されている（例 [VBG13, EVBG15, SDRD17]）。円滑性追跡眼球運動は動くアイコンを意図的に見ている時のみ検出されるため、ミダスタッチを始めとする意図しない操作に対して堅牢である。

これらの研究では、ミダスタッチの発生を減らすことが可能であると報告されているが、依然としてミダスタッチは発生している。また、ミダスタッチの発生を防ぐためには、専用のインタフェースを必要とするという制限もある。例えば、Isomotoら [IAST18] は、アイコンが隣接する場合にはミダスタッチを防ぐことが難しく、アイコン間には距離を置く必要があると述べている。円滑性追跡眼球運動を用いた手法では、画面上に動くアイコンを配置する必要がある。

2.1.2 凝視に基づく対象選択手法を用いてコマンドを実行する際の問題点

凝視に基づく対象選択手法を用いたコマンド実行には、プルダウンメニューやポップアップメニュー [Shn86]、パイメニュー [CHWS88] のようなインタフェースが用いられる。これは、手に基づく操作と同じようにコマンドを実行できるため、習熟度の観点では有効だと思われるが、凝視に基づく操作はミダスタッチ問題に直面しているため、メニューを用いてコマンドを実行するには不向きである。プルダウンメニューやポップアップメニュー、パイメニューを用いてコマンドを実行するには、1章にて述べたように、少なくとも2回の対象選択を行う必要がある。そのため、ミダスタッチ問題に直面している凝視に基づく対象選択手法では、意図しない対象へのコマンド実行や、意図しないコマンドの実行が生じる可能性がある。さらに、ミダスタッチが発生することにより、メニューの表示が頻繁に行われてしまう可能性もある。例えば、ミダスタッチを防ぐために凝視時間を長くした場合は、対象選択を少なくとも2回行う必要があるためコマンド実行に要する時間が大きくなる。

実際に、凝視に基づく対象選択手法を用いてコマンドを実行する場合の性能評価もされている。Kammererら [KSB08] は、プルダウンメニューおよびパイメニューを用いてコマンドを実行するには、両メニューともに4s以上の時間を要し、正確にコマンドを実行できた確率は90%であったことを報告している。なお、この調査に用いられた凝視時間は750msであった。さらに、凝視に基づく対象選択手法を用いた場合と他の視線に基づく操作手法の比較も行われており、円滑性追跡眼球運動を用いた手法と比べて、性能は同等か優れていると報告されている [vIK+16] が、コマンド実行に要する時間および正確にコマンドを実行できる確率ともに、ジェスチャに基づく操作手法が優れているという結果が報告されている [HIV12, DAH12]。これらの結果より、視線に基づく操作手法において、コマンド実行はジェスチャに基づく手法が適していると言える。

2.2 ジェスチャに基づくコマンド実行手法

ジェスチャに基づくコマンド実行手法においてユーザは、実行したいコマンドと紐付けられているジェスチャの通りに視線移動を行うことによりコマンドを実行できる。Drewesら [DS07] は、画面4角を通るような視線移動をジェスチャに用いた。例えば、画面左上 → 右上 → 右下 → 左下 → 左上を通るような時計回りの視線移動や、画面左上と右上を交互に3回ずつ見るような視線移動をジェスチャとして用いている。この時計回りの視線移動は、4段階の視線移動といえる。Rozadoら [RMSA⁺15] は、スマートフォンの画面4角を通るような3段階の視線移動をジェスチャとして用いている。またJungwirthら [JHF17] は、例えば家にあるランプのような実物体の縁に沿った視線移動をジェスチャとして用い、ジェスチャに「ランプを点灯する」というコマンドを紐付けた。Rajannaら [RH18] は、1段階の視線移動（例えば、右 → 左への視線移動）および‘U’字の視線移動からなる9種類の視線移動をジェスチャとして用いている。Istanceら [IHI⁺10] は、ジェスチャを行うための半透明な領域を画面中央に表示した。ユーザはその領域を見ながら視線を移動することができるためコマンド実行が容易になる。Wobbrockら [WRSD07, WRSD08] は、EdgeWrite [WMK03] を視線に基づく操作手法に適応し、ジェスチャを用いた文字入力手法を提案した。EdgeWriteは、表示されている追加のウィンドウにペンを用いてジェスチャを描くことにより、そのジェスチャに紐づく文字が入力される手法である。Møllenbachら [MHLG09, MLGH10, MHL13] は、画面左端から画面右端へ視線が移動するような1段階の視線移動をジェスチャとして用いた。

これらの手法には、コマンド実行の対象を選択することができないという課題がある。コマンド実行の対象を選択することは、マウスを用いた操作では対象への右クリックにより行われ、タッチパネルを用いた操作ではロングタップ（長い時間触れ続けること）により行われている。そのため、実行可能なコマンドは、例えば、Webブラウザにおける「前のページに戻る」や、「デスクトップの作業領域を切り替える」といったコマンドのみとなる。つまり、ある対象を選択し、その対象を「コピーする」といったコマンドは実行できない。また、単にコマンド実行手法を提案することが目的であれば、1段階の視線移動を用いることがコマンド実行の容易さの観点において有効であるが、1段階の視線移動は多く生じやすい視線移動である [DS07, RMSA⁺15] ため、意図しない操作を防ぐという観点においては不向きである。そのため、意図しない操作を防ぐことも目的としている研究では、3段階以上の視線移動が用いられている [DS07, RMSA⁺15]。

2.3 凝視に基づく対象選択とジェスチャに基づくコマンド実行を組み合わせた手法

凝視に基づく対象選択手法とジェスチャに基づくコマンド実行手法を組み合わせた手法も研究されている。この手法では、まず凝視を行うことにより、コマンド実行の対象を選択し、その後、実行したいコマンドが紐付けられているジェスチャの通りに視線移動を行うことによりコマンドを実行できる。凝視に基づく対象選択が行われた際に、コマンドを実行するた

表 2.1: 関連研究と本研究の位置づけ。

研究	対象選択	コマンド実行の方法	ガイド
[DS07]、[RMSA ⁺ 15]	不可能	3段階以上の視線移動	非表示
[JHF17]	不可能	実物体の縁に沿った視線移動	非表示
[MHLG09]、[MLGH10] [MHL13]、[RH18]、	不可能	1段階の視線移動	非表示
[IHI ⁺ 10]	不可能	2もしくは3段階の視線移動	常に表示（半透明）
[WRSD07] [WRSD08]	不可能	3段階以上の視線移動	常に表示
[ULH10]	可能	パイメニュー	表示（注視後）
[DHI17]	可能	円滑性追跡眼球運動	表示（凝視後）
[WM87]	可能	1段階の視線移動	表示（注視後）
提案手法	可能	2段階の視線移動	非表示

めのメニューや専用のアイコン（以降、ガイド）が表示され、ガイドを参考にユーザは視線移動を行うことができる。Urbinaら [ULH10] は、パイメニューをジェスチャに基づく操作に適応した。ガイド（ここではパイメニュー）は注視（凝視時間が短い凝視）が認識された後に表示され、パイメニューの縁を視線が横断すると、該当するコマンドが実行される。パイメニューの階層によって、視線移動の段数が決まり、2階層のパイメニューの場合、ジェスチャは2段階の視線移動と同様なものとなる。Delamareら [DHI17] は、凝視が認識された後に、円滑性追跡眼球運動を促すアイコンや、コマンドが紐付けられたジェスチャの軌跡をガイドとして表示した。Wareら [WM87] は、スクリーンボタンという操作手法を提案した。これは、凝視による選択を、2回の注視と1回の視線移動により行うものである。具体的には、選択したい対象へ視線を移し、その後常に表示されているボタンへ視線を移動し（つまり、1段階の視線移動を行い）、ボタンを注視することにより、その対象を選択することができる。ユーザは、スクリーンボタンを用いて、1段階の視線移動を2度行うことにより、対象の選択およびコマンドを実行することができる。

このように、凝視に基づく対象選択手法とジェスチャに基づくコマンド実行手法を組み合わせることにより、コマンド実行の対象を選択することもできる。しかし、凝視に基づく対象選択手法は依然として、ミダタッチ問題に直面しているため、凝視とジェスチャを組み合わせた手法においても課題がある。ミダタッチが生じることにより、意図しない対象へのコマンドが実行されてしまうことに加え、意図せずガイドが表示されてしまう。実際に、Delamareら [DHI17] はミダタッチの発生を防ぐために、凝視時間を2sとしている。さらに、意図しない対象が選択（ガイドが表示）された際は対象以外のアイコンを1s凝視し、コマンド実行を中止する（ガイドを隠す）必要がある。なお、コマンド実行の対象の選択に注視を用いている研究 [WM87, ULH10] においては、ミダタッチ問題に関して考慮されていない。

2.4 関連研究と本研究の差分

本研究では、視線に基づく操作において、意図しない操作が生じないコマンド実行手法を提案する。表 2.1 に、関連研究と本研究の位置づけを示す。意図しない操作に言及している先行研究 [DS07, RMSA⁺15] において用いられているジェスチャは、3 段階以上の視線移動であり、複雑なものが多い。より簡易な 1 段階の視線移動を用いる研究 [MHLG09, MLGH10, MHL13, RH18] では、コマンド実行を容易に行うことができるため有効であると考えられるが、日常的な視線の移動としても生じるため [DS07, RMSA⁺15]、意図しない操作を防ぐという観点においては不向きである。そこで、提案手法では 2 段階の視線移動を用いて、意図しない操作が生じないコマンド実行手法を提案する。

さらに、先行研究では、コマンド実行の対象を選択するために、凝視に基づく対象選択手法を用いている。しかし、ミダスタッチ問題が以前として解決されていないため、ガイドの表示が意図せず生じることがある。ジェスチャとなる視線移動を円滑性追跡眼球運動により促す研究 [DHI17] では、移動するアイコンの表示は必須となるため、意図しないガイドの表示は問題となる。そのため、ミダスタッチが生じない凝視時間として、2s の長い凝視時間が用いられている。提案手法においても、コマンド実行の対象を選択するために、凝視に基づく対象選択とジェスチャに基づくコマンド実行手法を組み合わせる。また、常にガイドを表示する必要のある研究 [WRSD07, WRSD08, IHI⁺10] では、コマンドを実行しない際にガイドの表示を消すことができない。提案手法は、2 段階の視線移動をジェスチャとして用い、このジェスチャは簡易な視線移動であるためガイドの表示を行わずとも、ユーザはコマンドの実行が可能である。そのため、凝視時間として比較的短い時間を用いることができる。また、ミダスタッチにより、意図しない対象が選択されたとしても、その対象にコマンドが実行されないようにシステムを実装する。このシステムの実装は、複数回の実験を通じて収集するユーザが視線を移動する際の軌跡を用いて行なう。

第3章 2段階の視線移動を用いたコマンド実行手法

本章において、どのようなジェスチャがコマンド実行に用いられるか述べ、実際にコマンド実行がどのように行われるかを説明する。その後、2段階の視線移動を認識するシステムをどのように設計する必要があるかを説明する。

3.1 コマンド実行に用いるジェスチャ

提案手法は、2段階の視線移動をジェスチャとして用いたコマンド実行手法である。2段階の視線移動とは、まず水平方向もしくは垂直方向に視線を移動させ（1段階目の視線移動）、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段階目の視線移動）ような視線移動である。つまり、提案手法において用いるジェスチャは、右を見た後に上（以降、 $R \rightarrow U$ ）、上を見た後に右（以降、 $U \rightarrow R$ ）、左を見た後に上（以降、 $L \rightarrow U$ ）、上を見た後に左（以降、 $U \rightarrow L$ ）、右を見た後に下（以降、 $R \rightarrow D$ ）、下を見た後に右（以降、 $D \rightarrow R$ ）、左を見た後に下（以降、 $L \rightarrow D$ ）や、下を見た後に左（以降、 $D \rightarrow L$ ）の8ジェスチャである。

3.2 コマンド実行の手順

提案手法においてコマンド実行は以下のように行われる。ユーザはまず、画面内のある点を一定時間（凝視時間）以上見つめる。その後、実行したいコマンドが紐付けられているジェスチャの通りに視線を移動する。このとき、凝視した点が、あるファイルアイコンの中に存在している場合は、そのアイコンがコマンド実行の対象となり、例えば、そのアイコンを「コピーする」といったコマンドが実行できる（図3.1a）。つまり、画面上にある複数のアイコンの中から、1つのアイコンを選択しコマンドを実行できる。また、凝視した点が画面に表示されているアイコンの外である場合は、コマンド実行の対象はアプリケーションやシステム全体となり、例えば、Webブラウザにおける「前のページに戻る」や、「デスクトップの作業領域を切り替える」といったコマンドを実行できる（図3.1b）。ユーザが自らの意思にて凝視する場所を変えることによりコマンド実行の対象を決めることが可能な点は、先行研究と比べ本研究の特色でもある。

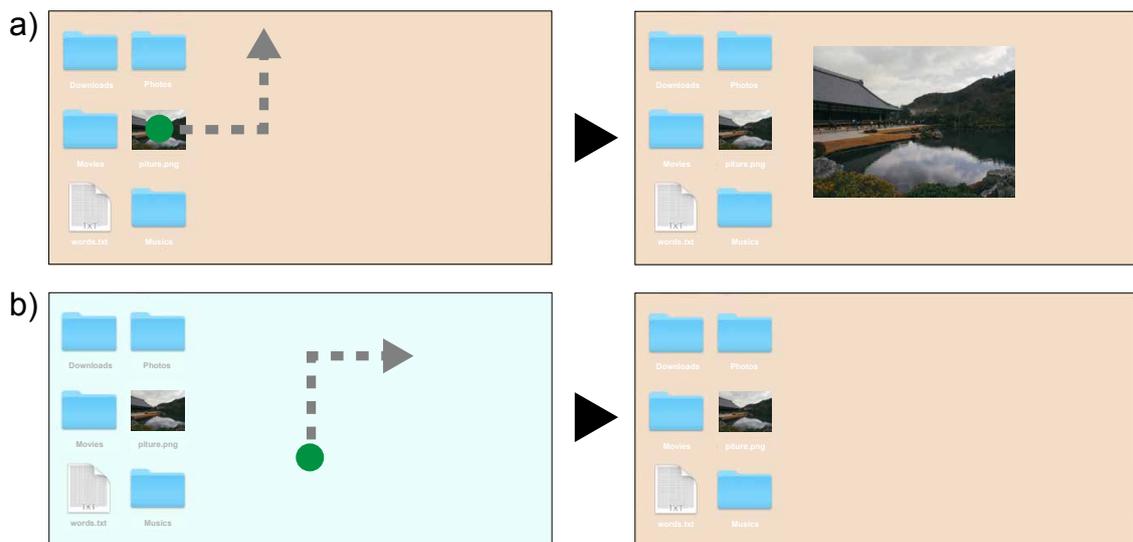


図 3.1: 提案手法において実行されるコマンド。a) あるアイコンを対象としたコマンド。画像アイコンを凝視した後に $R \rightarrow U$ の視線移動を行うことにより、画像アイコンを“開く”コマンドを実行する。b) アプリケーションやシステム全体を対象としたコマンド。アイコンが表示されていない場所を凝視した後に $U \rightarrow R$ の視線移動を行うことにより、「デスクトップの作業領域を切り替える」コマンドを実行する。

3.3 2段階の視線移動の認識システムの設計指針

提案手法を実現するために必要な、2段階の視線移動の認識システムを設計する際に考慮すべきことを示す。

2段階の視線移動は日常的に起きることが想定される。例えば、文章を読む際は視線を左から右へ移動する、その後、視線を下もしくは上へ移動した場合は2段階の視線移動となる。また、凝視の基づく選択手法を用いて“win”をQWERTYキーボードにて入力する際、‘w’を凝視した後、‘i’へ視線を移し凝視を行い、‘n’へ視線を移し凝視する必要がある。このとき、‘w’から‘i’は右への視線移動であり、‘i’から‘n’は下への視線移動であるため、2段階の視線移動となる。これらのように日常的に生じるとされる2段階の視線移動を認識せず、ユーザが意図的にコマンド実行をしようとした際のみ、視線移動を認識するようなシステムを設計する必要がある。そのため、本研究では、ユーザが視線を移動する際の軌跡を収集し、その軌跡を用いて認識システムを設計する

第4章 実験1：視線移動の軌跡の解析

実験1の目的は、ユーザが実際に視線移動を行う際に、どのように視線を移動させるかを調査することである。そのために実験1では、提案手法において意図しない操作として認識される1段階の視線移動、操作を行うために用いる2段階の視線移動を実験参加者に行うよう指示する。そして、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情報（視線座標およびそのタイムスタンプ）を収集し、その特徴を見出す。

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名が眼鏡、2名がコンタクトを着用していた。11名が以前に同様な実験において視線認識機器を使用したことがあったが、いずれも日常的には使用していなかった。また、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名ずつの群（A群およびB群）に分けた。

4.2 実験機器および実験環境

実験1は、図4.1に示すような環境にて実施された。視線認識機器としてTobii EyeX¹を用い、ディスプレイ下部に取り付け使用した。ディスプレイには、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（DELL社製G2410。画面サイズ：24インチ、解像度：1,920×1,080ピクセル）を用い、高さを変更可能なディスプレイアームに取り付けた。実験に関係ない物体が実験参加者の視界に入らないように、ディスプレイは白い壁の前に配置した。また、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、頭部固定機を用いて実験参加者の頭部位置を固定した。視線認識機器が正しく実験参加者の視線を認識できるよう、ディスプレイから頭部固定機までの距離を約60cmとし、ディスプレイの高さは実験参加者に応じて変更した。さらに、実験中の部屋の明かりを、人工的な照明による明かりのみとすることにより一定に保った。

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



図 4.1: 実験 1 の環境。

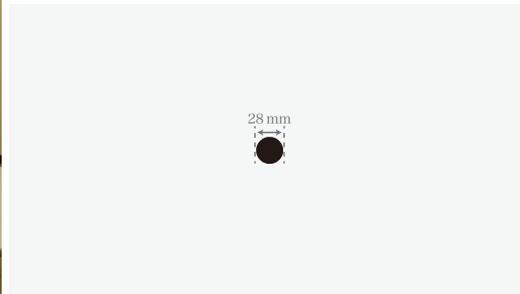


図 4.2: 実験 1 に用いた画面。

4.3 実験内容

実験 1 の条件および手順に関して述べる。

4.3.1 実験条件

実験 1 は、実験参加者に 1 段階の視線移動を促す条件 1 および 2 段階の視線移動を促す条件 2 の 2 条件からなる。条件 1 では、実験参加者に「(ある方向) を見てください」と指示する。この(ある方向)には UR 、 UL 、 DR 、および DL の 4 方向が該当する。つまり、条件 1 において UR 方向を指示する際は、「 UR を見てください」のような指示になる。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」と指示する。(1 段階目の方向を見て 2 段階目の方向) には、 $R \rightarrow U$ 、 $L \rightarrow U$ 、 $R \rightarrow D$ 、 $L \rightarrow D$ 、 $U \rightarrow R$ 、 $U \rightarrow L$ 、 $D \rightarrow R$ 、および $D \rightarrow L$ の 8 方向が該当する。つまり、条件 2 において $R \rightarrow U$ 方向を指示する際は、「右を見て上を見てください」のような指示になる。

なお、条件 2 において 1 段階目の方向から 2 段階目の方向へ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。さらに、各条件において、視線移動を行う際に画面上のどこまでを見るかまでは指示しなかった。また、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、視線認識の結果に関する情報を実験参加者に通知しなかった [Jac93]。条件間の順序効果を相殺するために、A 群は条件 1 から実験を開始し、B 群は条件 2 から実験を開始した。

4.3.2 実験手順

各条件を開始する前に視線認識機器のキャリブレーション（視線認識機器の認識精度をユーザがごとに適切にすること）を行った。実験 1 の各条件において、1 度指示に従い視線を移

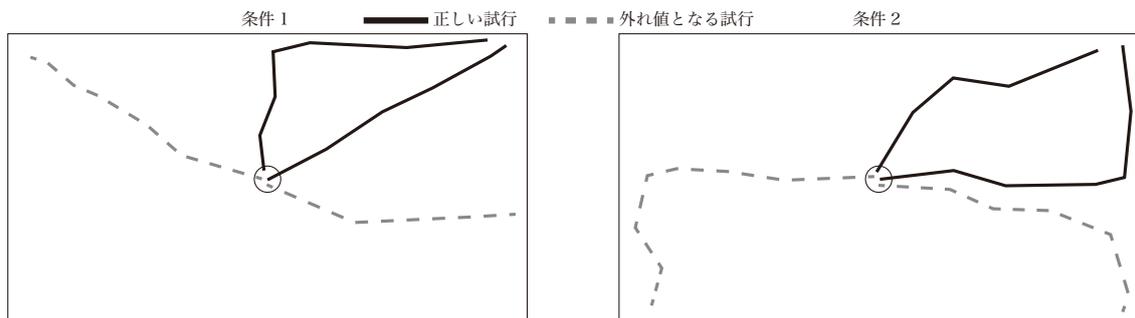


図 4.3: 実験 1 において取り除いた、各試行の指示と実際の視線移動の方向が異なる試行。

動させることを 1 試行とした。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円 (図 4.2 の黒塗りの円) を見るよう指示した。その後、キーボードのキーを押下すると音声にて再生される各条件における指示に従い視線移動を行うように伝えた。1 度目のキーの押下が行われると、試行が開始し、システムは視線移動の軌跡の収集を開始する。実験参加者には、試行を終了したとき (指示通りの視線移動を行った後) に再度キーを押すように指示した。2 度目のキーの押下が行われると、試行が終了し、システムは視線移動の軌跡の収集を終了する。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。

各条件の各指示それぞれ実験参加者 1 人あたり 16 試行ずつ行った。つまり実験 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。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平均 52 分であった。実験終了後、実験参加者は謝金として 5,000 円を受け取った。

4.4 事後処理

実験を通じて収集した視線移動の軌跡をもとに、ユーザはどのように視線を移動するか解析する前に、事前処理として外れ値となる試行を取り除いた。

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。ローパスフィルタを適用した後の各試行における視線移動の軌跡に対して、以下の処理を行った。

各試行において指示と実際の視線移動の方向が異なる試行を取り除いた。例えば、条件 1

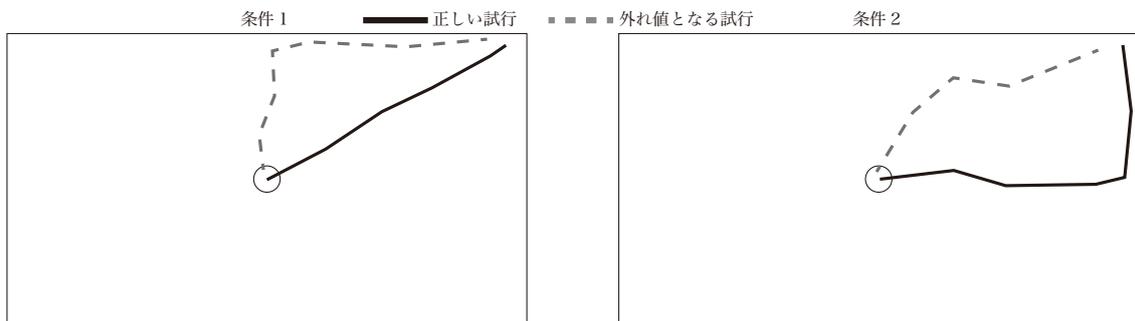


図 4.4: 実験 1 において取り除いた、各条件の指示と異なる方向へ視線が移動していた試行。

において、指示が UR 方向へ視線を移動する試行では、視線移動の方向が第 2-4 象限の方向である試行を外れ値として取り除く (図 4.3 左)。条件 2 においては、指示が $R \rightarrow U$ 方向へ視線を移動する試行では、1 段階目の視線移動の方向が下方向である、もしくは 2 段階目の視線移動の方向が左か下方向である試行を外れ値として取り除く (図 4.3 右)。結果として、条件 1 では 3.7%、条件 2 では 9.8% が外れ値として取り除かれた。

その後、各条件において指示と異なる方向へ視線が移動していた試行を取り除いた。例えば、条件 1 において、指示が UR 方向へ視線を移動する試行では、 $R \rightarrow U$ や $U \rightarrow R$ のように視線が移動していた試行を取り除く (図 4.4 左)。条件 2 においては、指示が $R \rightarrow U$ 方向へ視線を移動する試行では、 UR のように視線が移動していた試行を取り除く (図 4.4 右)。結果として、条件 1 では 1.9%、条件 2 では 2.0% が外れ値として取り除かれた。

これらの結果から、以降の解析には事後処理を通じて外れ値を取り除いた試行 (以降、正解試行) を用いる。つまり正解試行は、条件 1 において 967 ($1,024 \times (1 - (0.037 + 0.019))$) 試行、条件 2 において 1,806 ($2,048 \times (1 - (0.098 + 0.02))$) 試行となる。特に、条件 2 において、外れ値となった試行が 11.8% と大きくなった理由としては、実験参加者から「視線を移動させる方向を間違えた」とコメントを得ていたことから、実験タスクが簡単故に生じたと考えた。

4.5 解析

各条件・各指示ごとの正解試行における視線移動の軌跡を図 4.5 に示す。条件 2 の正解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、 $R \rightarrow U$ 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 (視線が移動時にブレている) ことである。

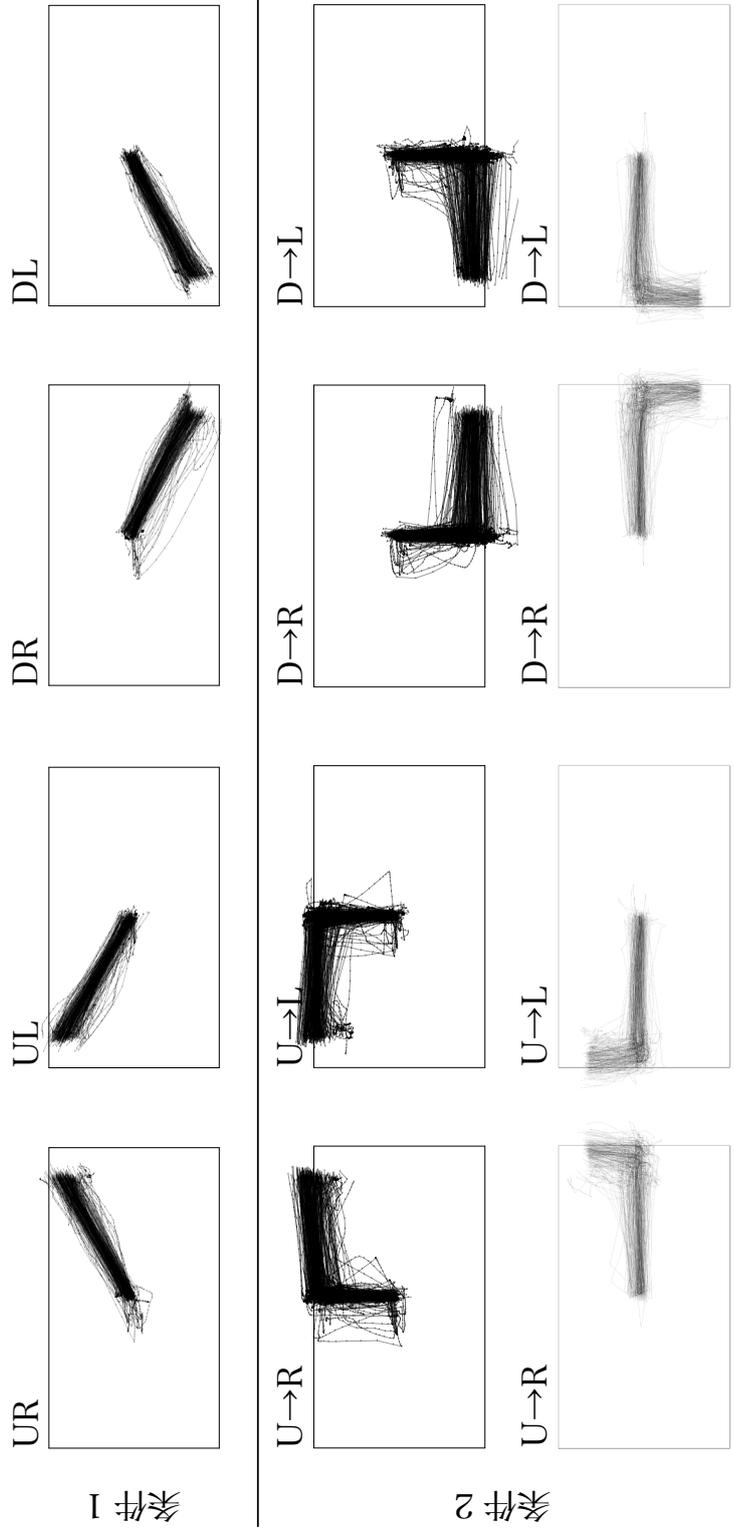


図 4.5: 実験 1 の条件 2 において収集した視線の軌跡。

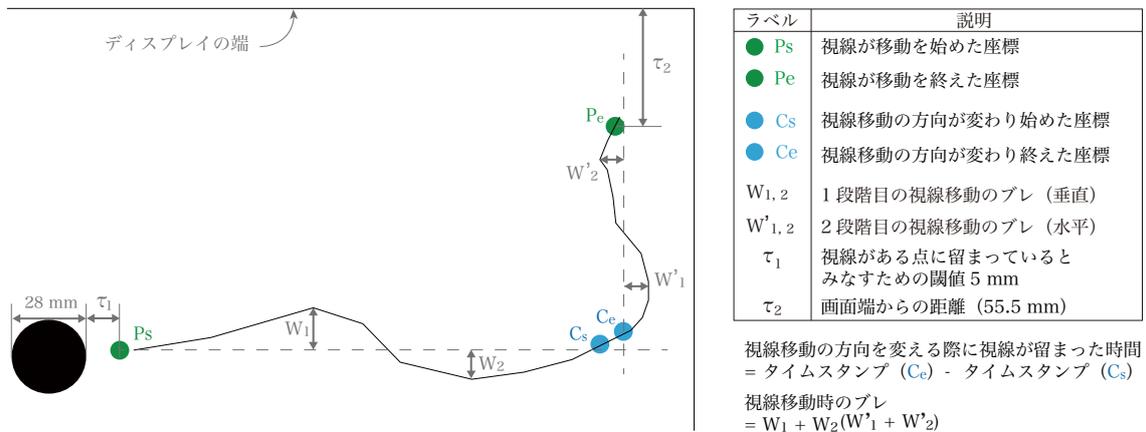


図 4.6: 実験 1 の解析方法。

これら 2 つの特徴を定量化するために、条件 2 の各正解試行の視線移動に対して、視線が移動を始めた座標 (P_s)、1 段階目の方向から 2 段階目の方向へ視線移動の方向が変わり始めた座標 (C_s)、1 段階目の方向から 2 段階目の方向へ視線移動の方向が変わり終えた座標 (C_e)、および視線が移動を終えた点 (P_e) の 4 つの座標を検出する (図 4.6)。これら 4 点の検出方法は以下の通りである。

P_s 1 段階目の方向が右の場合、視線が図 4.6 に示す黒色の円 (画面表示した黒い円と同じく直径 28 mm) から外に出ていて、かつ d_x (1 つ前に収集された視線の x 座標から現在の視線の x 座標までの距離) が閾値 τ_1 より大きくなった座標。1 段階目の方向が上の場合は、 d_x ではなく d_y (1 つ前の視線の y 座標からの距離) が $-\tau_1$ より小さくなった座標。

C_s P_s が検出された後に d_x が τ_1 より小さくなった座標。 d_x が τ_1 より大きくなった場合は C_s の検出を再度行う。

C_e 視線座標が C_s から垂直方向に τ_1 より移動し、 d_y が d_x より大きい座標。

P_e C_e が検出された後に、x および y 座標が画面右上端から τ_2 以内に移動した座標。

この検出方法において、ある点を見ている際の視線座標のばらつきは 5 mm であること [FWT+17] から、視線がある点に留まっているとみなすための閾値として τ_1 を 5 mm とした。また、 τ_2 には 55.5 mm を採用した。これは、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、全正解試行において P_e を決定できるように τ_2 を 55.5 mm とした。

4.5.1 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、各正解試行において C_e と C_s 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 ms ($SD = 86$) であった。なお、条件 1 においては視線移動の方向が変わらず視線は留まっていない。

4.5.2 視線移動時のブレ

視線移動時のブレを、各試行において P_s-C_s 間および C_e-P_e 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もとに決定した。例えば図 4.6 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値 (W_1) と下方向へのブレの最大 (W_2) 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値 (W'_1) と左方向へのブレの最大値 (W'_2) 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_s-C_s において 17.0 mm ($SD = 25.9$) であり、 C_e-P_e 間において 17.8 mm ($SD = 24.1$) であった。

第5章 2段階の視線移動の認識システムの実装

本章において、意図した2段階の視線移動の認識システムに関して述べる。はじめに、認識システムの概要を述べ、認識システムの実装に必要なパラメータを示す。そして、実験1において収集した視線移動の軌跡、およびその特徴から決定したパラメータに関して述べる。

5.1 認識システムの概要

提案手法における2段階の視線移動の認識方法を以下に示す(図5.1)。

- a) ある点に対して凝視が認識された時、凝視点を中心に幅 W_{path} の1段階目の経路を生成する。提案手法において、凝視は視線が5 mm (4.5節と同様に τ_1) 以内に T_{dwell} 間留まると認識される。
- b) 1段階目の視線移動は、水平(垂直)方向に凝視点からある閾値 D_{thld} 以上、視線が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。
- c) 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(P)を中心に幅 W_{path} の2段階目の経路が生成される。
- d) 2段階目の視線移動は、垂直(水平)方向に閾値 D_{thld} 以上 P から移動した場合認識される。2段階目の視線移動が認識されると、その方向に対応したコマンドが実行される。この時、凝視した点がある対象の中に存在する場合はその対象へコマンドが実行され、対象がない場合はアプリケーションやシステム全体へのコマンドが実行される。

なお、この認識システムでは経路の表示は行なわれない。この認識システムを実装するために、 T_{dwell} (凝視時間)、 W_{path} (1段階目および2段階目の経路の幅) および D_{thld} (視線移動を認識するために視線が移動すべき距離の閾値) の3つのパラメータを用いる。

5.2 認識システムに用いるパラメータの決定

認識システムに用いるパラメータは大きすぎず、小さすぎない値とする必要がある。本節では、それぞれのパラメータをなぜこのような値とする必要があるか、およびその決定方法に関して述べる。

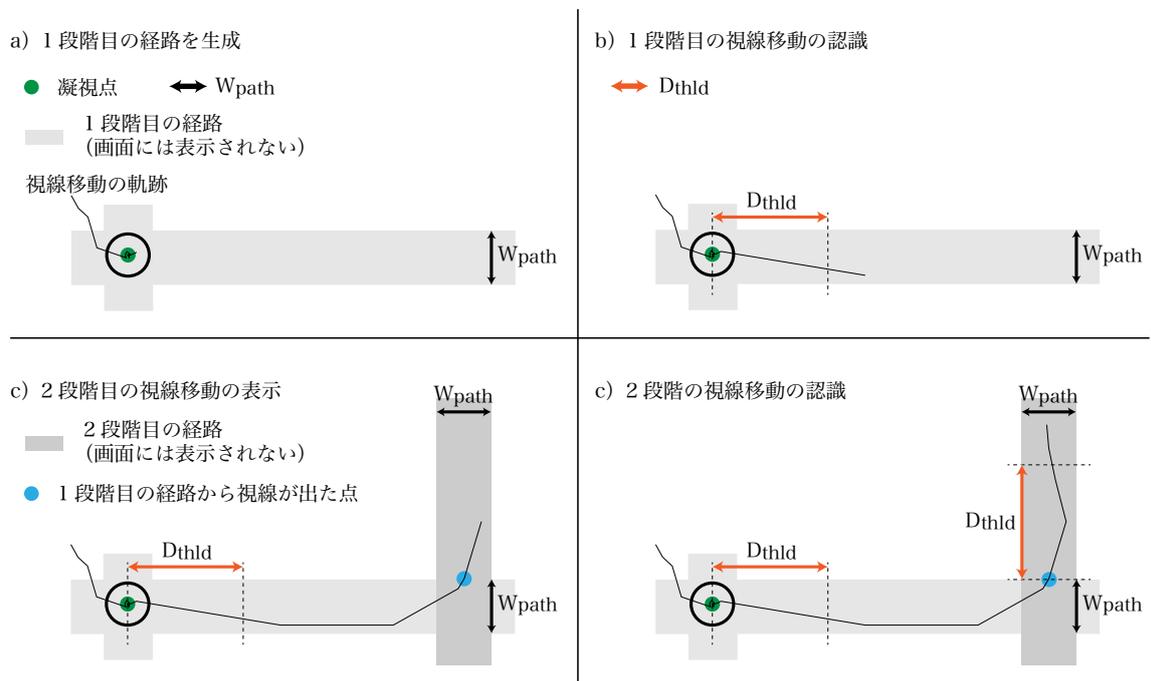


図 5.1: 2段階の視線移動の認識方法。

5.2.1 凝視時間 T_{dwell}

凝視が認識されると視線移動の認識を開始する提案手法では、凝視の認識が意図しない操作を防ぐという点においても寄与している。提案手法において、 T_{dwell} を小さくすると意図しない操作を防ぐことができる。 T_{dwell} が小さい場合は、凝視の認識が行われやすくなる。結果として、視線移動中に凝視が認識されることにより、意図的に2段階の視線移動を行っていないことを識別することができる。しかし、4.5.1節にて述べたように、1段階目の視線移動から2段階目の視線移動へ方向を変えるときに視線が留まることから、ある程度 T_{dwell} は大きくする必要がある。また、 T_{dwell} が大きすぎる場合は、凝視の認識が行われにくくなる。凝視が認識されにくいということは、凝視が認識されると視線移動の認識を開始する提案手法において、ユーザビリティの低下につながる。さらに、視線移動中に視線が留まるのが許容されるため、意図しない操作が生じやすくなる。例えば、あるアイコンを見ようとして右へ視線を移動し、アイコンを見た後に上へ視線を移動すると2段階の視線移動が認識される。しかし、ユーザはコマンド実行を行うためではなく、アイコンを見るために1段階目の視線移動をしているため、意図的に2段階の視線移動を行っていない。これらの理由より、 T_{dwell} は大きすぎず、小さすぎない適切な値とする必要がある。

そこで、実験1において収集した視線移動の軌跡をもとに、 T_{dwell} を決定した。具体的に、 T_{dwell} は4.5.1節にて述べた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。つまり、 T_{dwell} には219 (133+86) msを採用した。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パーセントイル値である。219 msは凝視に基づく操作において、ミダタッチが発生しやすいとされる短い凝視時間である。提案手法では、ミダタッチが発生したとしても、2段階の視線移動が認識されない限り、意図しない操作は生じないため、短い凝視時間を用いることができる。

5.2.2 経路の幅 W_{path}

W_{path} が小さい場合、ユーザは意図した操作を行うために直線的に視線を移動する必要がある。これにより、直線的に視線が移動していない場合は、ユーザは意図的に2段階の視線移動を行っていないとし、意図しない操作を防ぐことができる。しかし、4.5.2節にて述べたように、視線移動を行う際には直線的ではなくある程度ブレながら視線が移動することがわかっている。さらに、操作を行いやすくするためにも、ある程度 W_{path} を大きくする必要がある。

そこで、実験1において収集した視線移動の軌跡をもとに、 W_{path} を決定した。具体的に、 W_{path} は4.5.2節にて述べた視線移動時のブレの平均と標準偏差の和とした。1段階目の視線移動 (P_s-C_s 間) および2段階目の視線移動 (C_e-P_e 間) 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 (17.0+25.9) mm および41.9 (17.8+24.1) mm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、1段階目と2段階目の視線移動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} とした。これは、条件2における視線移動時のブレの93.2パーセントイル値である。

5.2.3 視線移動を認識するための閾値 D_{thld}

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。 D_{thld} が小さい場合は、視線の移動量が少なくても、1段階目および2段階目の視線移動が認識されるため、意図しない操作が生じやすくなる。対して、 D_{thld} が大きい場合は、意図しない操作の発生を防ぐことができる。しかし、移動量が増えるため、視線移動の認識が開始された点（凝視が認識された点）によっては、画面の大きさの制限により操作を実行できないことがある。そのため、 D_{thld} も適切な大きさとする必要がある。

D_{thld} も W_{path} と同様に、実験1の結果から定量的に決定することを試みたが、図4.5に示すように多くの視線移動が画面の端まで移動していたため、できなかった。そこで、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した。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% と低いが、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。

第6章 実験2：提案手法の性能評価およびパラメータ調整

実験2の目的は、5章にて述べた2段階の視線移動の認識システムを用いた際の提案手法の性能を評価し、より高い性能とするためにパラメータを調整することである。求める提案手法の性能は、画面中央と4角における意図した操作が正しく認識される確率、およびコマンド実行に要する時間である。実験2では、視線認識機器から収集される視線情報に対して実験1と同じローパスフィルタを適用した。パラメータは5.2節にて決定したパラメータを用いた。つまり、 T_{dwell} は 219 ms、 W_{path} は 42.9 mm、および D_{thld} は 55.8 mm である。

6.1 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 名が以前に同様な実験において視線認識機器を使用したことがあったが、いずれも日常的には使用していなかった。

6.2 実験機器および実験環境

実験2は、図 6.1 に示すような環境にて実施された。視線認識機器として Tobii EyeX¹ を用い、ディスプレイ下部に取り付け使用した。ディスプレイには、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（DELL 社製 G2410。画面サイズ：24 インチ、解像度：1,920 × 1,080 ピクセル）を用い、高さを変更可能なディスプレイアームに取り付けた。実験に関係ない物体が実験参加者の視界に入らないように、ディスプレイは白い壁の前に配置した。視線認識機器が正しく実験参加者の視線を認識できるよう、ディスプレイから頭部固定機までの距離を約 60 cm とし、ディスプレイの高さは実験参加者に応じて変更した。さらに、実験中の部屋の明かりを、人工的な照明による明かりのみとすることにより一定に保った。

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



図 6.1: 実験 2 の環境。

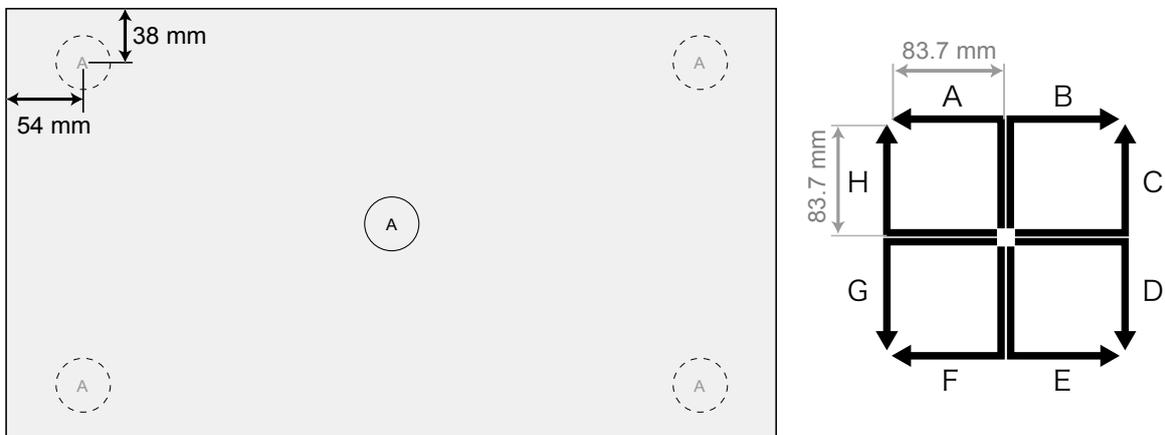


図 6.2: 実験 2 に用いる画面。左：円はターゲットの位置を示し、円の中の文字は視線移動の方向を示す。右：円の中の文字に対応する方向を示す。

6.3 実験手順

実験2には図6.2に示す2つの画面を用いた。図6.2左に示すように、コマンド実行の対象を左下 (Bottom-Left: BL)、左上 (Top-Left: TL)、右下 (Bottom-Right: BR)、右上 (Top-Right: TR)、および中央 (Center: C) の5箇所いずれかに表示した。また、図6.2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。使用する視線認識機器は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。

実験2では、実験参加者が219 ms (T_{dwell}) 間対象を凝視した後に、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、および説明の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’キー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’キーを押すよう指示した。‘Space’キーが押されると図6.2右が画面に表示され、再度‘Space’キー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’キー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音声にて通知するとともに、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)かを確認できる。その後、‘Enter’キーを押すと休憩に状態が遷移する。

40 (8方向 × 5箇所) 試行行うことを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、実験監督者が説明中になぜ認識された(されなかった)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。また、実験1と同様に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。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。実験終了後、実験参加者は謝金として5,000円を受け取った。

6.4 実験結果

本節では、画面中央および画面4角における提案手法の性能に関して述べる。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した時間である。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た。また、コマンド実行に要した時間は対象を凝視し始める時間から2段階の

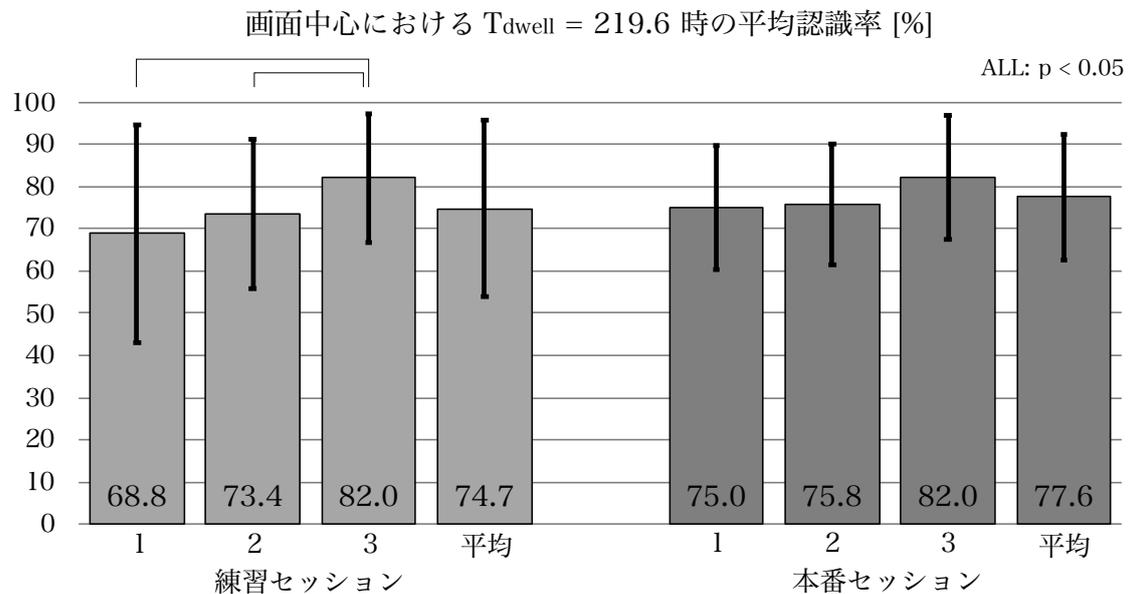


図 6.3: 実験 2 における画面中央での各セッションの平均認識率。

視線移動が認識されるまでの時間である。以降、検定にはウィルコクソンの符号順位検定を用いた ($\alpha = 0.05$)。

6.4.1 画面中央における性能

各セッションの認識率の平均を図 6.3 に示す。認識率の平均は、練習セッションの第 1 セッションにおいて 68.8% ($SD = 25.8$)、第 2 セッションにおいて 73.4% ($SD = 17.6$)、第 3 セッションにおいて 82.0% ($SD = 15.2$) であり、練習セッション全体では 74.7% ($SD = 20.1$) であった。本番セッションの第 1 セッションにおいて 75.0% ($SD = 14.7$)、第 2 セッションにおいて 75.8% ($SD = 14.3$)、第 3 セッションにおいて 82.0% ($SD = 14.6$) であり、本番セッション全体では 77.6% ($SD = 14.9$) であった。検定の結果、練習セッションの 1-3 および 2-3 セッション間にのみ有意差があった。なお、指示とは異なる方向の視線移動が認識された確率は練習セッションにおいて 1.5% であり、本番セッションにおいて 0.5% であった。

各セッションの時間の平均を図 6.4 に示す。認識率の平均は、練習セッションの第 1 セッションにおいて 824 ms ($SD = 132$)、第 2 セッションにおいて 840 ms ($SD = 107$)、第 3 セッションにおいて 812 ms ($SD = 100$) であり、練習セッション全体では 825 ms ($SD = 114$) であった。本番セッションの第 1 セッションにおいて 812 ms ($SD = 122$)、第 2 セッションにおいて 812 ms ($SD = 83$)、第 3 セッションにおいて 791 ms ($SD = 87$) であり、本番セッション全体では 810 ms ($SD = 100$) であった。検定の結果、練習セッションおよび本番セッションの各セッション間に有意差がなかった。

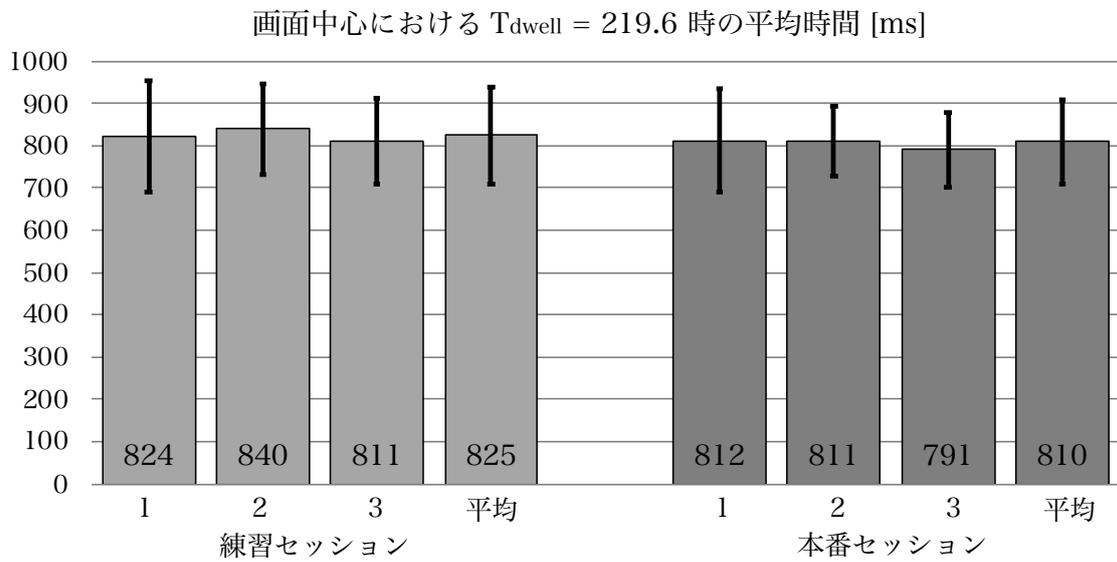


図 6.4: 実験 2 における画面中央での各セッションの平均時間。

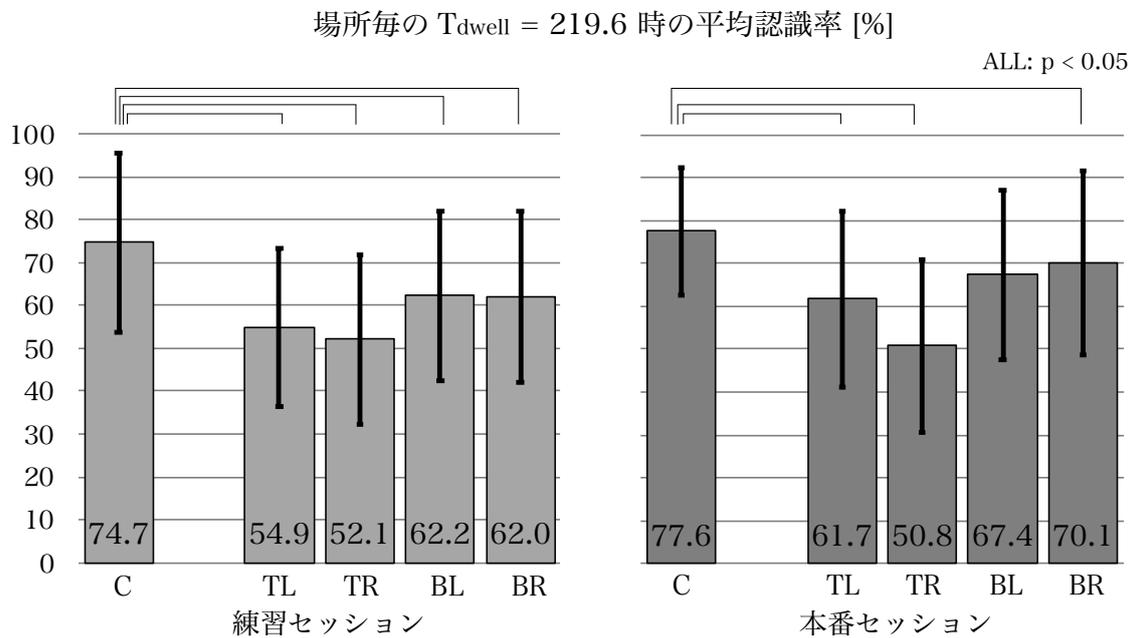


図 6.5: 実験 2 における場所毎の平均認識率。

場所毎の $T_{\text{dwell}} = 219.6$ 時の平均時間 [ms]

ALL: $p < 0.05$

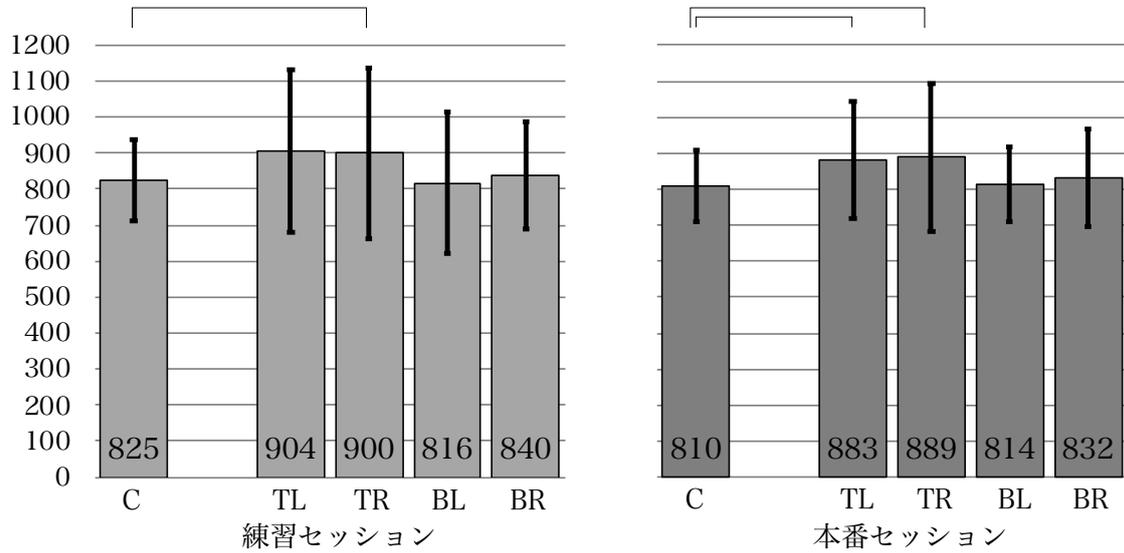


図 6.6: 実験 2 における場所毎の平均時間。

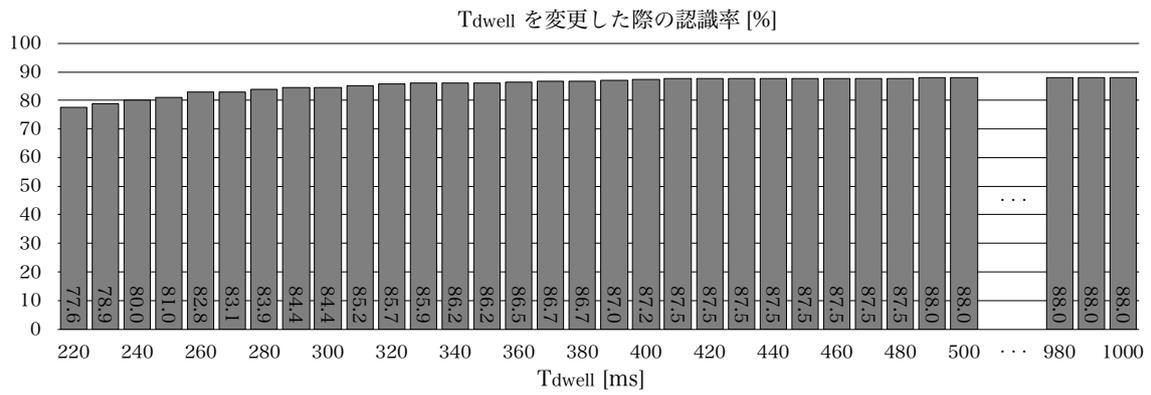


図 6.7: 実験 2 の本番セッションにおける凝視時間 T_{dwell} を 220 ms–1000 ms まで 10 ms ずつ変更した際の認識率。

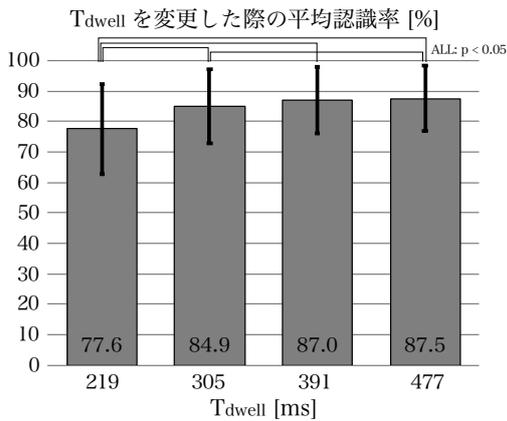


図 6.8: 実験 2 の本番セッションにおける凝視時間 T_{dwell} を 219 ms、305 ms、391 ms、および 477 ms とした際の認識率。

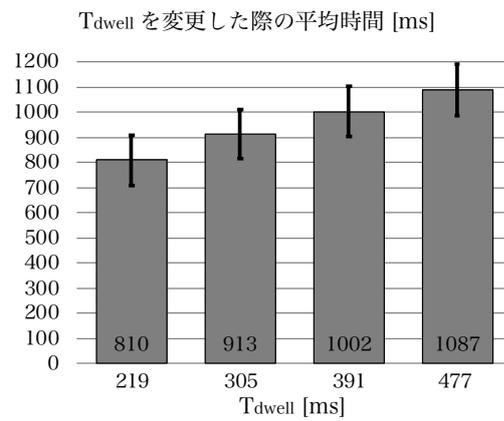


図 6.9: 実験 2 の本番セッションにおける凝視時間 T_{dwell} を 219 ms、305 ms、391 ms、および 477 ms とした際の時間。

6.4.2 画面 4 角における性能評価

画面 4 角 (TL、TR、BL、および BR) における認識率の平均は、練習セッションにおいて 57.8% ($SD = 13.2$)、本番セッションにおいて 62.5% ($SD = 15.3$) であった。また、画面 TL、TR、BL、および BR 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% ($SD = 19.7$)、54.9% ($SD = 18.4$)、62.0% ($SD = 19.9$)、および 62.2% ($SD = 21.9$) であった (図 6.5)。本番セッションの認識率の平均は、それぞれ 50.8% ($SD = 20.1$)、61.7% ($SD = 20.5$)、70.1% ($SD = 21.3$)、および 67.4% ($SD = 19.7$) であった。検定の結果、本番セッションの BL-C 間にはのみ有意差がなかった。

画面 4 角における時間の平均は、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 868 ms ($SD = 91$)、本番セッションにおいて 855 ms ($SD = 92$) であった。また、画面 TL、TR、BL、および BR における練習セッションの時間の平均は、それぞれ 904 ms ($SD = 226$)、900 ms ($SD = 237$)、816 ms ($SD = 197$)、および 840 ms ($SD = 149$) であった (図 6.6)。本番セッションの時間の平均は、それぞれ 883 ms ($SD = 162$)、889 ms ($SD = 206$)、814 ms ($SD = 105$)、および 832 ms ($SD = 137$) であった。検定の結果、練習セッションの C-TR 間、本番セッションの C-TL および C-TR 間に有意差があった。

6.5 議論

実験結果をもとに、パラメータの調整を行う。その後、5 章にて決定し本節において調整するパラメータの汎用性に関して議論する。

表 6.1: 実験 2 における画面角 → 画面中央の組み合わせ。

場所	方向
BL	R→U、U→R
BR	L→U、U→L
TL	R→D、D→R
TR	L→D、D→L

表 6.2: 実験 2 における画面角 → 画面外の組み合わせ。

場所	方向
BL	L→U、U→L、R→D、D→R、L→D、D→L
BR	R→U、U→R、R→D、D→R、L→D、D→L
TL	R→U、U→R、L→U、U→L、L→D、D→L
TR	R→U、U→R、L→U、U→L、R→D、D→R

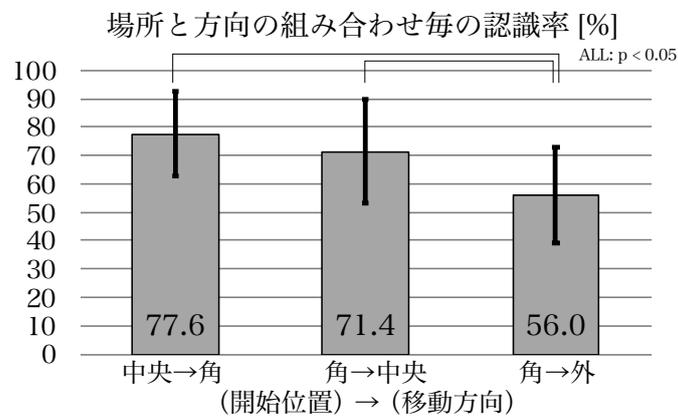


図 6.10: 実験 2 における、対象の場所と指示の方向の組み合わせごとの認識率。中央 → 角は画面中央における認識率である。

6.5.1 パラメータ調整

パラメータを調整することにより、2段階の視線移動の認識率（画面中央の本番セッションにおいて77.6%）をより高くする。認識率が100%にならなかった原因として、219 msとした T_{dwell} が影響していることを2つの理由から発見した。1つは、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繁に指示していたことであり、もう1つは、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である。そのため、より大きな T_{dwell} を用いた際に、どの程度認識率が向上するかを調査する。なお、 W_{path} および D_{thld} に関する回答はなかった。この調査は、パラメータを変更した2段階の視線移動の認識システムに、実験を通じて収集した試行時の視線の軌跡を入力したときの認識率をシミュレートすることにより行う。

T_{dwell} の影響を調査するために、 T_{dwell} を 220 ms から 1000 ms まで 10 ms ずつ増加させながら、本番セッションにおける2段階の視線移動の認識率をシミュレートした。シミュレーションの結果を図 6.7 に示す。認識率は T_{dwell} の増加に伴い向上し、 T_{dwell} が 490 ms の時に認識率が 88.0% となり上限が見られた。5.2.1 節にて述べたように、 T_{dwell} は大きすぎないほうが好ましい。そこで、4.5.1 節の視線が留まった時間の平均（133 ms）とそれぞれ 1SD（= 86）、2SD、3SD、および 4SD の和、つまり、219 ms、305 ms、391 ms、477 ms を T_{dwell} とした際の、認識率もシミュレートした。なお、これらは、4.5.1 節の視線が留まった時間の 85.9、95.5、98.6、および 99.5 パーセンタイル値である。シミュレーションの結果を図 6.8 に示す。認識率はそれぞれ 77.6%（ $SD = 14.9$ ）、84.9%（ $SD = 12.2$ ）、87.0%（ $SD = 10.8$ ）、87.5%（ $SD = 10.8$ ）検定の結果、305–391 ms 間、305–477 ms 間、および 391–477 ms 間に有意差が見られなかった。また、219 ms、305 ms、391 ms、477 ms を T_{dwell} とした際のコマンド実行に要した時間を図 6.9 に示す。時間はそれぞれ、810 ms（ $SD = 100$ ）、913 ms（ $SD = 97$ ）、1002 ms（ $SD = 101$ ）、および 1087 ms（ $SD = 101$ ）であった。コマンド実行に要する時間は T_{dwell} を含むため、検定を行っていない。なお、指示とは異なる方向の視線移動が認識された確率は本番セッションにおいて 0.5% であり、 T_{dwell} を大きくする前から変わらなかった。これらの結果から、305 ms を認識率および意図しない操作を防ぐこと両方の観点から適切な T_{dwell} として扱う。

6.5.2 パラメータの汎用性

画面中央に比べ画面4角の認識率は低く、画面中央にて決定したパラメータを画面4角において用いることは難しいと考えられる。この原因として、画面中央に比べ画面端は視線を移動しにくいということがわかった。5名の実験参加者は「画面角から画面の外に向かって視線を移動することは難しかった。」とアンケートに回答した。さらに、11名の実験参加者は「画面角から画面中央へ向かって視線を移動することは、ディスプレイの縁に沿って視線を移動することができたため容易だった。」とアンケートに回答した。そこで、画面角から画面中央（角 → 中央）および、画面角から画面外（角 → 外）へ視線が移動するような場所と方向の組み合わせ（それぞれ表 6.1 および表 6.2）における認識率を調査した。結果を図 6.10 に示す。本番セッションの角 → 中央における認識率は 71.4%（ $SD = 18.2$ ）であり、角 → 外

における認識率は 56.0% ($SD = 17.0$) であった。検定の結果、角 → 外と中央 → 角および角 → 中央の間に有意差が見られた。この結果から、5 章において画面中央での視線移動を元に決定した認識方法は、画面 4 角から画面中央に向かうような視線移動の認識には利用可能であると考えられる。

第7章 実験3：2段階の視線移動の認識システムの改良

実験3の目的は、意図しない操作を防ぐためにはどのようにシステムを改良する必要があるか、および実際に視線に基づく操作を用いて複数アプリケーションの操作を行った際の印象を調査することである。システム改良のための調査は、凝視に基づく対象選択手法を用いてアプリケーションの操作および、2段階の視線移動に基づくコマンド実行手法を用いてアプリケーションの操作をするときの視線移動の軌跡をもとに行う。印象の調査は実際に2段階の視線移動を行った際のアンケート結果をもとに行う。そして、これら結果をもとに2段階の視線移動の認識システムの改良を行う。

実験3では、視線認識機器から収集される視線情報に対して実験1および2と同じローパスフィルタを適用した。実験3において使用するパラメータは5.2節にて決定したパラメータを用いた。つまり、 T_{dwell} は 305 ms、 W_{path} は 42.9 mm、および D_{thld} は 55.8 mm である。

7.1 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 21–25 歳（平均 22.5 歳）であり、全員日本人の男性であった。3 名が眼鏡を着用していて、2 名がコンタクトを着用していた。13 名が以前に実験参加者として視線認識機器を使用したことがあり、2 名はゲームを体験したことがあったが、いずれも日常的には使用していなかった。実験参加者 16 名の内 8 名が実験 2 に参加していた。この 8 名を C 群とし、残りの 8 名を D 群とした。

7.2 実験機器および実験環境

実験3は、図7.1に示すような環境にて実施された。視線認識機器として Tobii Eye Tracker 4C¹ を用い、ディスプレイ下部に取り付け使用した。ディスプレイには、画面における背景などの反射を防ぐためにノングレタタイプの画面（DELL 社製 G2410。画面サイズ：24 インチ、解像度：1,920×1,080 ピクセル）を用い、高さを変更可能なディスプレイアームに取り付けた。実験に関係ない物体が実験参加者の視界に入らないように、ディスプレイは白い壁の前に配置した。視線認識機器が正しく実験参加者の視線を認識できるよう、ディスプレイから頭部固定

¹<https://gaming.tobii.com/tobii-eye-tracker-4c/>
最終閲覧 2020 年 1 月 15 日

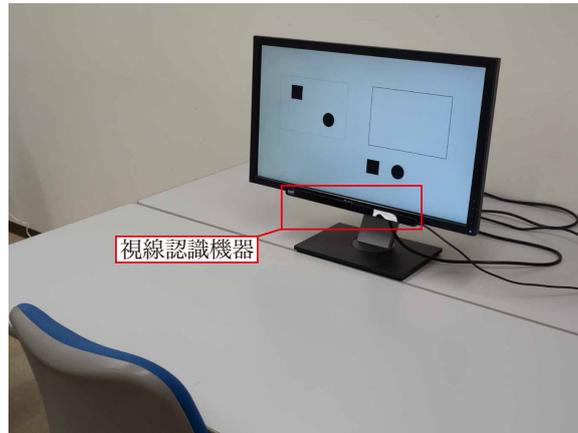


図 7.1: 実験 3 の環境。

機までの距離を約 60 cm とし、ディスプレイの高さは実験参加者に応じて変更した。さらに、実験中の部屋の明かりを、人工的な照明による明かりのみとすることにより一定に保った。

7.3 実験に用いたアプリケーション

図 7.2 に実験 3 にて用いたアプリケーションを示す。実験に用いたアプリケーションは、ペイントアプリケーション、クイズアプリケーション、読書アプリケーション、およびタイピングアプリケーションである。ペイントおよびクイズアプリケーションは、2 段階の視線移動に基づくコマンド実行を用いて操作が行われ、以降、併せてジェスチャアプリケーションと呼ぶ。また、読書およびタイピングアプリケーションは、凝視に基づく対象選択を用いて操作が行われ、以降、併せて凝視アプリケーションと呼ぶ。実験 3 における凝視に基づく対象選択の凝視時間は 1.0 s とした。

7.3.1 ペイントアプリケーション

図 7.2a にペイントアプリケーションの画面を示す。実験参加者には、画面右に表示されている対象（四角および円）のコピーを行い、画面左に配置されている対象と同じ位置にペーストするよう指示した。実験参加者は、ペイントアプリケーションのタスクを実施するためには各コマンドを 2 度ずつ（四角および円の対象へのコピー・ペースト）行うことになる。

7.3.2 クイズアプリケーション

図 7.2b にクイズアプリケーションの画面を示す。実験参加者には、画面に表示されている文章問題に回答するよう指示した。問題は正・誤にて回答可能な文章問題 4 問であり、回答

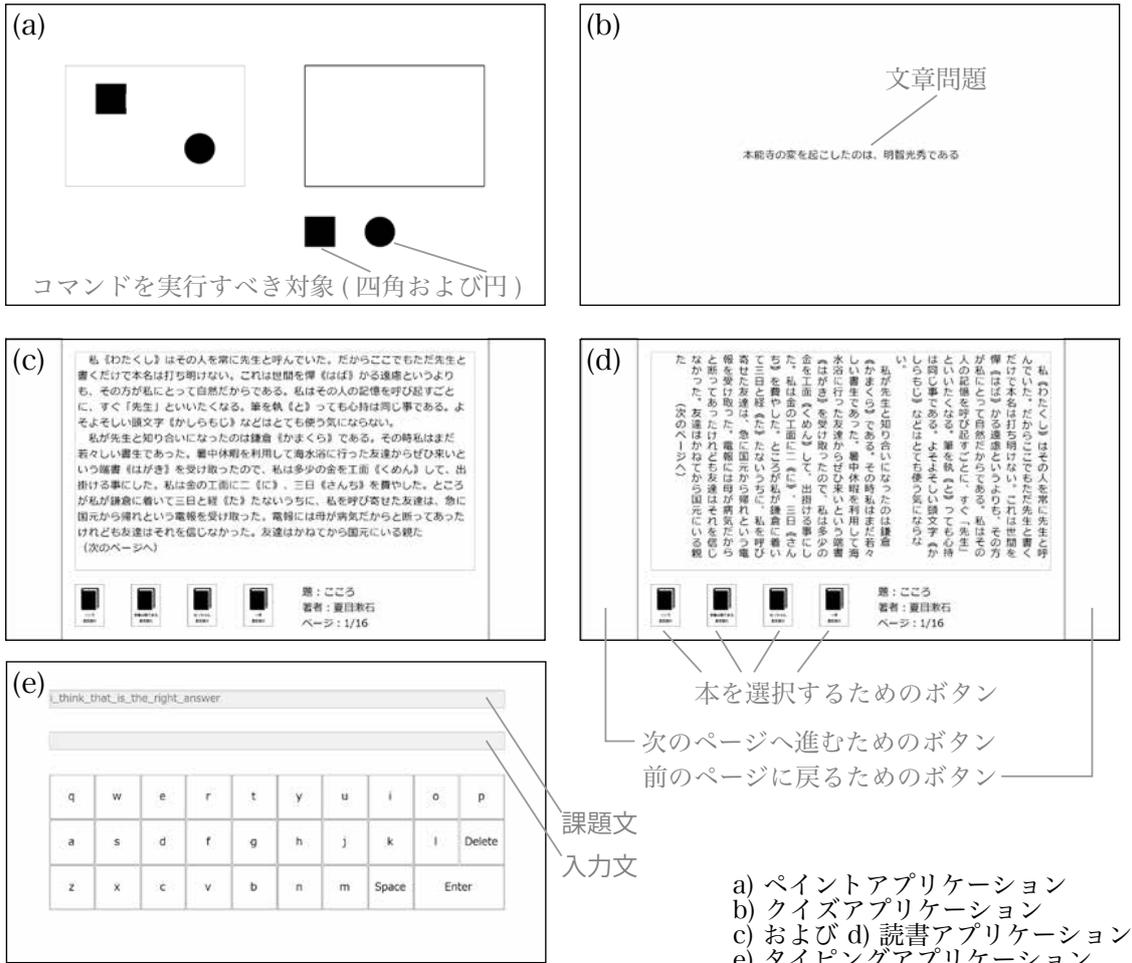


図 7.2: 実験 3 に用いたアプリケーション。a) ペイントアプリケーション。b) クイズアプリケーション。c) および d) 読書アプリケーション。d) タイピングアプリケーション。

表 7.1: 実験 3 のタイピングアプリケーションに使用した課題文。

課題文	文字数 (スペース含む)
i_think_that_is_the_right_answer	32
i_wanted_to_go_drinking_with_you	32
we_need_to_talk_about_this_month	32
please_call_tomorrow_if_possible	32
they_are_more_efficiently_pooled	32

は正・誤それぞれ 2 度ずつ出現する。実際に使用した問題文を表 7.2 に示す。なお、実験参加者には回答が正・誤 2 回ずつ出現するとは伝えていない。実験参加者は、クイズアプリケーションのタスクを実施するために各コマンドを 2 度ずつ行うことになる。

7.3.3 読書アプリケーション

図 7.2c および d に読書アプリケーションの画面を示す。実験参加者には、アプリケーションを操作しながら読書をするよう指示した。読書アプリケーションにおいて実験参加者は、表示する本の選択および次のページへ進む・前のページに戻ることが可能である。本の選択は、画面下部に配置している 4 つのアイコンをそれぞれ凝視することによりできる。次のページへ進むは画面左に配置しているボタンを、前のページに戻るは画面右に配置しているボタンを凝視することによりできる。使用した本は夏目漱石著の「こころ」、「吾輩は猫である」、「坊っちゃん」、および「一夜」であり、全て青空文庫²から選択した。図 7.2c の文章は横書き、図 7.2d の文章は縦書きとした。これは、縦方向および横方向への視線移動の軌跡を偏りなく取得するためである。

7.3.4 タイピングアプリケーション

図 7.2e にタイピングアプリケーションの画面を示す。実験参加者には、画面上部に表示されている課題文と同じ文章を作成するよう指示した。文字の入力は対応するキーを凝視することによりできる。スペースの表示は“_”（アンダーバー）とした。同じ文章の入力を終えた際は、‘Enter’ キーを入力するよう指示した。文章が正しい場合は次の課題文が表示される。課題文は計 5 文章あり、それぞれスペースを含め 32 文字から構成される。実際に使用した文章を表 7.1 に示す。これらの文章は MacKenzie らが提案したフレーズセット [MS03] から抽出した。

²<https://www.aozora.gr.jp> 最終閲覧 2020 年 1 月 15 日

表 7.2: 実験 3 のクイズアプリケーションに使用した問題文およびその回答。

出現順 (n 回目-m 番)	問題文	回答
1-1	本能寺の変を起こしたのは、明智光秀である	正
1-2	$13 \times 6 = 72$ である	誤
1-3	「顔に泥をぬる」とは恥をかかせるという意味である	正
1-4	人口の最も少ない県は高知県である	誤
2-1	徳川家光は、徳川 5 代目将軍である	誤
2-2	1 個 150 円のリンゴがある。これを 5 個買うと 750 円である	正
2-3	「目が高い」とは良いものを見分ける力がないことである	誤
2-4	国民が政治に参加する権利を参政権という	正

表 7.3: 実験 3 のジェスチャタスクにおける、2 段階の視線移動の方向に対応するコマンド。

	コマンド	1 回目	2 回目
ペイント	コピー	$U \rightarrow R$	$U \rightarrow L$
	ペースト	$D \rightarrow L$	$D \rightarrow R$
クイズ	正	$R \rightarrow D$	$R \rightarrow U$
	誤	$L \rightarrow U$	$L \rightarrow D$

7.4 実験手順

実験は、ペイントおよびクイズアプリケーションそれぞれを用いたタスク（以降、それぞれペイントタスクおよびクイズタスク）を 2 度ずつ、横書きおよび縦書きの読書アプリケーションを用いて 5 分間読書を行うタスク（以降、読書タスク）、タイピングアプリケーションを用いた 5 文章入力を行うタスク（以降、タイピングタスク）から構成される。提案手法への慣れが意図しない操作の発生に影響するかを調査するために、実験 2 にて提案手法を経験している C 群の実験参加者にはジェスチャアプリケーションを用いたタスクから実験を開始するよう指示し、D 群の実験参加者には凝視アプリケーションを用いたタスクから実験を開始するよう指示した。実験 3 に要した時間は約 55 分であった。実験終了後、実験参加者は謝金として 5,000 円を受け取った。

7.4.1 ジェスチャアプリケーションを用いたタスク

ジェスチャアプリケーションを用いたタスクは、2 段階の視線移動を用いてコマンドを実行する練習を行った後に実施された。この練習は、実験参加者の意図した視線移動が連続して 8 方向認識されるまで行い、約 212s 要した。その後、各タスクを 2 度ずつ行うよう指示した。タスクの実施順は、ペイント（1 回目）→ペイント（2 回目）→クイズ（1 回目）→クイズ（2 回目）の順である。各タスクの各回数における、コマンド（ペイントアプリケーションにおけるコピー・ペースト、クイズアプリケーションにおける正・誤）と 2 段階の視線移動の

方向の組み合わせは表 7.3 に示す通りである。例えばペイントタスクの 1 回目において四角の対象にコピー・ペーストを行う場合、実験参加者はまず四角の対象を凝視する。その後 $U \rightarrow R$ の方向へ視線を移動しコピーする。そして、図 7.2a 左の四角と同じ位置を凝視し、 $U \rightarrow L$ の方向へ視線を移動し、コピーした対象をペーストする。また、例えばクイズタスクの 1 回目では、実験参加者は問題が正解だと思える場合は適当な位置を凝視した後に $R \rightarrow D$ の方向へ視線を移動し、間違いだと思える場合は適当な位置を凝視した後に $R \rightarrow U$ の方向へ視線を移動する。なお、ジェスチャアプリケーションを用いたタスクの際は、凝視が認識された点を描画することにより実験参加者へ視覚的に通知し、8 方向いずれの視線移動が認識された場合は音声による通知を行った。実験による疲れの影響を取り除くため、各タスク終了後に 5 分間の休憩を設けた。合計、256 (8 方向 \times 2 回 \times 16 名) 回の意図的に 2 段階の視線移動に基づくコマンド実行を行なう際の視線移動の軌跡を収集した。

ジェスチャアプリケーションを用いたタスクの終了後に、実験参加者には、2 段階の視線移動を用いたコマンド実行に対する印象に関するアンケートへの記入を指示した。アンケートは以下の内容へ 5 段階のリッカート尺度 (5 がポジティブな回答、1 がネガティブな回答) を用いた回答および、自由記述からなる。

質問 1 視線を 2 段階に移動させることは容易でしたか？

質問 2 練習時は除いて 2 段階の視線移動がうまく認識されていないと感じましたか？

質問 3 視線を 2 段階に移動させることは疲れませんか？

7.4.2 凝視アプリケーションを用いたタスク

凝視アプリケーションを用いたタスクは、選択したい対象を 1.0s 間見続けると選択できることを説明し、実際にどのように選択できるかを試した後に実施された。タスクの実施順は、読書 (横書き) \rightarrow 読書 (縦書き) \rightarrow タイピングの順である。読書タスクでは、実験参加者に読む本は自ら決めていいと指示した。実験による疲れの影響を取り除くため、各タスク終了後に 5 分間の休憩を設けた。読書タスクでは実験参加者に、横書きおよび縦書きそれぞれ 5 分間読書をするように指示した。また、タイピングタスクでは 5 文章の入力に約 7 分間要した。そのため、合計 272 (17 (= 5 + 5 + 7) 分 \times 16 名) 分間の凝視に基づく操作を行う際の視線移動の軌跡を収集した。

7.5 実験結果

本節において、意図しない 2 段階の視線移動がどの程度認識されたか、および 2 段階の視線移動を用いたコマンド実行に関するアンケート結果を示す。

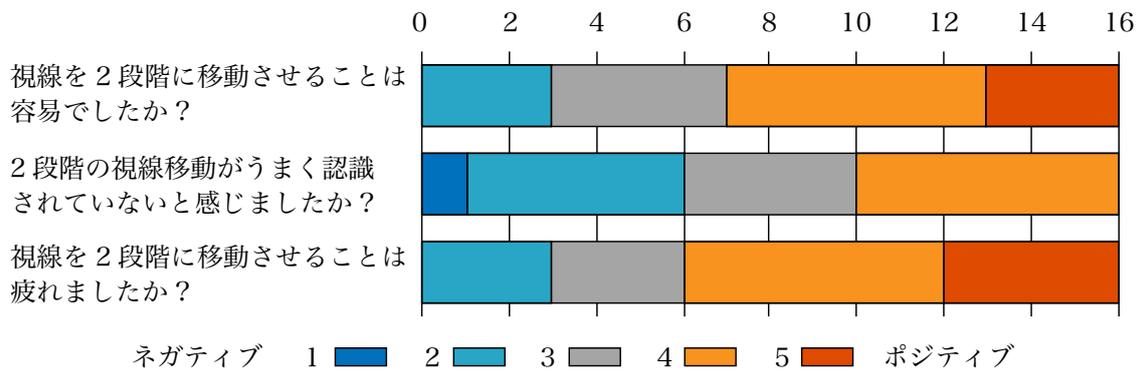


図 7.3: ジェスチャアプリケーションを用いたタスク時に実施したアンケートの結果。

7.5.1 意図しない2段階の視線移動の認識回数

本実験では、意図しない2段階の視線移動の認識回数を、凝視アプリケーション使用時に得られた視線移動の軌跡において認識された2段階の視線移動の認識回数とした。なお、凝視アプリケーション使用時に2段階の視線移動が認識されてもコマンドは実行されていない。結果、意図しない2段階の視線移動の認識回数は26回であった。横書きの読書アプリケーションでは5回、縦書きの読書アプリケーションでは4回、タイピングアプリケーションでは17回であった。C群では13回でありD群では13回であった。マンホイットニー検定の結果、C-D群(実験2に参加しているかどうか)間に有意差は見られなかった。なお、ジェスチャアプリケーション時に、表7.3に該当しない方向が認識された回数は5回であった。また、方向別ではR→Uは3回、U→Rは3回、L→Uは10回、U→Lは5回、R→Dは3回、D→Rは0回、L→Dは1回、およびD→Lは1回であった。

7.5.2 2段階の視線移動を用いたコマンド実行に対する印象

図7.3に、2段階の視線移動を用いたコマンド実行に対する印象に関するアンケートの結果を示す。質問1では、3名の実験参加者が2段階に視線を移動させることは難しいと回答し、質問2では6名の実験参加者が2段階の視線移動が正しく認識されなかったと回答した。しかし、全実験参加者はジェスチャアプリケーションの全タスクを正しく終えることができていた(コマンド実行ができないために実験を中止する必要はなかった)。これらのことから、実験参加者はコマンドを実行するために、2段階の視線移動を2回以上実施していたと考えられる。実験2において2段階の視線移動の認識率は84.9%であったことから、実験参加者が意図的にコマンドを実行しようとして、視線移動を2回行った際に1回でも正しく2段階の視線移動が認識される確率は97.7% $(1 - (1 - 0.849)^2)$ であると言える。また質問3の結果、3名の実験参加者が2段階の視線移動を行うことは疲れると回答した。しかし、提案手法のみ

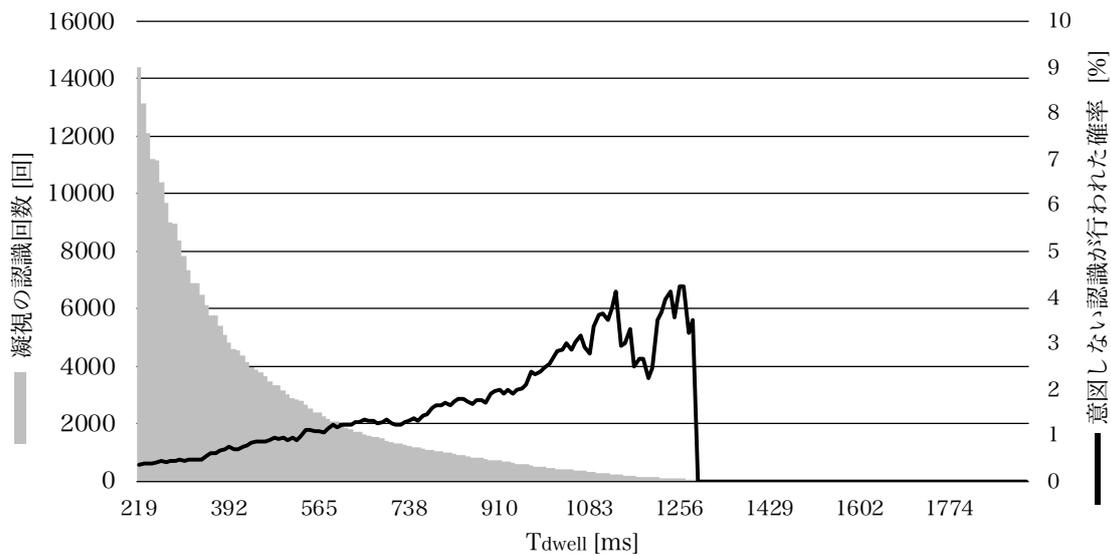


図 7.4: T_{dwell} ごとの凝視の認識回数と凝視が認識された際に意図しない認識が行われた確率。

ならず視線に基づく操作を行う際、一般的に目が疲れるという意見が得られていることから、この結果は妥当なものだと考える。

7.6 解析

凝視に基づく対象選択および2段階の視線移動を用いたコマンド実行を用いてアプリケーションを操作するときの視線移動の軌跡をもとに、2段階の視線移動の認識システムを改良するための解析を行う。

7.6.1 パラメータを変更した際の意図しない操作の認識回数

まず、各パラメータを変更した際に意図しない操作がどの程度生じるかをシミュレートする。7.5.1 節と同様に、意図しない2段階の視線移動の認識回数は、凝視アプリケーション使用時に得られた視線移動の軌跡において、認識された2段階の視線移動の認識回数である。シミュレーションに用いる各パラメータ (T_{dwell}、W_{path}、および D_{thld}) の値は以下の通りである。

T_{dwell} 219 ms から 1000 ms まで 10 ms ごとの値。計 178 値。
(219 ms、229 ms、239 ms、…、979 ms、989 ms、1000 ms)

W_{path} 42.9 mm から 5.0 mm まで 4.2 mm ごとの値。計 10 値。
(42.9 mm、38.7 mm、34.5 mm、…、11.5 mm、9.3 mm、5.0 mm)

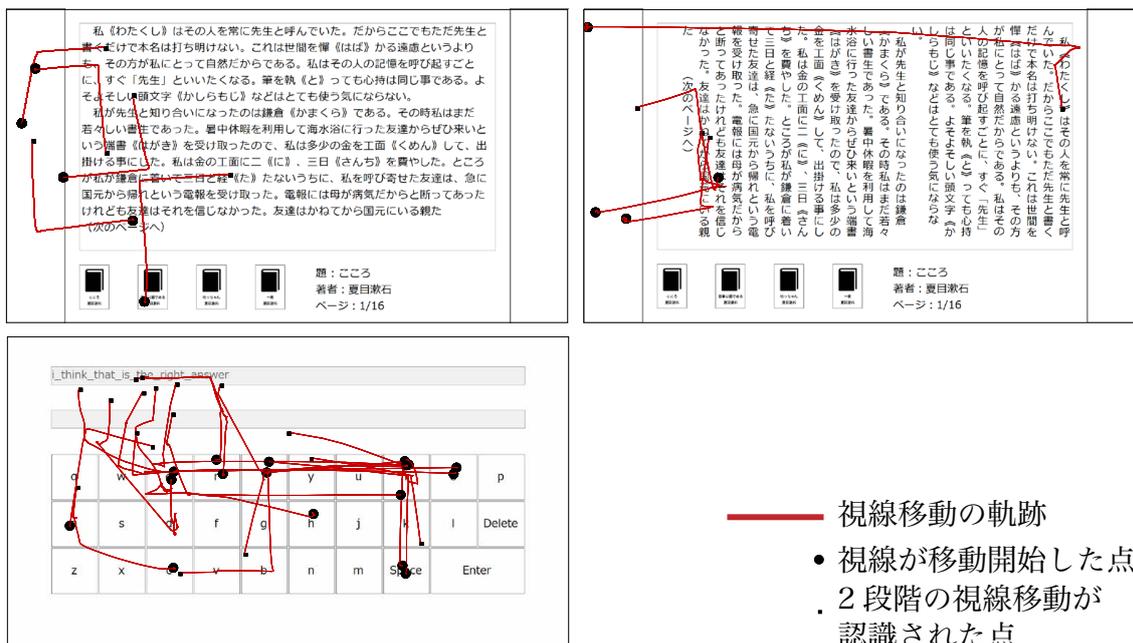


図 7.5: 意図しない 2 段階の視線移動が認識された際の視線移動の軌跡。

D_{thld} 55.8 mm から 537.0 mm まで 5.6 mm ごとの値。計 86 値。

(55.8 mm、61.4 mm、67.0 mm、…、526.2 mm、531.8 mm、537.0 mm)

これら合計 153,080 (178 × 10 × 86) のパラメータの組み合わせを用いて認識回数を求めた結果、多くの組み合わせにおいて認識回数が 0 回となった。認識回数が 0 回となったパラメータの組み合わせの中から、よりコマンド実行を容易だと考えられるパラメータを求める。

T_{dwell} に関して、5.2.1 節にて述べたように、 T_{dwell} が大きい場合は意図しない操作が発生しやすくなる。しかし、 W_{path} および D_{thld} は変更せずに T_{dwell} のみを 305 ms から 1290 ms まで増加させたシミュレーションを行った結果、意図しない操作の認識回数は 26 回から 0 回へ減少し、大きい T_{dwell} を用いることにより意図しない操作の発生を抑えられていた。この結果を提案手法では凝視の認識された後に 2 段階の視線移動の認識を行うため、大きい T_{dwell} を用いたときは、単に凝視の認識回数が少ないために 2 段階の視線移動の認識が行われなかったと考えた。図 7.4 に T_{dwell} ごとの、凝視の認識回数と凝視が認識された際に意図しない認識が行われた確率（意図しない認識の回数を凝視の認識回数にて割った確率）を示す。 T_{dwell} の増加に伴い、意図しない認識が行われた確率も増加していることがわかる。つまり、 T_{dwell} を大きくすることにより、意図しない 2 段階の視線移動の認識は行われやすいといえる。しかし、 T_{dwell} を大きくすることはコマンド実行の容易さにおいて有効であり、提案手法において凝視の認識回数を減らすことは、意図しない認識を減らすことにもつながるため、305 ms より適度に大きい値の使用を検討する。

次に、 W_{path} と D_{thld} を変更することの影響を調べる。 T_{dwell} を 305 ms として、 W_{path} と D_{thld} を変更し、意図しない 2 段階の視線移動の認識回数をシミュレートした。この場合においても、意図しない認識の回数が 0 回となる組み合わせは多くあった。これらの組み合わせの例としては、 $(W_{\text{path}}, D_{\text{thld}})$ が (5.3 mm, 55.8 mm) や (42.9 mm, 117.0 mm) の組み合わせが挙げられる。これらは、極端に小さい W_{path} もしくは大きい D_{thld} を用いた例であるが、このように極端な値を用いると意図しない認識の発生を抑えることができる。しかし、ユーザは視線をより直線的に、もしくは長い距離移動させる必要があるため意図したコマンド実行が難しくなる。そこで、意図しない認識の回数が 0 回となる組み合わせの中から、最も W_{path} が大きく D_{thld} が小さい組み合わせを求めた。この組み合わせでは、 W_{path} が 13.7 mm であり D_{thld} が 100.0 mm であった。しかし、依然として W_{path} が小さくかつ D_{thld} が大きく、実際に著者がこの組み合わせのパラメータ（つまり T_{dwell} が 305 ms、 W_{path} が 13.7 mm、 D_{thld} が 100.0 mm）において、コマンド実行をしようと試みたが失敗することが多かった。そこで、全 26 回の意図しない認識が行われた際の、2 段階の視線移動がどのように行われていたかを、視線移動の軌跡をプロットすることにより調査した。

図 7.5 に意図しない認識が行われた際の視線移動の軌跡のプロットを示す。図が示すように、多くの軌跡はジグザグではなく直線的になっていた。この結果から、 W_{path} を小さくすることは、意図しない操作を防ぐことにそこまで貢献していないことがわかる。さらに、多くの軌跡の移動量は、水平方向に比べ垂直方向が少ないこと、および意図しない認識の 1 段階目の方向は水平方向が多い（水平方向：17 回、垂直方向：9 回）こともわかる。そこで、 D_{thld} を水平方向 (D_h) と垂直方向 (D_v) に分けた場合の影響を調査した。 D_h および D_v それぞれが小さくなるのは D_h が 117.0 mm、 D_v が 89.3 mm の組み合わせのときであった。実験に用いたディスプレイのアスペクト比が 16 : 9 であったことも影響していると考えられるが、 D_v を D_h に比べ小さくすることも、意図しない認識を防ぐことだけではなく、コマンド実行の容易さに貢献していると考えられる。また、 D_{thld} を D_h と D_v に分けることは、人間の視野角が水平方向と垂直方向にて異なることから妥当である。人間の視野角は、左方向に 90°、右方向に 90°、上方向に 50°、および下方向に 80° である [RF61]。さらに、Møllenbach ら [MLGH10] の研究においても、垂直方向の 1 段階の視線移動は水平方向の 1 段階の視線移動に比べ、時間がかかると述べられていることから、 D_v を D_h より小さくすることは妥当である。

7.6.2 意図しない 2 段階の視線移動に要した時間

26 回認識された意図しない 2 段階の視線移動のうち、いくつかの視線移動には長い時間を要していた。そこで意図的に 2 段階の視線移動を行った際、および意図しない認識が行われた際の視線移動に要した時間を調査した。視線移動に要した時間は、視線が移動し始めた時間から 2 段階の視線移動が認識されるまでの時間である。意図的に視線移動を行った際に要した時間は、ジェスチャアプリケーションを用いたタスク、意図しない認識が行われた際に要した時間は、凝視アプリケーションを用いたタスクの際に収集した視線移動から求めた。図 7.6 に、意図的に視線移動を行った際に要した時間（上）および、意図しない認識が行われた際に要した時間（下）の 50 ms ごとの頻度を示す。意図的に視線移動を行った際に要した時間の平

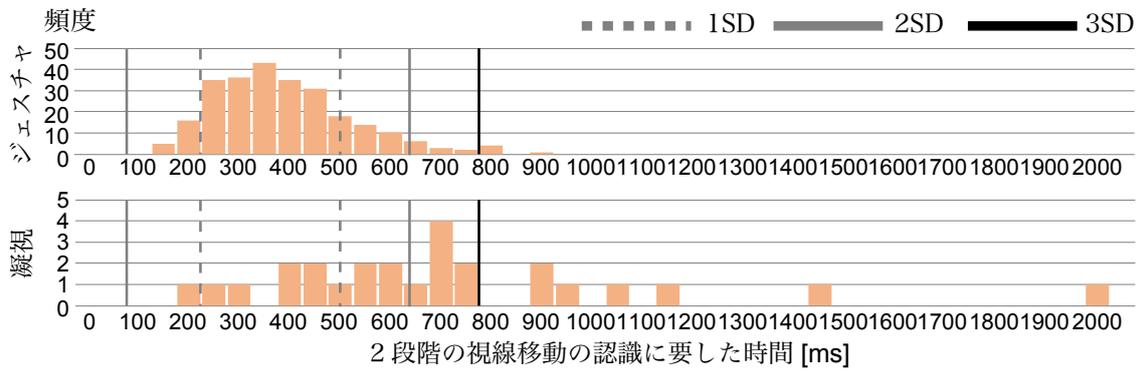


図 7.6: 2 段階の視線移動を認識に要した時間の頻度。上) 意図的に 2 段階の視線移動を行った際に要した時間の頻度。下) 意図しない 2 階の視線移動の認識が行われた際に要した時間。

均は 362 ms ($SD = 137$) であり、多くは 100–800 ms の範囲に取まっていた。対して、意図しない認識が行われた際に要した時間はばらついていることがわかる。この結果より、2 段階の視線移動を認識するための時間の範囲に制限を設けることにより、意図しない操作を防ぐことが可能だと考えた。実際に認識を行うための時間を、意図的に視線移動を行った際に要した時間の平均 $\pm 1SD$ (225–499 ms)、平均 $\pm 2SD$ (88–636 ms)、もしくは平均 $\pm 3SD$ (0–773 ms) とした際は、それぞれ 20 回 (76.9%)、13 回 (50.0%)、もしくは 7 回 (26.9%) の意図しない操作の認識を防ぐことができた。また、認識を行うための時間を 0–773 ms に制限した際に、意図した視線移動が認識されなかったのは全 256 回のうち 3 回 (1.2%) であった。これらのことから、2 段階の視線移動を認識するための時間を制限することは、意図しない操作を防ぐ点において有効であると考えられる。

7.6.3 解析のまとめ

以下に、上記の解析および実験 1 および 2 から発見した、意図しない 2 段階の視線移動を防ぐことが可能なシステムの実装に有効な項目を示す。

- 大きい T_{dwell} を用いることにより、意図しない認識を防ぐことおよびコマンド実行を容易にすることが可能である。しかし、大きい T_{dwell} を用いることは、意図しない認識を防ぐための根本的な解決方法ではない。
- 小さい W_{path} を用いることは、意図しない操作を防ぐために有効だと考えられるが、小さすぎる W_{path} はコマンド実行を難しくする。
- 大きい D_{thld} を用いることは、意図しない操作を防ぐために有効である。さらに、垂直方向に移動すべき距離 (D_v) を水平方向に移動すべき距離 (D_h) より短くすることはコマンド実行の容易にするという点においても有効である。

表 7.4: 改良後の 2 段階の視線移動の認識システムに用いたパラメータ。

パラメータ	値
T_{dwell}	506 ms
W_{path}	34.6 mm
D_h	116.0 mm
D_v	66.8 mm
2 段階の視線移動の認識に要して良い時間の上限	0-773 ms

- 2 段階の視線移動の認識に要して良い時間の上限を 773 ms とすること、つまり、773 ms 以上要している視線移動は意図的な 2 段階の視線移動ではないとすることにより、意図しない認識を防ぐことができる。

これらの項目を適切に組み合わせることにより、意図しない 2 段階の視線移動を防ぐこと、およびコマンド実行を容易にすることができる。

7.7 認識システムの改良

前節にて述べた項目を基に、意図しない 2 段階の視線移動を防ぐことが可能であり、かつ容易にコマンドを実行できるようにシステムを改良する。2 段階の視線移動の認識に要して良い時間の上限を 773 ms とし、 T_{dwell} 、 W_{path} 、 D_h 、および D_v を変更しながらシミュレーションを行う。このシミュレーションを通じて意図しない 2 段階の視線移動の認識が生じない、かつ、意図した 2 段階に視線移動が認識されやすいパラメータの組み合わせを求める。意図した 2 段階に視線移動が認識されやすいかどうかは、ジェスチャアプリケーションを用いたタスク時に収集したコマンド実行を実際に行う際の視線移動が正しく認識されるかをもとに決定する。つまり、合計 256 回のうち何回認識されるかという、意図した操作の認識率を用いる。なお、実験 3 ではこれまでに決定したパラメータをもとに実験を行っているため、意図した 2 段階に視線移動の認識率は低くなると考えらる。シミュレーションの結果、 T_{dwell} を 506 ms、 W_{path} を 34.6 mm、 D_h を 116.0 mm、および D_v を 66.9 mm としたパラメータの組み合わせが最適であるとした (表 7.4)。このパラメータの組み合わせでは、意図しない認識は生じず、かつ意図した操作の認識率は 40.6% (256 回中 104 回) である。この 40.6% という認識率は、意図しない認識が生じないパラメータの組み合わせの中では最大の認識率である。 T_{dwell} として用いる 506 ms は凝視に基づく対象選択手法に関する先行研究において、大きすぎない凝視時間である。また、 D_h として用いる 116.0 mm は実験に用いたディスプレイの横幅 (531 mm) の 21.8% の大きさであり、 D_v として用いる 66.9 mm は実験に用いたディスプレイの縦幅 (299 mm) の 22.3% である。そのため、視線を移動する必要のある絶対距離は水平方向と垂直方向において異なるが、ディスプレイ上の相対的な距離 (つまり、各方向ディスプレイ上の 22.0% 程度の距離) は近しいため、コマンド実行の観点においても有効であると考えられる。

第8章 実験4：システム改良後の提案手法の性能評価

実験4の目的は、実験3にて意図しない2段階の視線移動を防ぐことが可能であり、容易にコマンドを実行できるように改良したシステムを用いた場合、ユーザの意図した2段階の視線移動はどの程度認識され、どの程度時間を要するかを調べることである。また、7.5.2節にて述べたように、実験参加者はコマンドを実行するために視線移動を複数回行っていたと考えられるため、実験4ではコマンドを実行するために何度2段階の視線移動を行ったかも調査する。実験4では、視線認識機器から収集される視線情報に対して実験1-3と同じローパスフィルタを適用し、使用するシステムは7.7節にて改良したシステムである。つまり、 T_{dwell} は506 ms、 W_{path} は34.6 mm、 D_h は116.0 mm、 D_v は66.8 mmであり、2段階の視線移動の認識に要して良い時間の上限は773 msである。

8.1 実験参加者

男性10名を実験参加者として募った。実験参加者は著者が所属する研究室内の学生を含む21-25歳（平均23.1歳）であり、4名が眼鏡を着用していた。10名いずれも以前に実験参加者として視線認識機器を使用したことがあるが、日常的には使用していなかった。

8.2 実験機器および実験環境

実験4は、図7.1に示すような環境にて実施された。実験に用いた機器、その配置、照明条件は実験3と同様である。

8.3 実験手順

実験アプリケーションは実験2に用いたアプリケーションと同様である。実験参加者には、このアプリケーションを用いて、画面中心(C)を始点とした8方向($R \rightarrow U$ 、 $U \rightarrow R$ 、 $L \rightarrow U$ 、 $U \rightarrow L$ 、 $R \rightarrow D$ 、 $D \rightarrow R$ 、 $L \rightarrow D$ 、 $D \rightarrow L$)の視線移動が正しく認識されるまで行うように指示した。実験開始前に視線認識機器のキャリブレーションを行った。その後、実験参加者には2段階の視線移動がどのように認識されるかに慣れるための練習を少なくとも5分間行い、練習セッションを1回行うよう指示した。練習セッションの終了後、計5回の本番セッションを



図 8.1: 実験 4 の環境。

行う。各セッション間に少なくとも 1 分の休憩を設けた。実験 4 を通じて、合計 400 回（8 方向 × 5 セッション × 10 人）の意図した 2 段階の視線移動を収集した。実験 4 に要した時間は実験参加者ごとに約 25 分であった。

8.4 実験結果

図 8.2 に改良したシステムを用いた際の提案手法の性能を示す。全方向の認識率（図 8.2、濃いグレー）は、2 段階の視線移動を 1 度行った場合（左）において平均 85.8% ($SD = 5.1$) であった。このとき実験参加者ごとの認識率の最大は 90.0% であり、最小は 72.5% であった。1 度行った場合の結果から、推測される 2 度行った際の認識率は 97.9% ($1 - (1 - 0.858)^2$) であるが、実際に 2 段階の視線移動を 2 度行った場合（右）における平均認識率は 99.0% ($SD = 1.2$) であった。また、実験参加者ごとの認識率の最大は 99.0% であり、最小でも 97.5% と非常に高くなった。全方向の認識率は（図 8.2、薄いグレー）は、2 段階の視線移動 1 度行った場合（左）において平均 956 ms ($SD = 119$) であり、1 回目に失敗し 2 回目を行った場合は 1031 ms ($SD = 112$) であった。各方向ごとの認識率を図 8.3 に示す。 $L \rightarrow U$ および $R \rightarrow D$ の認識率がそれぞれ 76.0% および 72.0% と低くなったが、2 度行った場合はそれぞれ 98.0% および 100% となった。これらの結果より、ユーザはコマンド実行を行うために視線移動をした際に、コマンドが実行されなかったとしても、再度視線移動を行うことによりコマンドを実行できると言える。そしてこれは、ユーザや方向に限らず言えることである。

2 段階の視線移動の認識に要して良い時間の範囲（0–773 ms）に関して、4 回の 2 段階の視線移動がこの範囲を超えたために認識されなかった。2 段階の視線移動の認識に要した時間は、2 段階の視線移動を 1 度行ったときに 450 ms ($SD = 119$) であり、2 度行ったときに 525 ms ($SD = 112$) であった。1 度目と 2 度目で要した時間が異なった理由としては、実験参加者が 2 度目の視線移動をより慎重に行ったためだと考えられる。この範囲は実験 3（つまり、実験

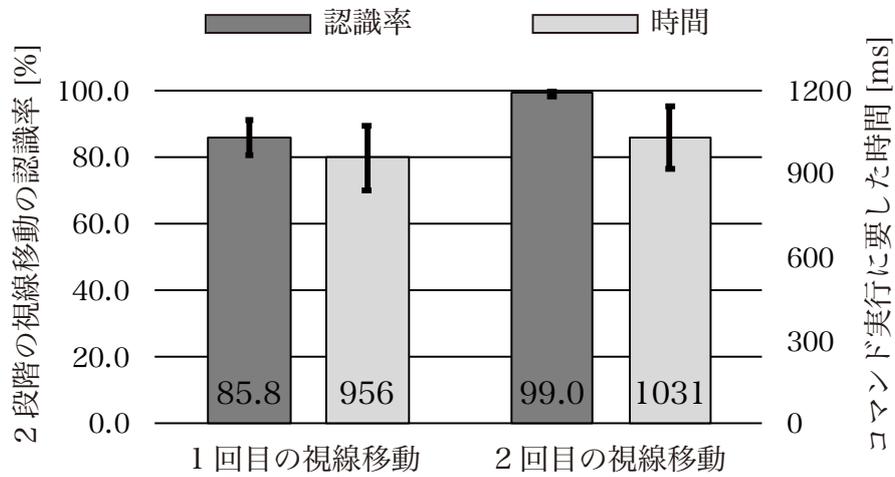


図 8.2: 2段階の視線移動を1度および2度行った際の、2段階の視線移動の認識率およびコマンド実行に要した時間。

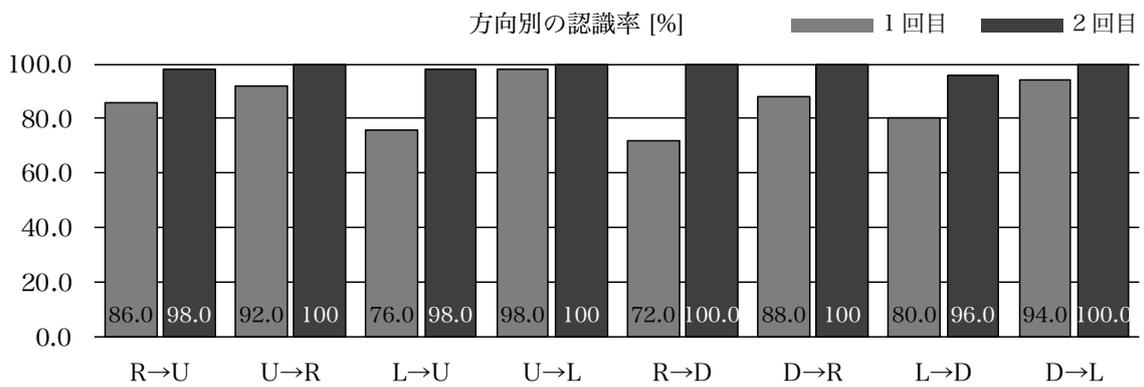


図 8.3: 2段階の視線移動を1回および2回行った際の、2段階の視線移動の方向ごとの認識率。

4とは異なるパラメータを用いた実験)の結果から求めたものであるが、実験4においても有効であった。

第9章 議論

本章において、本研究から明らかとなった今後の課題および本研究の結果を議論する。

9.1 実験条件による2段階の視線移動の認識システムへの影響

本研究における2段階の視線移動の認識システムの実装は、実験1-3（第4、6、7章）において収集した視線移動の軌跡を基に行われた。これらの実験時の制限による影響を述べる。

9.1.1 実験参加者

実験1の実験参加者1名は、眼球振盪（もしくは眼振。眼球が自分の意志とは関係なく規則的に繰り返す往復運動のこと）が生じると述べており、実際に実験1において収集した視線移動の軌跡を観察したところ、画面中心において視線が左右に移動していることが観測された（特に、図4.5の $U \rightarrow R$ 、 $U \rightarrow L$ 、 $D \rightarrow L$ 、 $D \rightarrow R$ において見られた）。さらに、実験1-3の実験参加者は全員若く普段からコンピュータを扱っていた。眼球振盪の影響は、凝視に基づく対象選択手法においても影響すると考えられるため、提案手法のみに影響する問題ではないが、眼球（もしくは眼）に何らかの症状を患っているユーザ、または、高齢者のユーザにおいても、本研究において実装システムおよびそのパラメータを用いることが適切かを調査する必要がある。

9.1.2 実験アプリケーション

実験に用いたアプリケーションは、実験1および2（第4および6章）では背景が無地のアプリケーションであり、実験3（第7章）では、ペイント、クイズ、読書、およびタイピングアプリケーションの4つであった。意図しない操作を防ぐためには、より多くの種類のアプリケーションや環境にて視線移動の軌跡を収集し、システムをさらに改良する必要がある。例えば、より視線が移動すると考えられる動画視聴を促すアプリケーションの使用時や、複数アプリケーションが画面に表示されている場合における操作時の視線移動をさらなる改良のために調査する必要がある。

9.1.3 画面背景の影響

実験2および実験3のジェスチャアプリケーションを用いたタスク時（つまり、実験参加者に2段階の視線移動を行うよう指示したとき）は、実験に関係のないアイコンなどを画面に表示しなかった。しかし、実際にコンピュータを操作する際は、アイコンや文章、画像などが表示され、これらが視線移動に影響すると考えられる。実際に、実験2では2名、実験3では3名の実験参加者が、画面に視線移動を行う際の参考となるアイコンなどが表示されていないため、視線移動が難しかったとコメントした。さらに、実験2の1名の実験参加者が、画面の縁にそって視線移動を行うと、視線移動が容易に感じたコメントした。これらのコメントから、画面背景の表示が視線移動の容易さに影響し、ひいては2段階の視線移動の認識率に良い影響を与えると考えられる。対して、例えば、ユーザが意図的に2段階の視線移動を行おうとしてる際に、画面に表示されているアイコンに視線が移り、視線移動を失敗するとも考えられる。そのため、画面背景の表示がどのような影響を与えるかも調査する必要がある。

9.2 2段階の視線移動の使用およびその認識システムの実装に関する議論

本研究では、2段階の視線移動をジェスチャとして用い、意図しない操作が生じないように認識システムを実装した。ジェスチャに関して、2段階の視線移動より簡易な1段階の視線移動をジェスチャとして用いなかったことが適切であったかを検証する。また、手に基づく操作手法において用いられているような、既存のジェスチャ認識システムを使用しなかった理由を述べる。

9.2.1 1段階の視線移動の使用に関して

意図した1段階の視線移動のみを認識するシステムを実装することによる、意図しない操作が生じない1段階の視線移動を用いたコマンド実行手法の提案に関して述べる。1段階の視線移動は、2段階の視線移動より日常的に生じる視線移動である [DS07, RMSA+15] ため、このシステムを実装する場合、パラメータをより制限する必要がある。そこで、どのようなパラメータであれば、意図しない操作を防ぐことができるかを調べるために、実験3にて収集した視線移動の軌跡を用いたシミュレーションを行った。例えば、 T_{dwell} を 506 ms とした場合に意図しない1段階の視線移動の発生を0回にするには、 W_{path} を 5.3 mm より小さく、 D_h を 368 mm（実験に用いたディスプレイの横幅の 69.3%）より大きく、かつ D_v を 167 mm（167 mm は実験に用いたディスプレイの縦幅の 55.9%）より大きくする必要があることがわかった。このパラメータの組み合わせにおいてユーザは、視線をより直線的にかつ、水平・垂直方向ともにディスプレイの半分以上視線を移動する必要があり、意図的にコマンドを実行することが非常に難しくなる。また、ユーザがコマンド実行に慣れ、容易にコマンドを実行

できるようになったとしても、ある対象に対して実行可能なコマンドの数が制限される。例えば、ディスプレイを4領域（右上、左上、左下、右下領域）に分割したとして、右上にある対象に実行可能なコマンドは下、左もしくは左下方向への3ジェスチャに紐づく3つのコマンドのみとなる。つまり、1段階の視線移動は、アプリケーションやシステム全体へのコマンドを実行するには向いていると思われるが、複数あるアイコンの中から1つのアイコンにコマンドを実行することには不向きと言える。そのため、本研究において2段階の視線移動をジェスチャ用いたことは、適切であると言える。

9.2.2 ジェスチャ認識システムの使用に関して

本研究において既存のジェスチャ認識システムを用いずに、 $R \rightarrow U$ 、 $U \rightarrow R$ 、 $L \rightarrow U$ 、 $U \rightarrow L$ 、 $R \rightarrow D$ 、 $D \rightarrow R$ 、 $L \rightarrow D$ 、および $D \rightarrow L$ の8ジェスチャを認識するシステムを実装した理由を述べる。手に基づく操作手法においては、例えば、\$1 Recognizer [WWL07]、Protractor [Li10] や動的時間伸縮法を用いてジェスチャ認識が行われてきている。例えば、マウスを用いた操作におけるジェスチャ認識を\$1 Recognizerにて行う場合¹は、マウスの左ボタンを押しながら任意のジェスチャを行い、終えた点にて左ボタンから指を離すことによりできる。認識システムは、マウスの左ボタンが押下された点をジェスチャの開始点、指が離れた点をジェスチャの終了点とし、この間のマウスの軌跡を用いてジェスチャを認識する。これらの認識システムを用いなかった理由は、視線に基づく操作においてジェスチャ認識の開始点と終了点を決定することが難しいためである。例えば、ジェスチャ認識の開始点と終了点を凝視が認識された点とすると、凝視が2度認識された時点で何らかのジェスチャが認識されコマンドが実行される。なぜなら、これらの認識システムは、手に基づく操作手法において使用されることを想定しており、一度開始点が認識されると何かしらのジェスチャを必ず認識するように実装されているためである。つまり、ミダタッチ問題に直面している凝視に基づく対象選択手法では、意図せずジェスチャの開始点が認識され意図しないコマンドが実行される可能性がある。そのため、これらの認識システムを使用せず、2段階の視線移動の認識システムを実装した。なお、実装した2段階の視線移動の認識システムにおいても、凝視が認識された点をジェスチャの認識開始点をとしているが、視線移動の認識に用いられる1段階目もしくは2段階目の経路から視線が外れると、ジェスチャの認識を終了するよう設計しているため、何かしらのコマンドが必ず実行されることはない。

9.3 さらなるアプリケーション例

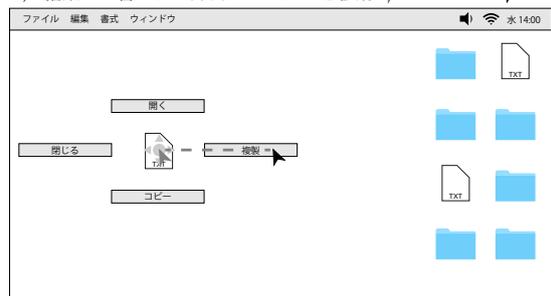
2段階の視線移動に基づくコマンド実行手法をより有効に利用するためのアプリケーション例に関して述べる。

¹<http://depts.washington.edu/acelab/proj/dollar/index.html> 参考

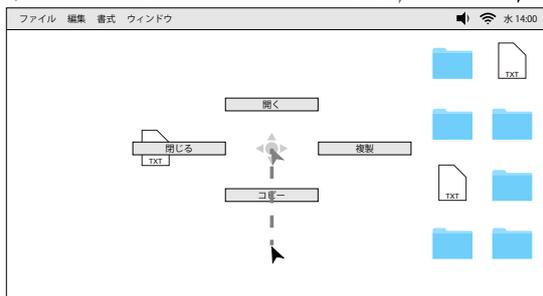
a) 任意のアイコンの上にて右ボタンを押す ▶



b) “編集” と書かれた方向へカーソルを移動 ▶



c) “コピー” と書かれた方向へカーソルを移動 ▶



d) 右ボタンから指を離す ▶

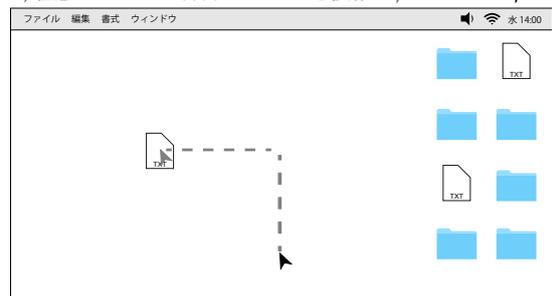


図 9.1: マウスを用いた操作における、マーキングメニューを表示する場合のコマンド実行の手順。

a) 任意のアイコンの上にて右ボタンを押す



b) 任意のコマンドの方向へカーソルを移動



c) 右ボタンから指を離す

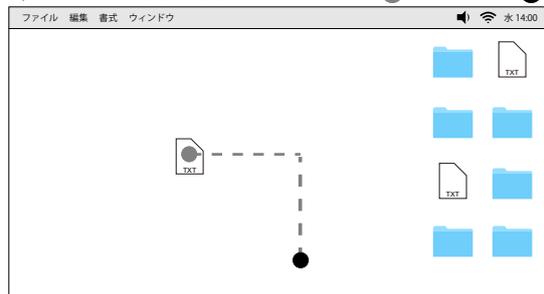


図 9.2: マウスを用いた操作における、マーキングメニューを表示しない場合のコマンド実行の手順。

a) 任意のアイコンを 506ms 間見つめる ●



b) 任意の方向へ視線を移動 ● - - - - ●



c) コマンドが実行される ●



図 9.3: 提案手法における、マーキングメニューを表示しない場合のコマンド実行の手順。

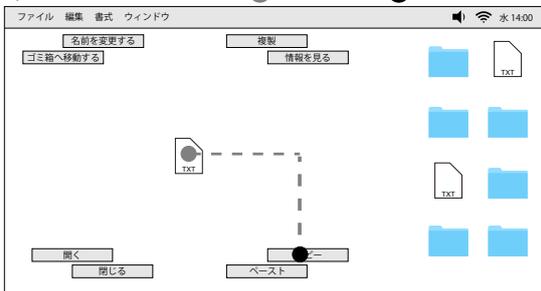
a) 任意のアイコンを 506ms 間見つめる ●



b) さらに 794 (1300 - 506)ms 間見つめる ●



c) 任意の方向へ視線を移動 ● - - - ●



d) コマンドが実行される ●

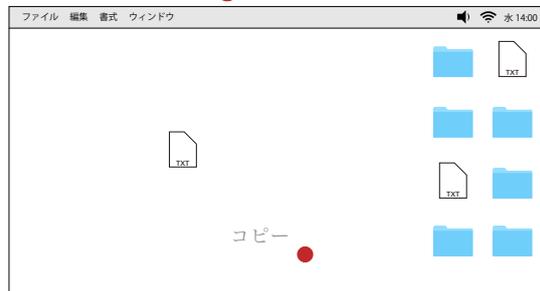


図 9.4: 提案手法における、マーキングメニューを表示する場合のコマンド実行の手順。

9.3.1 マーキングメニューへの応用

提案手法では、ガイドの表示を行わないため、慣れていないユーザは、どのコマンドがどのジェスチャに紐付けられているかがわからなくなる可能性がある。この課題を解決するために、マーキングメニュー [Kur93, KB94] の概念を取り入れることを検討する。

マウスを用いた操作において、マーキングメニューを用いてコマンドを実行する手順は以下の通りである (図 9.1)。まず、マウスの右ボタンを押しメニューを表示する (図 9.1a)。その後、表示されるメニューを参考に任意のコマンドを実行可能な方向へカーソルを移動する (図 9.1b)。そして、新たに表示されるメニューを参考に任意のコマンドを実行可能な方向へカーソルを移動し (図 9.1c)、右ボタンから指を離すとコマンドを実行できる (図 9.1d)。また、実行したいコマンドが紐付いている方向を既に知っているユーザは、メニューの表示を待たずにコマンドを実行することも可能である。この場合、コマンドを実行する手順は以下の通りである (図 9.2)。まず、マウスの右ボタンを押した後に (図 9.2a)、任意のコマンドを実行可能な方向へカーソルを移動し (図 9.2b)、右ボタンから指を離すとコマンドを実行できる (図 9.2c)。

現在、実装した 2 段階の視線移動の認識システムは、ガイドを表示することができないため、実行したいコマンドが紐付いているカーソルを移動方向を既に知っているユーザを対象としたシステムであると言える。そこで、長い時間凝視を続けるとガイドを表示するようにシステムを改良する。しかし、視線に基づく操作手法を用いてコマンドを実行する際に、ミダスタッチの発生により、意図せずガイドが表示されてしまう問題が先行研究において生じている。そのため、ガイドを表示するための凝視時間は、本研究において行った実験の結果から決定する。具体的には、図 7.6 が示すように、実験 3 において凝視の認識回数が 0 回となった凝視時間である 1,300 ms を用いる。つまり、実行したいコマンドが紐付いている方向を既に知っているユーザは、短い時間 (506 ms (= T_{dwell})) の凝視を行った後に視線移動を行うことによりコマンドを実行でき (図 9.4)、そうでないユーザは、長い時間 (1,300 ms) の凝視を行い、表示されるガイドを参考にしながらコマンドを実行できる (図 9.3)。なお、図 9.3 におけるガイドの表示方法は一例であり、先行研究 (例えば [DCN15]) を参考に、視線に基づく操作手法に適切な表示方法を模索する必要がある。

9.3.2 仮想現実空間への応用

2 段階の視線移動に基づく操作を仮想現実空間に応用することにより、認識システムの課題の解決を試みる。実験 6 の結果から、画面 4 角を開始とし、画面外側へ 2 段階の視線移動を行う際に、実装した認識システムを使用することが難しいことがわかった。この原因としては、主に視線認識機器が認識可能な視線範囲が画面内に制限されている (厳密には画面外も少しの範囲は認識可能である) ことが考えられる。また、画面 4 角から視線を移動するには、視線移動を開始する際に既に視線を移動させておく必要があるため、さらなる視線移動が難しいことが考えられる。

仮想現実空間において、視線認識はユーザが頭部に装着する仮想現実空間を体験するため

の装置により行われる。この装置では、頭部の向きを変更することによりユーザの視点を変更することが可能であるため、視線認識が可能な範囲が制限されない。つまり、ある視点において端を開始点とした2段階の視線移動を行う際は、予めその点の方向へ頭部の向きを変更することにより、視界の中心を開始とした2段階の視線移動を行うことができる。これにより、実装したシステムの解決が可能だと考える。

なお、上記のことは、仮想現実空間に限らず拡張現実空間や、メガネ型の視線認識機器（例えば、Tobii Pro Glass²）を使用することにより解決できると考えられる。

9.3.3 パラメータの動的な調整

2段階の視線移動の認識システムに用いるパラメータを動的に調整することにより、意図しない操作が生じない認識システムとすることを試みる。具体的には、 W_{path} を小さく、 D_h および D_v を大きく、および、2段階の視線移動の認識に要している時間の範囲を狭くしていく。このパラメータの調整は、ユーザが意図したコマンドを容易に実行できる範囲内にて行う必要があるため、意図したコマンドが実行されるときに視線移動の軌跡を用いて行う。この動的な調整を実現することにより、各パラメータをユーザやアプリケーションなどに対して適切な値とすることができる。結果として、9.1節にて述べた本研究における実験時の制限の影響を受けにくい認識システムへと改良することができる。

²<https://www.tobii.com/ja/product-listing/tobii-pro-glasses-2/>

第10章 結論

本論文において、2段階の視線移動に基づく操作手法を示した。ユーザは、「右を見た後に上を見る」ような視線移動を行なうことにより、コマンドを実行できる。これを実現するために、ユーザがコマンドを実行するために意図的に視線を2段階に移動したか、視線が不意に2段階に移動したかを識別するシステムを実装した。このシステムを用いることにより、視線に基づく操作手法の課題である意図しない操作を防ぐことができる。システムの実装は3つの実験を通じて収集したユーザの視線移動の軌跡を基に行なわれた。まず1つ目の実験では、ユーザに1段階および2段階に視線を移動するように指示した際の視線移動の軌跡を収集し、この視線移動の軌跡からその特徴を発見し、認識システムを実装した。2つ目の実験では、実装した認識システムを用いた場合にどの程度の性能となるかを調査した。結果から、ユーザはコマンド実行を74.7%の精度にて成功でき、要する時間は810msであることがわかった。また、このときに収集した視線移動の軌跡を用いて、認識精度が向上するようにシステムの改良を行った。続く3つ目の実験では、視線に基づく操作手法を用いてユーザに様々なアプリケーションを操作するよう指示した際の視線移動の軌跡を収集した。これらのアプリケーションを操作している際に、意図しない操作が生じないようにシステムのさらなる改良を行った。これら3つの実験を通じて、意図しない操作が生じないかつ、高い認識率にて意図した2段階の視線移動を認識可能なシステムの実装および改良を行った。さらに、3つの実験を通じて実装した認識システムを用いた場合にどの程度の性能となるかを調査した。この結果から、ユーザはコマンド実行を85.8%の精度にて成功でき、このときに要した時間は956msであることを示した。また、ユーザは一度コマンド実行を失敗したとしても、再度2段階の視線移動を行なうことにより、99.0%の精度にてコマンド実行を成功できることも示した。

謝辞

本研究を進めるにあたり、志築文太郎先生、高橋伸先生には多くのご意見とご指導を頂きました。心から感謝致します。特に、志築文太郎先生には、常日頃より、手法の内容、実験設計、また、論文の執筆手法といった研究のことのみならず、日常生活における研究者の心構えなど多くのことをご指導頂きました。重ねて感謝致します。

インタラクティブプログラミング研究室の後輩達、同期、先輩方には様々な面において多くのご助言を頂きました。特に WAVE チームの皆様にはチームゼミにおけるご意見だけでなく、研究室にて相談に乗っていただき多くのご意見、アドバイスを頂きました。深く感謝致します。中でも、安藤宗孝氏には研究室に配属されて以来、研究方針、実験設計などに関して幾度となく相談および議論をさせて頂きました。お礼申し上げます。

また、学生生活の多くの面において私を支えてくださった友人達、大学生活にてお世話になった皆様に感謝申し上げます。最後に、私を育て、様々な面において私を支えてくださった両親に深く感謝致します。

参考文献

- [BVH12] Roman Bednarik, Hana Vrzakova, and Michal Hradis. What Do You Want to Do Next: A Novel Approach for Intent Prediction in Gaze-based Interaction. In *Proceedings of the 2012 ACM Symposium on Eye Tracking Research & Applications*, ETRA '12, pp. 83–90, New York, NY, USA, 2012. ACM.
- [CHWS88] Jack Callahan, Don Hopkins, Mark Weiser, and Ben Shneiderman. An Empirical Comparison of Pie vs. Linear Menus. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '88, pp. 95–100, New York, NY, USA, 1988. ACM.
- [CXH15] Ishan Chatterjee, Robert Xiao, and Chris Harrison. Gaze+Gesture: Expressive, Precise and Targeted Free-Space Interactions. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, ICMI '15, pp. 131–138, New York, NY, USA, 2015. ACM.
- [DAH12] Morten Lund Dybdal, Javier San Agustin, and John Paulin Hansen. Gaze Input for Mobile Devices by Dwell and Gestures. In *Proceedings of the 2012 ACM Symposium on Eye Tracking Research & Applications*, ETRA '12, pp. 225–228, New York, NY, USA, 2012. ACM.
- [DCN15] William Delamare, Céline Coutrix, and Laurence Nigay. Designing Guiding Systems for Gesture-based Interaction. In *Proceedings of the 7th ACM SIGCHI Symposium on Engineering Interactive Computing Systems*, EICS '15, pp. 44–53, New York, NY, USA, 2015. ACM.
- [DHI17] William Delamare, Teng Han, and Pourang Irani. Designing a Gaze Gesture Guiding System. In *Proceedings of the 19th International Conference on Human-Computer Interaction with Mobile Devices and Services*, MobileHCI '17, pp. 26:1–26:13, New York, NY, USA, 2017. ACM.
- [DS07] Heiko Drewes and Albrecht Schmidt. Interacting with the Computer Using Gaze Gestures. In *Proceedings of the 11th IFIP TC 13 International Conference on Human-computer Interaction - Volume Part II*, INTERACT '07, pp. 475–488, Berlin, Heidelberg, 2007. Springer-Verlag.

- [EVBG15] Augusto Esteves, Eduardo Velloso, Andreas Bulling, and Hans Gellersen. Orbits: Gaze Interaction for Smart Watches using Smooth Pursuit Eye Movements. In *Proceedings of the Annual ACM Symposium on User Interface Software and Technology*, UIST '15, pp. 457–466, New York, NY, USA, 2015. ACM.
- [Fit54] P. M. Fitts. The Information Capacity of the Human Motor System in Controlling the Amplitude of Movement. *Journal of Experimental Psychology*, Vol. 74, pp. 381–391, 1954.
- [FWT⁺17] Anna Maria Feit, Shane Williams, Arturo Toledo, Ann Paradiso, Harish Kulkarni, Shaun Kane, and Meredith Ringel Morris. Toward Everyday Gaze Input: Accuracy and Precision of Eye Tracking and Implications for Design. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '17, pp. 1118–1130, New York, NY, USA, 2017. ACM.
- [HIV12] Aulikki Hyrskykari, Howell Istance, and Stephen Vickers. Gaze Gestures or Dwell-based Interaction? In *Proceedings of the 2012 ACM Symposium on Eye Tracking Research & Applications*, ETRA '12, pp. 229–232, New York, NY, USA, 2012. ACM.
- [IAST18] Toshiya Isomoto, Toshiyuki Ando, Buntarou Shizuki, and Shin Takahashi. Dwell Time Reduction Technique Using Fitts' Law for Gaze-based Target Acquisition. In *Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications*, ETRA '18, pp. 26:1–26:7, New York, NY, USA, 2018. ACM.
- [IBHV08] Howell Istance, Richard Bates, Aulikki Hyrskykari, and Stephen Vickers. Snap Clutch, a Moded Approach to Solving the Midas Touch Problem. In *Proceedings of the 2008 Symposium on Eye Tracking Research & Applications*, ETRA '08, pp. 221–228, New York, NY, USA, 2008. ACM.
- [IHI⁺10] Howell Istance, Aulikki Hyrskykari, Lauri Immonen, Santtu Mansikkamaa, and Stephen Vickers. Designing Gaze Gestures for Gaming: An Investigation of Performance. In *Proceedings of the 2010 ACM Symposium on Eye-Tracking Research & Applications*, ETRA '10, pp. 323–330, New York, NY, USA, 2010. ACM.
- [Jac90] Robert J. K. Jacob. What You Look at is What You Get: Eey Movement-based Interaction Techniques. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '90, pp. 11–18, New York, NY, USA, 1990. ACM.
- [Jac91] Robert J. K. Jacob. 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, 1991.

- [Jac93] Robert J. K. Jacob. Eye Movement-Based Human-Computer Interaction Techniques : Toward Non-Command Interfaces. In *Advances in Human-Computer Interaction*, pp. 4: 151–190, 1993.
- [JHF17] Florian Jungwirth, Michael Haslgrübler, and Alois Ferscha. Contour-guided Gaze Gestures: Eye-based Interaction with Everyday Objects and IoT Devices. In *Proceedings of the Seventh International Conference on the Internet of Things, IoT '17*, pp. 26:1–26:2, New York, NY, USA, 2017. ACM.
- [JS03] Robert J. K. Jacob and Keith S.Karn. Commentary on Section 4 - Eye Tracking in Human-Computer Interaction and Usability Research: Ready to Deliver the Promises. *The Mind's Eye: Cognitive and Applied Aspects of Eye Movement Research*, pp. 573–605, 2003.
- [KB94] Gordon Kurtenbach and William Buxton. User Learning and Performance with Marking Menus. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '94*, pp. 258–264, New York, NY, USA, 1994. ACM.
- [KSB08] Yvonne Kammerer, Katharina Scheiter, and Wolfgang Beinhauer. Looking My Way Through the Menu: The Impact of Menu Design and Multimodal Input on Gaze-based Menu Selection. In *Proceedings of the 2008 ACM Symposium on Eye Tracking Research & Applications, ETRA '08*, pp. 213–220, New York, NY, USA, 2008. ACM.
- [Kur93] Gordon Kurtenbach. *The Design and Evaluation of Marking Menus*. PhD thesis, Toronto, Ont., Canada, 1993. UMI Order No. GAXNN-82896. University of Toronto.
- [Li10] Yang Li. Protractor: A Fast and Accurate Gesture Recognizer. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '10*, pp. 2169–2172, New York, NY, USA, 2010. ACM.
- [MAv09] Päivi Majaranta, Ulla-Kaija Ahola, and Oleg Špakov. Fast Gaze Typing with an Adjustable Dwell Time. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '09*, pp. 357–360, New York, NY, USA, 2009. ACM.
- [MHL13] Emilie Møllenbach, John Paulin Hansen, and Martin Lillholm. Eye Movements in Gaze Interaction. *Journal of Eye Movement Research*, Vol. 6, No. 2, pp. 1–15, 2013.
- [MHLG09] Emilie Møllenbach, John Paulin Hansen, Martin Lillholm, and Alastair G. Gale. Single Stroke Gaze Gestures. In *Extended Abstracts on the SIGCHI Conference on Human Factors in Computing Systems, CHI EA '09*, pp. 4555–4560, New York, NY, USA, 2009. ACM.

- [MLGH10] Emilie Møllenbach, Martin Lillholm, Alastair Gail, and John Paulin Hansen. Single Gaze Gestures. In *Proceedings of the 2010 ACM Symposium on Eye-Tracking Research & Applications*, ETRA '10, pp. 177–180, New York, NY, USA, 2010. ACM.
- [MS03] I. Scott MacKenzie and R. William Soukoreff. Phrase Sets for Evaluating Text Entry Techniques. In *Proceedings of the SIGCHI Conference Extended Abstracts on Human Factors in Computing Systems*, CHI EA '03, pp. 754–755, New York, NY, USA, 2003. ACM.
- [MWWM17] Martez E. Mott, Shane Williams, Jacob O. Wobbrock, and Meredith Ringel Morris. Improving Dwell-Based Gaze Typing with Dynamic, Cascading Dwell Times. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '17, pp. 2558–2570, New York, NY, USA, 2017. ACM.
- [NDA⁺17] Aanand Nayyar, Utkarsh Dwivedi, Karan Ahuja, Nitendra Rajput, Seema Nagar, and Kuntal Dey. OptiDwell: Intelligent Adjustment of Dwell Click Time. In *Proceedings of the 22nd International Conference on Intelligent User Interfaces*, IUI '17, pp. 193–204, New York, NY, USA, 2017. ACM.
- [PG16] Ken Pfeuffer and Hans Gellersen. Gaze and Touch Interaction on Tablets. In *Proceedings of the Annual ACM Symposium on User Interface Software and Technology*, UIST '16, pp. 301–311, New York, NY, USA, 2016. ACM.
- [PLW12] Abdul Moiz Penkar, Christof Lutteroth, and Gerald Weber. Designing for the Eye: Design Parameters for Dwell in Gaze Interaction. In *Proceedings of the 24th Australian Computer-Human Interaction Conference*, OzCHI '12, pp. 479–488, New York, NY, USA, 2012. ACM.
- [PSS12] Prateek Panwar, Sayan Sarcar, and Debasis Samant. EyeBoard: A Fast and Accurate Eye Gaze-based Text Entry System. In *Proceedings of 4th International Conference on Intelligent Human Computer Interaction*, IHCI '12, pp. 1–8, 2012.
- [RF61] Theodore C. Rush and Jon F. Fulton. *Medical Physiology and Biophysics*. 1961.
- [RH18] Vijay Rajanna and Tracy Hammond. A Gaze Gesture-based Paradigm for Situational Impairments, Accessibility, and Rich Interactions. In *Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications*, ETRA '18, pp. 102:1–102:3, New York, NY, USA, 2018. ACM.
- [RMSA⁺15] D. Rozado, T. Moreno, J. San Agustin, F. B. Rodriguez, and P. Varona. Controlling a Smartphone Using Gaze Gestures as the Input Mechanism. *Human-Computer Interaction*, Vol. 30, No. 1, pp. 34 – 63, 2015.

- [SD12] Sophie Stellmach and Raimund Dachselt. Look & Touch: Gaze-supported Target Acquisition. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pp. 2981–2990, New York, NY, USA, 2012. ACM.
- [SDRD17] Simon Schenk, Marc Dreiser, Gerhard Rigoll, and Michael Dorr. GazeEverywhere: Enabling Gaze-only User Interaction on an Unmodified Desktop PC in Everyday Scenarios. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '17, pp. 3034–3044, New York, NY, USA, 2017. ACM.
- [Shn86] Ben Shneiderman. Designing Menu Selection Systems. *Journal of the American Society for Information Science*, Vol. 37, No. 2, pp. 57–70, 1986.
- [TABG15] Jayson Turner, Jason Alexander, Andreas Bulling, and Hans Gellersen. Gaze+RST: Integrating Gaze and Multitouch for Remote Rotate-Scale-Translate Tasks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '15, pp. 4179–4188, New York, NY, USA, 2015. ACM.
- [ULH10] Mario H. Urbina, Maike Lorenz, and Anke Huckauf. Pies with EYES: The Limits of Hierarchical Pie Menus in Gaze Control. In *Proceedings of the 2010 ACM Symposium on Eye-Tracking Research & Applications*, ETRA '10, pp. 93–96, New York, NY, USA, 2010. ACM.
- [VBG13] Mélodie Vidal, Andreas Bulling, and Hans Gellersen. Pursuits: Spontaneous Interaction with Displays Based on Smooth Pursuit Eye Movement and Moving Targets. In *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '13, pp. 439–448, New York, NY, USA, 2013. ACM.
- [vIK⁺16] Oleg Špakov, Poika Isokoski, Jari Kangas, Deepak Akkil, and Päivi Majaranta. PursuitAdjuster: An Exploration into the Design Space of Smooth Pursuit-based Widgets. In *Proceedings of the 2016 ACM Symposium on Eye Tracking Research & Applications*, ETRA '16, pp. 287–290, New York, NY, USA, 2016. ACM.
- [WM87] Colin Ware and Harutune H. Mikaelian. An Evaluation of an Eye Tracker As a Device for Computer Input. In *Proceedings of the SIGCHI/GI Conference on Human Factors in Computing Systems and Graphics Interface*, CHI '87, pp. 183–188, New York, NY, USA, 1987. ACM.
- [WMK03] Jacob O. Wobbrock, Brad A. Myers, and John A. Kembel. EdgeWrite: A Stylus-based Text Entry Method Designed for High Accuracy and Stability of Motion. In *Proceedings of the Annual ACM Symposium on User Interface Software and Technology*, UIST '03, pp. 61–70, New York, NY, USA, 2003. ACM.

- [WRSD07] Jacob O. Wobbrock, James Rubinstein, Michael Sawyer, and Andrew Duchowski. Not Typing but Writing: Eye-based Text Entry Using Letter-like Gestures. In *Proceedings of the 3rd Conference on Communication by Gaze Interaction, COGAIN '07*, pp. 61–64, 2007.
- [WRSD08] Jacob O. Wobbrock, James Rubinstein, Michael W. Sawyer, and Andrew T. Duchowski. Longitudinal Evaluation of Discrete Consecutive Gaze Gestures for Text Entry. In *Proceedings of the 2008 ACM Symposium on Eye Tracking Research & Applications, ETRA '08*, pp. 11–18, New York, NY, USA, 2008. ACM.
- [WWL07] Jacob O. Wobbrock, Andrew D. Wilson, and Yang Li. Gestures Without Libraries, Toolkits or Training: A \$1 Recognizer for User Interface Prototypes. In *Proceedings of the Annual ACM Symposium on User Interface Software and Technology, UIST '07*, pp. 159–168, New York, NY, USA, 2007. ACM.

著者論文リスト

本論文に関する論文および発表

- 査読なし国内会議論文

1. Toshiya Isomoto, Shota Yamanaka, and Buntarou Shizuki. Gaze-based Command Activation Technique using Two-level Stroke. Proceedings of the 2019 ACM CHI symposia on ASIAN CHI SYMPOSIUM: EMERGING HCI RESEARCH COLLECTION (Asian CHI Symposium 2020), 6 pages, ACM, New York, NY, USA, Honolulu, Hawaii, April 2020. (to appear)
2. 磯本俊弥, 山中祥太, 志築文太郎. 2段階の視線移動を用いたコマンド実行手法. 情報処理学会研究報告, Vol.2019-HCI-182 No.30, 8 pages, 情報処理学会, 東京, 2019年3月. 学生奨励賞受賞.

その他の論文および特許

- 査読あり国際会議論文

1. Toshiya Isomoto, Shota Yamanaka, and Buntarou Shizuki. Gaze-based Command Activation Technique using Two-level Stroke. Proceedings of the 2019 ACM CHI symposia on ASIAN CHI SYMPOSIUM: EMERGING HCI RESEARCH COLLECTION (Asian CHI Symposium 2020), 6 pages, ACM, New York, NY, USA, Honolulu, Hawaii, April 2020. (to appear)
2. Kyohei Hakka, Toshiya Isomoto, Buntarou Shizuki, Shin Takahashi. Bounded Swipe: Swipe Gesture Inside a Target. Proceedings of the 30th Australian Conference on Computer-Human Interaction (OzCHI 2019), pp. 312–316, ACM, New York, NY, USA, Perth/Fremantle, Australia, December 2019.
3. Kyohei Hakka, Toshiya Isomoto, Buntarou Shizuki. One-Handed Interaction Technique for Single-Touch Gesture Input on Large Smartphones. Proceedings of the ACM Spatial User Interaction 2019 (SUI 2019), 2 pages, ACM, New York, NY, USA, New Orleans, USA, October 2019.
4. Toshiya Isomoto, Toshiyuki Ando, Buntarou Shizuki, and Shin Takahashi. Investigation of Midas-touches in Dwell Time Reduction Technique using Fitts' Law for Dwell-Based Target Acquisition. Proceedings of the 2019 ACM CHI symposia on ASIAN CHI SYMPOSIUM: EMERGING HCI RESEARCH COLLECTION (Asian CHI Symposium 2019), 8 pages, ACM, New York, NY, USA, Glasgow, UK, May 2019.
5. Toshiyuki Ando, Toshiya Isomoto, Buntarou Shizuki, and Shin Takahashi. One-handed Rapid Text Selection and Command Execution Method for Smartphones. Proceedings of Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems (CHI EA 2019), 6 pages, ACM, New York, NY, USA, Glasgow, UK, May 2019.
6. Toshiyuki Ando, Toshiya Isomoto, Buntarou Shizuki, and Shin Takahashi. One-handed Rapid Text Selection and Command Execution Method for Smartphones. Proceedings of the 30th Australian Conference on Computer-Human Interaction (OzCHI 2018), pp. 401–405, ACM, New York, NY, USA, Melbourne, Australia, December 2018.
7. Toshiya Isomoto, Toshiyuki Ando, Buntarou Shizuki, and Shin Takahashi. Dwell Time Reduction Technique using Fitts' Law for Gaze-Based Target Acquisition. Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications (ETRA 2018), pp. 26:1–26:7, ACM, New York, NY, USA, Warsaw, Poland, June 2018.
8. Ryosuke Takada, Toshiya Isomoto, Wataru Yamada, Hiroyuki Manabe, and Buntarou Shizuki. ExtensionClip: Touch Point Transfer Device Linking Both Sides of a Smartphone for Mobile VR Environments. Proceedings of Extended Abstracts of the 2018

CHI Conference on Human Factors in Computing Systems (CHI EA 2018), 6 pages, ACM, New York, NY, USA, Montréal, Canada, April 2018.

9. Toshiya Isomoto, Akira Ishii, Shuta Nakamae, and Buntarou Shizuki. Target Selection Technique Using Space Below Cardboard VR Goggles. Proceedings of the 2018 ACM CHI symposia on ASIAN CHI SYMPOSIUM: EMERGING HCI RESEARCH COLLECTION (Asian CHI Symposium 2018), 8 pages, ACM, New York, NY, USA, Montréal, Canada, April 2018.

- 査読あり国内会議論文

1. 八箇恭平, 儀本俊弥, 志築文太郎. ターゲット内に両端が存在するスワイプジェスチャ. 第27回インタラクティブシステムとソフトウェアに関するワークショップ(WISS 2019), pp. 61–66, 日本ソフトウェア科学会, 長野, 2019年9月.

- 査読なし国内会議論文

1. 川口航平, 儀本俊弥, 志築文太郎, 高橋伸. VR向けの掌上における日本語フリック入力手法の提案. ヒューマンインタフェースシンポジウム2019論文集, pp. 676–682, ヒューマンインタフェース学会, 京都, 2019年9月.
2. 高田 峻介, 儀本俊弥, 山田 渉, 真鍋 宏幸, 志築文太郎. プロペラを用いた頭部装着型歩行牽引デバイス. 第23回一般社団法人情報処理学会シンポジウムインタラクシオン2019, pp. 236–237, 情報処理学会, 東京, 2019年3月.
3. 安藤宗孝, 儀本俊弥, 志築文太郎, 高橋伸. ソフトウェアキーボードのキーに基づく文字列操作手法. 第26回インタラクティブシステムとソフトウェアに関するワークショップ(WISS2018), 日本ソフトウェア科学会, 山梨, 2018年9月(予稿なし).
4. 儀本俊弥, 安藤宗孝, 志築文太郎. 近接センサおよび照度センサを用いたスマートフォンベースHMD向けの操作手法. 第26回インタラクティブシステムとソフトウェアに関するワークショップ(WISS2018), 2 pages, 日本ソフトウェア科学会, 山梨, 2018年9月.
5. 安藤宗孝, 儀本俊弥, 志築文太郎, 高橋伸. スマートフォンにおける傾きを利用した文字列操作手法. 報処理学会研究報告, Vol.2018-HCI-179, 7 pages, 情報処理学会, 京都, 2018年8月.
6. 儀本俊弥, 安藤宗孝, 志築文太郎, 高橋伸. フィッツの法則に基づく視線を用いたターゲット選択システム. 第25回インタラクティブシステムとソフトウェアに関するワークショップ(WISS2017), 日本ソフトウェア科学会, 山梨, 2017年12月(予稿なし).
7. 儀本俊弥, 宮崎亮一. 音声認識システムおよび顔の情報をを用いた上肢不自由者のためのテキスト作成アプリケーションの開発. 電子情報通信学会技術研究報告, 115(359), pp. 59–62, 電子情報通信学会, 石川, 2015年12月.

8. 礒本俊弥, 宮崎亮一. 顔の方向情報を用いたテキスト編集補助アプリケーションの開発. 第21回日本高専学会年会講演会講演論文集, 高専学会, 山口, 2015年8月(予稿なし).

• 出願済み特許

1. 真鍋宏幸, 山田渉, 高田峻介, 礒本俊弥, 志築文太郎. 歩行制御装置. 特願2019-142453(出願中)
2. 真鍋宏幸, 山田渉, 高田峻介, 礒本俊弥, 志築文太郎. 入力装置. 特開2019-185217, 2019年10月24日公開