

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

フリックと視線移動の組み合わせによる
スマートフォンの操作拡張

鈴木 雄太郎

修士（工学）

（コンピュータサイエンス専攻）

指導教員 高橋 伸

2021年3月

概要

スマートフォンを使用するユーザは片手の少なくとも一部を、スマートフォンを安定させるために使用するため、手を用いる入力には制限されてしまうことがある。本論文において、フリックと視線移動を組み合わせたスマートフォン向けの新たな入力手法である *FlickSight* を提案する。ユーザは、視線移動を2本目の指による入力と見なすことにより、片手操作時であってもピンチインなどのマルチタッチジェスチャの代替となる操作を実行できる。また、把持姿勢を変更しないまま物理ボタンやベゼルを利用するような操作を実行できる。これを実現するために、スマートフォンを用いた視線推定システムを実装し、その性能を調査するための予備実験を行った。続いて、*FlickSight* について説明し、入力手法を設計するために2つの予備実験を行った。2つの予備実験から、*FlickSight* が通常のフリックと競合しないための閾値を設定し、ユーザにとって視線を移動させやすい方向を調査した。以上の結果を元に、*FlickSight* の認識アルゴリズムを実装し、*FlickSight* の実現可能性を調査するための評価実験を行った。結果として、平均で69.0%の成功率を得たが、下方方向への視線移動を視線移動なし、および視線移動なしを上方向への視線移動とする誤認識が頻繁に確認された。これらは、顔とスマートフォンの相対的な位置関係、および瞬きが影響していると考えられる。またフリックに要する時間が短い場合に、フリック後の視線移動を考慮する補正を加えることによって80.8%の成功率を得た。

目次

第1章	序論	1
1.1	スマートフォンの操作手法とその問題点	1
1.2	視線を用いるコンピュータの操作手法	2
1.3	目的と提案手法	3
1.4	本研究の貢献	5
1.5	本論文の構成	5
第2章	関連研究	6
2.1	スマートフォンの操作拡張	6
2.2	視線によるスマートフォンの操作手法	7
2.3	視線と手を組み合わせたコンピュータの操作手法	8
2.4	本研究の位置付け	9
第3章	スマートフォンによる視線推定システムの実装	10
3.1	視線推定に用いるデバイスおよびフレームワーク	10
3.2	視線推定アルゴリズム	11
3.3	予備実験1：実装した視線推定アルゴリズムの精度	14
3.3.1	参加者および実験環境	14
3.3.2	タスク	14
3.3.3	結果と考察	15
第4章	<i>FlickSight</i> の設計	18
4.1	<i>FlickSight</i>	18
4.1.1	フリックの方向と視線移動の方向を任意の数に設定	19
4.1.2	フリックの方向を基準とした視線移動の方向を利用	21
4.2	予備実験2：フリック中に発生する視線移動の観察	22
4.2.1	参加者および実験環境	22
4.2.2	タスクと実験手順	23
4.2.3	結果	24
	フリック	24
	視線移動	24
4.3	予備実験3：意図した直線的な視線移動の観察	26

4.3.1	参加者および実験環境	26
4.3.2	タスクと実験手順	26
4.3.3	視線移動の検出	28
4.3.4	視線移動の分析および議論	29
	直線距離	30
	角度の誤差	31
第 5 章	<i>FlickSight</i> の評価実験	34
5.1	<i>FlickSight</i> 認識アルゴリズムの実装	34
5.1.1	入力	34
5.1.2	<i>FlickSight</i> の分類	35
5.2	評価実験	35
5.2.1	参加者および実験環境	35
5.2.2	タスクと実験手順	36
5.2.3	結果	37
	フリックに要する時間	38
	視線移動の距離	41
	視線移動の方向	42
	瞬き	42
5.2.4	フリックに要する時間が短い場合の補正による成功率の向上	44
第 6 章	<i>FlickSight</i> を用いた操作例	46
6.1	2本の指を用いる操作への割り当て	46
6.2	ベゼルや物理ボタンを用いる操作への割り当て	47
6.3	画面内における指の届きづらい範囲にて実行される操作への割り当て	48
6.4	フリックの拡張	49
6.5	メニューの多いアプリケーションにおけるショートカット	49
第 7 章	今後の課題	50
7.1	<i>FlickSight</i> の視線移動距離	50
7.2	指と視線を意図的に異なる方向へ動かすことによる負担	51
7.3	実験環境および実験参加者の偏り	51
第 8 章	結論	52
	謝辞	53
	参考文献	54

目次

1.1	<i>FlickSight</i> を実行するイメージ図.	4
3.1	ARKitにより構築された拡張現実空間におけるデバイスの位置および顔の位置.	11
3.2	顔空間における両目の位置および視線ベクトル.	12
3.3	実装した視線推定アルゴリズムによる視点の算出.	13
3.4	スクリーン平面の座標系	13
3.5	予備実験1に用いたアプリケーション画面のスクリーンショット.	15
4.1	フリックの方向と視線移動の方向を任意の数に設定した入力手法.	19
4.2	<i>FlickSight-Follow</i> および, <i>FlickSight-Reverse</i> のうち, 上方向へのフリックを基準とする入力の例.	21
4.3	予備実験2に用いたアプリケーション画面のスクリーンショット.	23
4.4	予備実験2にてフリック中に観察された視線移動の直線距離の対数分布.	25
4.5	予備実験3にて指示される視線移動の方向と角度の対応.	27
4.6	予備実験3に用いたアプリケーション画面のスクリーンショット.	27
4.7	予備実験3における視線移動検出の概略図.	28
4.8	予備実験3における視線移動終了の条件.	29
4.9	予備実験2にて記録したフリック中の視線移動を視線移動ではないと識別する確率(青), 予備実験3にて検出した視線移動を視線移動であると識別する確率(赤), およびそれらの合計(緑).	31
4.10	予備実験3にて検出した視線移動の角度の誤差に対する影響.	32
4.11	予備実験3にて検出した視線移動の角度の誤差の絶対値に対する影響.	33
5.1	フリックと視線情報の取得.	34
5.2	ジェスチャ方向の分類.	35
5.3	実験時に <i>FlickSight</i> の指示を行うアプリケーションの表示例. (a) 本番セッション. (b) 訓練セッション(直前の <i>FlickSight</i> が正解だった場合). (c) 訓練セッション(直前の <i>FlickSight</i> が不正解だった場合).	36
5.4	各参加者のセッション毎の成功率および平均の成功率.	37
5.5	フリックに要した時間の対数分布.	39
5.6	指示された視線移動の方向毎のフリックに要した時間(対数表記).	40
5.7	フリックに要した時間の対数分布.	40

5.8	視線移動の方向毎の混同行列.	41
5.9	視線移動の方向毎の視線移動距離に対する影響. 左) 視線移動の方向毎の距離のグラフ. 右) Tukey-Kramer 法による多重比較の結果.	42
5.10	補正を行った際の視線移動の方向毎の混同行列.	45
6.1	<i>Flick-Up-Sight-Reverse</i> によって画像の縮小を行っている例.	46
6.2	<i>Flick-Up-Sight-Follow</i> によって起動中のアプリケーションを確認する画面へ遷移する例.	47
6.3	<i>Flick-Left-Sight-Right</i> によってスクリーンショットの撮影を行う例.	48

第1章 序論

スマートフォンと呼ばれる，タッチスクリーン及び様々なセンサを搭載した情報端末が広く普及しており，その操作は多くの場合タッチスクリーンへのタッチによって行われる．タッチによる操作は主に操作対象となる文字列や画面内のオブジェクトを選択する際に用いられる．また，タッチした指の動き（以降，タッチジェスチャ）を利用することによって選択した対象を操作するための入力の種類（以降，入力語彙）を増やしている．市販製品ではその他にも，スクリーン外部の物理ボタンあるいはタッチスクリーンの枠（以降，ベゼル），音声入力など，複数の入力モダリティを活用することにより入力語彙を増加させており，最近では視線を用いて操作することも研究されている [KAB18]．本論文では，スマートフォンの操作における現状の問題点を示し，この問題点を解決するためにタッチジェスチャと視線移動を組み合わせた入力手法を提案する．さらに，提案した入力手法の評価を行う．

本章ではまず，主なスマートフォンの操作方法およびその問題点について述べる．続いて，問題解決のために用いる，スマートフォンを含むコンピュータを視線により操作する研究の概要，およびその問題点を示す．その後，これらの問題点を解決するための手法を提示し，最後に本論文の構成を述べる．

1.1 スマートフォンの操作手法とその問題点

本節ではスマートフォンの操作手法として代表的なものとその問題点について述べる．

スマートフォンにはタッチスクリーンが搭載されており，多くの操作はタッチスクリーンへのタッチ，あるいはタッチジェスチャによって行われる．タッチによる操作はボタンあるいはアプリケーションなどを選択する際に用いられる．タッチジェスチャにはタッチを開始した位置から移動した点にてタッチを終了するフリックや，2本の指をつまむように動かすピンチイン/アウトなどがある [Inc20b]．フリックはキーボード入力，ページめくり，スクロールなどの操作に用いられることが多く，ピンチイン/アウトは縮小/拡大に割り当てられることが多い．しかし，スマートフォンを使用するユーザは片手の少なくとも一部を，スマートフォンを安定させるために使用するため，手を用いる入力は制限されてしまうことがある．

両手にて操作する場合には，一方の手を自由に動かすことができるため，複数の指を用いる操作などを比較的自由に行うことができる．一方で，タッチによってスマートフォンを操作する際には，指一本による操作であっても，指によって画面が隠れてしまうファットフィンガー問題 [SRC05, BLC⁺12] と呼ばれる問題に遭遇することがある．そのため，ラップトップPCに搭載されているようなタッチパッドにて利用されるような3本以上の指を用いるマ

マルチタッチジェスチャは、スマートフォン操作において利用されることは少ない。スマートフォン操作の際に複数の指を用いる場合には、多くの場合2本の指を用いるが、シングルタッチと比較して、指が隠す画面の範囲が広がることは明らかである。また、荷物を持っている場合や、電車内にてつり革に掴まっている場合など、日常生活において両手によるスマートフォン操作は制限されることが多い。

片手にて操作する場合には、親指のみによる操作が好まれている [NBW14, KBCV06]。これは、複数の指を用いるマルチタッチジェスチャを利用しないということであり、指一本による操作時の入力語彙拡張が求められると考えられる。また、ベゼルあるいは物理ボタンを用いる操作では、スマートフォンの把持姿勢によってはベゼルあるいは物理ボタンに指が届かないために、把持姿勢を調整する必要が生じる可能性がある。しかし、頻繁なスマートフォンの把持姿勢変更は把持の安定を損なうため、ユーザの快適な操作を妨げることが報告されている [ERGT17, ERGT18]。加えて、スマートフォンの把持姿勢変更は、スマートフォンの落下につながる危険性がある。

タッチによる操作の代替手段の一つとして、音声による操作がある。これは、ユーザがスマートフォンに搭載されたマイクに話しかけることによって、操作を行うものである。音声による操作は、ハンズフリー、及びアイズフリーでの操作を可能とし、運転中などの両手が塞がっており、視線をスマートフォンに向けられない環境においても操作できるという利点がある。一方で音声による操作は、電車内などの混雑している状況や騒がしい環境、あるいは図書館などの発声が憚られる環境に不向きであるという問題点を抱えている [PPJ03]。

1.2 視線を用いるコンピュータの操作手法

視線を用いるコンピュータの操作手法とは、ユーザの視線情報のみを用いる操作を指し、ユーザは画面上の対象を一定時間以上見続けることにより操作対象として選択することができる。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]。また、視線をある特定の軌跡に沿って移動させることにより、その軌跡（視線ジェスチャ）に紐づくコマンドを実行することも可能である [MLGH10, ULH10, DHI17]。特にスマートフォンの操作環境において視線は、デバイスを持つために使用する手と比較して自由に動かしやすいことや、調理中や運転中などの手が塞がっている状態でも操作できる可能性があること、指の届きづらい範囲にも容易に到達することができるといったメリットがあり、視線によるスマートフォン操作が幅広く研究されている [KAB18]。

視線を用いる操作手法には、ユーザが意図しない操作が発生してしまうことがある。手を用いる操作では、ユーザがマウスをクリックする、あるいはタッチスクリーンに触れるなどの明示的な動作が発生しない限り操作は行われませんが、視線を用いる操作ではユーザが画面をただ見ていたとしても、凝視が検出され、操作が行われてしまうことがある。このように、ユーザが意図していない場面で凝視が認識され、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、凝視を用いる操作手法において大きな問題とされている [JK03, IBHV08]。

また、視線ジェスチャを用いる手法でも同様の問題が発生する可能性がある。例えば、視線を左から右へ移動するようなジェスチャにコマンドを紐付けたとすると、ただ文章を読んでいるユーザの視線移動が、操作を行うための視線移動として検出されてしまい、コマンドが実行されてしまう。つまり、視線を用いる操作手法を提案する場合には、意図しない操作を防ぐこと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。凝視を用いる操作手法は、対象を凝視する時間を長く設定することによって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るため、ユーザビリティが低下する。そのため、短い凝視時間にて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いる [IAST18, BVH12, NDA⁺17, PLW12, MWWM17]。また、ジェスチャを用いる操作手法では、日常的に発生し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って、ユーザが意図的に視線を移動させたときのみコマンドを実行することが提案されている [DHI17, DS07]。複雑な視線移動を採用することによって意図しない時の発生を抑制できる一方で、実行が難しく、ガイドが必要であったり、時間がかかることもある。

1.3 目的と提案手法

本研究の目的は、一本の指によるスマートフォンの操作を視線の動きと組み合わせることにより拡張することである。そのための手法として、フリックと視線移動を組み合わせたスマートフォン向けの入力手法である *FlickSight* を提案する。図 1.1 に *FlickSight* を実行するイメージ図を示す。図中にある橙色の点は視点を示しており、矢印は上方向へ視線を移動させている様子を示している。また、緑色の矢印は上方向へフリックしている様子を示しており、これらは同時に行われている。

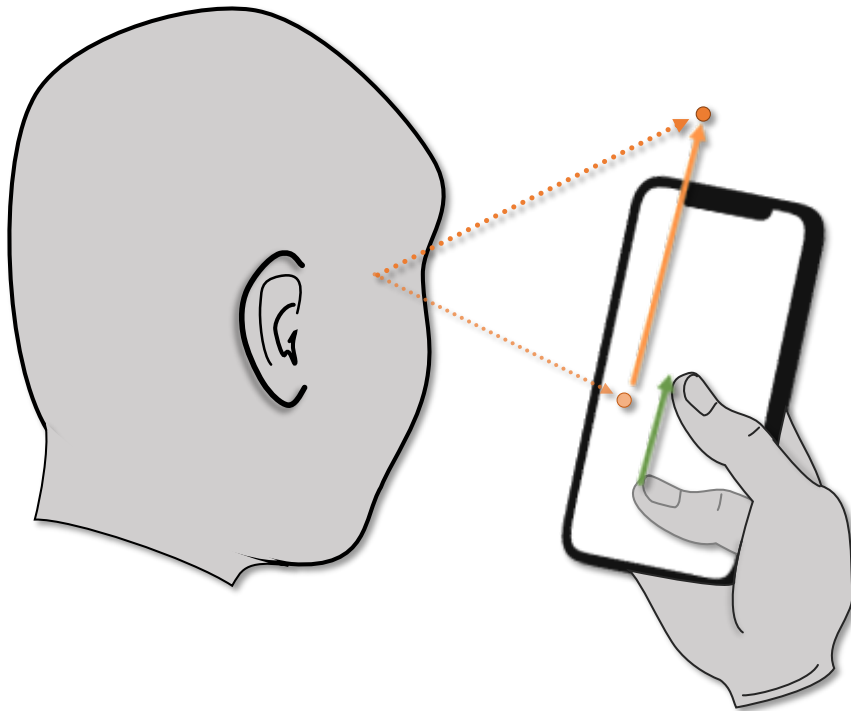


図 1.1 *FlickSight* を実行するイメージ図. 橙色の円は視点を示しており, 橙色の矢印が示す上方向へ視線を移動させている様子を示している. 緑色の矢印は上方向へフリックしている様子を示しており, これらは同時に行われている.

FlickSight はフリックと視線移動を同時に実行する入力手法であり, それぞれを任意の方向に動かすことによって入力語彙を拡張する. ここでフリックとは, タッチを開始した位置から短く移動した点にてタッチを終了するタッチジェスチャの一つであり, 一般的に一本の指にて実行され, 上下左右の方向によって4種類に分類される. 一本の指にて実行されることから片手操作および両手操作中に実行可能であり, 実行時間の観点については単なるタッチと大きく変わらず, 短い時間にて実行可能であると言える. フリックの開始位置に制約はないため, キーボードなどを操作する場合には, フリックの開始位置にも入力としての意味を持たせることが可能である. また, *FlickSight* は直線的な視線移動, 特に開始位置に依存しない視線移動を採用する. これはユーザにとって, 高速に実行可能, かつ視線を移動させる軌跡にガイドが不要といった利点がある. 直線的な視線移動はユーザが意図しない場合にも認識されやすいが, *FlickSight* はフリックと視線移動を同時に実行した時のみ認識されるため, 意図しない場合の入力としての認識を抑制できると考えられる.

FlickSight に採用する視線移動の方向には二つの設計方針がある. いずれの方針も, 通常の(視線移動のない)フリックと競合しないことが想定される. 一つ目の設計方針は, フリックの方向によらず任意の方向数を設定する方針であり, 例えば4方向に設定した場合には上下左右方向への視線移動が含まれる. この方針では, 方向数によって豊富な入力語彙を用意する

ことが可能となるため、システムショートカットやアプリケーション内メニューなどに割り当てることができると考えられる。また、把持姿勢の変更を伴うことのあるベゼルや物理ボタンなどを用いる操作の代替として利用できる可能性がある。二つ目の設計方針は、フリックの方向に依存する方針であり、フリックの方向と同じ方向および逆方向の二種類を含む。これらのジェスチャには、視線移動を2本目の指による入力と見なすことにより、ピンチインなどのマルチタッチジェスチャの代替となる操作を実行できる可能性がある。また、スクロール速度の調整など、スクロール操作を拡張するような特有の操作を実現できる可能性がある。

1.4 本研究の貢献

本研究の貢献は以下の通りである。

- フリックと視線移動を組み合わせた入力手法である *FlickSight* を提案した。
- *FlickSight* を設計するために、日常的なスマートフォン操作環境におけるフリック中の視線移動距離を明らかにした。
- *FlickSight* を設計するために、直線的な視線移動について、ユーザが望む移動距離の長さ、およびユーザが実行可能な角度の正確性を明らかにした。
- ユーザが着席した状態における *FlickSight* の入力性能を明らかにした。

1.5 本論文の構成

本章では本論文の背景、目的および提案する入力手法の概要について述べた。第2章では関連研究について説明し、本研究の位置付けを明らかにする。第3章では提案手法を実現するために実装したスマートフォンによる視線推定システムについて説明する。第4章では提案する入力手法について説明し、入力手法を設計するために行った予備実験について述べる。第5章では提案する入力手法の実現可能性を調査するための評価実験について述べる。第6章では提案する入力手法を用いた操作の例を説明する。第7章では今後の課題について述べる。最後に第8章にて結論を述べる。

第2章 関連研究

本研究は、一本の指によるスマートフォンの操作を視線の動きと組み合わせることにより拡張することを目的とする。目的を達成するためのアプローチとして、フリックと視線移動を組み合わせることによる入力手法である *FlickSight* を提案する。そのため本研究に関連する研究として、「スマートフォンへの入力語彙拡張」、「視線によるスマートフォンの操作手法」および「視線と手を組み合わせたコンピュータの操作手法」がある。本章では、これらの研究に関して述べた後に、本研究の位置付けを示す。

2.1 スマートフォンの操作拡張

スマートフォンに対する入力語彙あるいは操作を拡張するための手法は、これまでに数多く提案されている。始めに、タッチから取得可能な情報を用いて入力語彙を拡張する研究に言及する。Boringら [BLC⁺12] は、スクリーンと接触した指の面積の変化を用いることによって入力語彙を拡張することを提案した。Wangら [BLC⁺12, WCRI09] は、タッチする指の角度を用いることを提案した。Heoら [HL12]、および McCallumら [MMIS09] は、タッチの強さを入力に用いた。また、Harrisonら [HSH11] は、タッチ時に発生する音を、池松ら [池松 19] はタッチによって発生する振動を用いることを提案した。これらの手法は、通常のタッチによる入力と遜色ない時間にて入力することが可能である。

タッチに用いる部位を識別することによって入力語彙を拡張する研究も行われている。Leら [LKB⁺18] は、ユーザが手のひらによって画面に触れたことを識別する手法であり、手のひらを用いたタッチによって入力語彙を拡張する。Ikematsuら [IY20a, IY20b] は、爪によるタッチを識別し、爪を用いたタッチジェスチャによって入力語彙を拡張する。

画面へのタッチ以外のジェスチャを用いる手法では、それらの動きを認識し、タッチと組み合わせることによって入力語彙を拡張することが提案されている。Hasanら [HAKI17, HKAI16] は、ユーザがスマートフォンを把持している手の親指、あるいはスマートフォンを把持していない手を用いたジェスチャによる操作を提案した。Seiら [SFS19] は、指輪型デバイスを用いてスマートフォン背面におけるユーザの人差し指の動きを認識することによって入力語彙を拡張した。これらの研究は、スマートフォン周辺における手や指の動きを認識するために、追加のハードウェアあるいはセンサが必要である。画面へのタッチ以外のジェスチャと組み合わせることにより入力を拡張している点にて本研究と似ているが、本研究は視線を利用している点、および視線を認識するために追加の機器を用いない点でこれらの研究と異なる。また、Voelkerら [VHCR20] は、頭の向きを認識することによって、スマートフォンにお

けるターゲット選択を拡張した。頭の向きを認識する際に追加の機器を使用しないが、入力モダリティとして頭を用いる点にて本研究とは異なる。

日常的な操作中に発生しにくい指の動作を新たなジェスチャとして用いる研究も行われている。Rothら [RT09] は、スマートフォンのベゼルからターゲットへのスワイプという新たなジェスチャをカットやコピーなどの操作に割り当てることによって操作を拡張した。Liら [LF13] は、スマートフォンのベゼルからのスワイプをトリガとした、カーソルを用いた片手操作時のターゲット選択手法を提案した。Heoら [HL13] は、離れた2点を連続してタップするジェスチャを提案し、マルチタッチ操作やメニューの呼び出しなどに割り当てることを検討した。Roudautら [RLG09] は、画面をタッチしている親指を回転させるジェスチャを提案した。Hakkaら [HIST19, 八箇19] は、既存のスワイプと競合しづらいジェスチャとして、開始点および終了点の両方がターゲット内にあるスワイプを認識することによって入力語彙を拡張した。

2.2 視線によるスマートフォンの操作手法

スマートフォンを視線によって操作するための研究は、これまでに数多く行われている。それらの一種として、凝視している位置、あるいは画面を見ている状態などの暗黙的な視線の動きを用いる手法が提案されている。Pfeiffer [Pfe18] は、凝視を用いたタイピングを提案した。また、Songら [SWRX16] は、スクリーン上の凝視点を推測することによる生体認証を提案した。Mariakakisら [MGA⁺15] によって提案された SwitchBack は、ユーザが画面に注視しているかを認識することによって、ユーザが画面に視線を戻したときにタスクを再開できるシステムである。Wachnerら [WEB18] は、モバイルデバイスに組み込まれたカメラを用いて、四肢の欠損や麻痺が原因で手による入力が困難な人向けの凝視ベースのインタラクションを可能にするミドルウェアを開発した。これらの凝視点を利用する多くのアプローチはキャリブレーションを必要とする [HLNL17], もしくはキャリブレーションを行うことによって大幅な精度向上が見られる [KKK⁺16]。しかし、システムが頻繁にキャリブレーションを求めるとは、ユーザに不快感を与える可能性があると考えられる。

キャリブレーションを必要としない手法として、凝視時の目の特性を利用することが提案されている。Vidalら [VBG13] は、円滑性追跡眼球運動を利用することにより、動く対象の注視を検出した。また、Kirstら [KB16] は、自発的視線収束を利用することによって、注視による視線入力のタイミングを正確にすることを提案した。Jalaliniyaら [JM16] は、視覚運動性眼振を利用することによって、注視を検出した。このように、キャリブレーションを必要としない研究が提案されていることを踏まえ、本研究でもキャリブレーションを必要としない手法を試みる。

より明示的な視線移動を用いる手法もまた、数多く研究されている。Valitukaitisら [VB12] は、ハンドヘルド型ポータブルデバイスのみを用いて、カメラ正面で行われた4つの視線ジェスチャ (*Diamond, Plus, Vertical, Horizontal*) を認識した。Zhangら [ZKM17] は、スマートフォンの背面カメラを用いて、上下左右への視線移動を検出することによる文字入力を可能

にした。Drewes ら [DDLS07] は、モバイルデバイス環境において、予め定めた図形に沿って視線を動かすことによるコマンド実行について調査を行った。Majaranta ら [MLKI19] は、視線を動かす方向および、動かした結果実行されるコマンドを画面内にイラストとして表示することによって、ユーザがジェスチャおよびジェスチャとコマンドの対応を記憶する労力を低減した。これらの手法では、意図的に目を動かすために自然な目の振る舞いを利用するよりも認知的努力が必要となる一方で、凝視推定の不正確性に対しては堅牢であるため、明示的な視線移動を用いる手法はより有望な手法であることが明らかになっている [BG10]。

2.3 視線と手を組み合わせたコンピュータの操作手法

視線と手を組み合わせたコンピュータの操作手法は、ラップトップコンピュータやウェアラブルデバイスなど、多様なデバイスを操作するために研究されてきた。

始めに、ディスプレイがタッチスクリーンを搭載していない環境における研究について説明する。大型ディスプレイを対象とした研究として、Stellmach ら [SD12] は、ユーザの視線とハンドヘルドデバイスへのタッチ入力を組み合わせることによって、視線による選択の支援技術を提案した。また、Turner ら [TABG15] は、大型ディスプレイ上の回転、スケーリング、平行移動タスクを滑らかに実行するために、視線とマルチタッチを使用することを調査した。Chatterjee ら [CXH15] は、デスクトップ環境における視線とハンドジェスチャを補完的に用いるインタラクション技術である Gaze+Gesture を提案した。Kumar ら [KHMS20] が提案した TAGSwipe は、視線によるキーの選択モードと、単に画面を見ている観察モードを、タッチによって区別することによって、デスクトップ環境における視線によるテキスト入力を実現した。なお、TAGSwipe は、デスクトップ PC に接続したスマートフォンにてタッチ入力を取得した。

続いて、タッチスクリーンを搭載したディスプレイを利用する研究について説明する。Pfeuffer ら [PG16] は、タブレット上におけるタッチと視線を組み合わせた操作を提案した。これは、タッチによる操作を注視点に転送するもので、単純なタッチや、注視点周辺におけるカーソル操作が可能である。特にカーソル操作は通常のドラッグジェスチャなどと衝突してしまうため、タブレット端末上におけるタッチ位置と凝視位置の距離によってその意図を区別した。Rivu ら [RAM⁺19] は、タブレット端末において、画面上の UI ボタンをタッチ、あるいは凝視することにより入力モードを切り替えて、一つのボタンを用いて画面全体とインタラクションできる GazeButton を提案した。Nagamatsu ら [NYS10] が提案した MobiGaze は、視線による操作とタッチによる操作を組み合わせることによって、モバイルデバイスを操作することを提案した。この操作は、視線入力の意図を確認するためにタッチを用いており、タッチした際に凝視している対象が選択されるというものである。Elleuch ら [EWA14, EWA16] は、上下左右への直線的な視線ジェスチャと頭の向きを利用した操作手法を提案した。また、Hanene ら [EWSA17] は、フロントカメラによって撮影された映像から、上下左右への視線ジェスチャおよび手の形状を認識することによって、モバイルデバイスを操作することを提案した。Khamis ら [KHZ⁺17, KHBA17, KAH⁺16] は、タッチと視線を組み合わせることによる個人認証手法

を提案した。Wang ら [WG20] は、瞬きとタッチおよび瞬きとフリックを同時に実行するジェスチャである BlyncSync を提案し、スマートウォッチに対する入力語彙を拡張した。

2.4 本研究の位置付け

本研究では、スマートフォンにおけるフリックと視線移動を組み合わせることによる入力手法を提案する。本研究にて提案する入力手法は、日常的なフリック中に発生しにくい視線移動を用いる。これは、日常的に発生しづらい指の動きを用いることによってスワイプを拡張する Roth ら [RT09]、および Hakka ら [HIST19, 八箇 19] の研究と似ている。しかし、Roth らはスワイプの開始点をベゼルに、Hakka らはスワイプの開始点および終了点をターゲット内に限定する一方、本研究の提案手法はフリックの開始点および終了点を限定しないという点にてこれらの研究と異なる。また、2.1 節にて説明した手法は、本研究の提案手法と競合しないため、共存できる可能性があると考えられる。

フリックと目の動きを組み合わせる研究として BlyncSync [WG20] があるが、これは瞬きとフリックを組み合わせしており、視線移動とフリックを組み合わせる本研究とは異なる。また、BlyncSync はウェアラブルデバイスを対象としている点にて、スマートフォンを対象とする本研究と異なる。なお、BlyncSync をスマートフォン向けに実装した場合には競合することがないため、共存できる可能性があると考えられる。

第3章 スマートフォンによる視線推定システムの実装

本研究では、フリックと視線移動を同時に実行する入力手法である *FlickSight* を認識するために、ユーザの視点を推定する必要がある。本章では、視線推定を行うシステムの実装に用いるデバイスとフレームワークの選定について述べる。その後、実装に用いるフレームワークを利用した視線推定アルゴリズムについて説明する。最後に、実装した視線推定システムの精度を評価するための予備実験について説明する。

3.1 視線推定に用いるデバイスおよびフレームワーク

視線推定に用いるデバイスとして、既存の商用スマートフォンである iPhone X を利用する。近年、商用スマートフォンに搭載されたインカメラの性能が向上したことにより、追加のカメラなどを使用することなく視線追跡が可能になった。これは研究室外での調査を可能にし、生態学的に妥当性の高い調査結果を得ることにつながるという利点がある [KAB18]。これを踏まえて本研究では、既存の商用スマートフォンである iPhone に着目した。特に、iPhone X 以降の機器は True Depth カメラと呼ばれる、顔による個人認証などに用いられるセンサ群を搭載している。そのため iPhone X を用いることにより、高性能ビデオカメラなどの追加機器を用いることなく、本研究にて必要とする視線推定が可能になると考えた。また iPhone X にはタッチスクリーンが搭載されており、フリックを検知することも可能である。

iPhone によって視線位置を推定するためのフレームワークとして、ARKit [Inc20a] を採用する。ARKit は、iPhone の背面カメラを利用して周辺環境に対する拡張現実を構築したり、インカメラを利用して顔の向きや表情などを用いた拡張現実を構築できる。なおインカメラを利用して顔の情報を取得するためには、True Depth カメラを搭載した iPhone あるいは iPad が必要である。ARKit とインカメラを用いた顔認識では、顔の向き、鼻や口の位置、口や目の開き具合などの情報を、最大で毎秒 60 フレーム取得することができる。この速度から、ARKit による視線追跡は、視線入力によるインタフェースのユーザビリティテストを実行するには十分であるとされている [GVA20]。実際に、ARKit による頭部および視線追跡機能を用いたインタラクション手法が研究されており、HeadReach [VHCR20] では、ARKit を用いて頭部の向きを認識することによる、スマートフォンに対する新たな選択手法が提案された。また、GazeConduits [VHH⁺20] では、机上に設置した複数台のタブレットデバイスに対する、ARKit を利用したマルチデバイスインタラクションが提案された。

3.2 視線推定アルゴリズム

本研究では，ARKit を用いてスマートフォンのスクリーン平面内における視点座標を算出することにより視線推定を行う．スクリーン平面における視点座標は，スマートフォンのスクリーン平面および視線ベクトルの交点と一致する．その交点を算出するには，拡張現実空間におけるデバイスの位置と向き，およびユーザの目の位置と向きが必要である．ARKit は，カメラ位置を原点とした拡張現実空間を構築し，デバイスが撮影する映像の変化および加速度センサの値を用いることによってデバイスの位置および角度を更新する [Inc20c]．この更新の際に，カメラ映像内に映り込んだ平面，物体，あるいは人物には ARAnchor¹が設置され，撮影範囲に存在する限り追跡が可能となる．特に，人の顔には ARFaceAnchor²が設置され（図 3.1），拡張現実空間における顔の位置および向き，または目や口などの位置，角度および開閉などの状態を取得することが可能となる．目については特に多様な情報を取得可能であり，瞼の開閉具合，および眼球の回転角などを，左右の目それぞれについて取得することができる．しかし，目の位置や向きなどは，ARFaceAnchor に紐づけられた座標および回転角によって表現される（図 3.2）ため，目の位置および向きを利用するためには，拡張現実空間における座標に変換する必要がある．この変換によって，拡張現実空間における視線ベクトルの始点および向きを算出する．

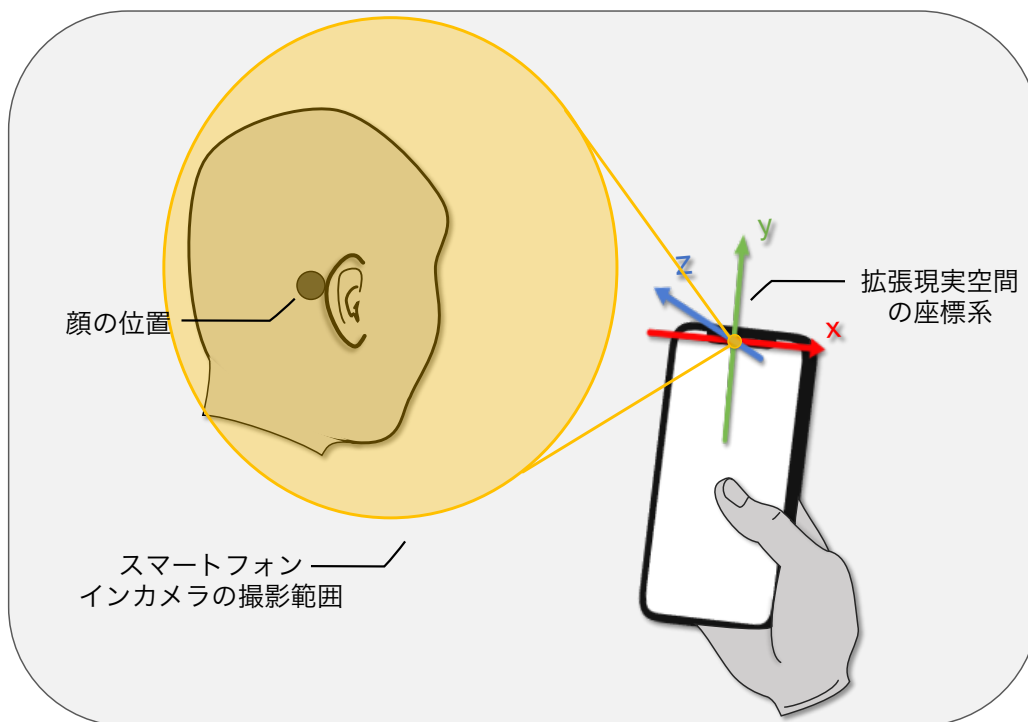


図 3.1 ARKit により構築された拡張現実空間におけるデバイスの位置および顔の位置．

¹<https://developer.apple.com/documentation/arkit/anchor>

²<https://developer.apple.com/documentation/arkit/faceanchor>

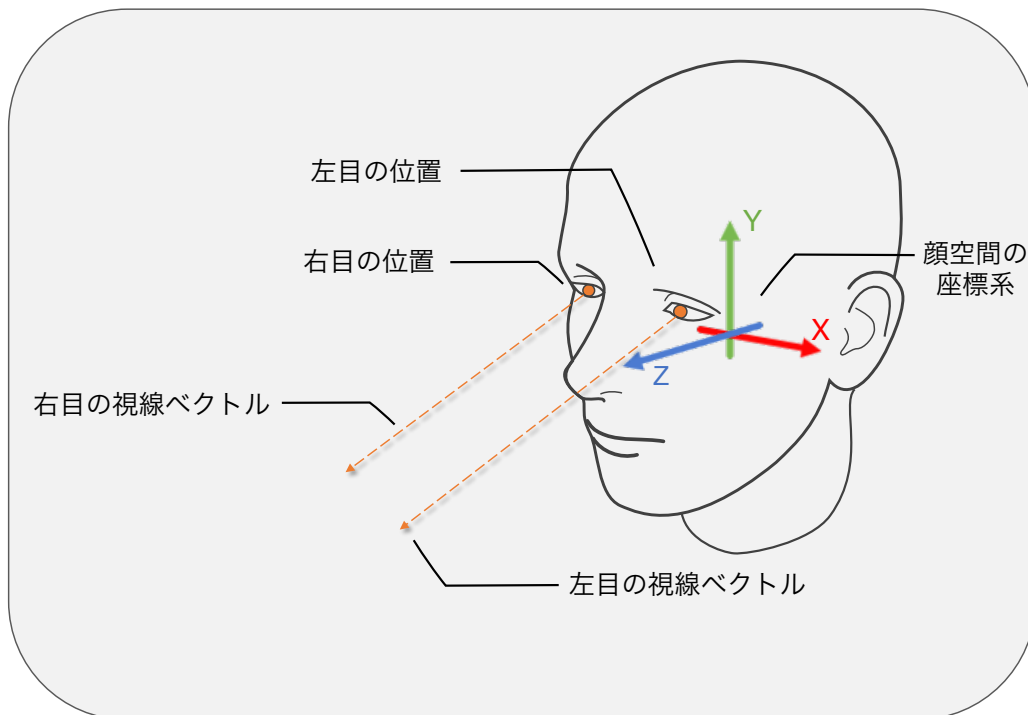


図 3.2 顔空間における両目の位置および視線ベクトル。

このようにして，拡張現実空間におけるスマートフォンのスクリーン平面の位置と向き，ユーザの右目の視線ベクトルの始点と向き，および，ユーザの左目の視線ベクトルの始点と向きが算出される．多くの場合，右目の視線ベクトルとスクリーン平面の交点，および左目の視線ベクトルとスクリーン平面の交点にはわずかに差異が生じる．そのため本アルゴリズムでは，左右の視線ベクトルとスクリーン平面の交点を算出したのち，その中間にある点をスクリーン平面におけるユーザの視点座標とした．図 3.3 に，視点算出方法のイメージ図を示す．なお視点座標は，最後に，拡張現実空間における座標から，スクリーン平面における座標（図 3.4）に変換する．スクリーン平面における座標系は，画面左上を原点とし，X 軸正の向きは水平方向右であり，Y 軸正の向きは垂直方向下である．この座標系において各点は，画面を描画するための座標を示す抽象的な単位であるポイント（以降，pt）によって表現される．また，本研究において実装に用いた iPhone X の画面サイズは，横幅が 375 pt，縦の長さが 812 pt であり，1 pt はおよそ 0.17 mm である．

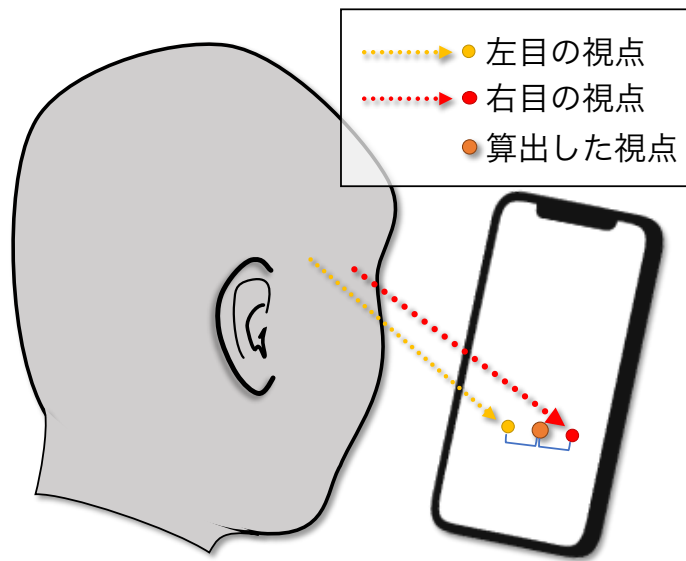


図 3.3 実装した視線推定アルゴリズムによる視点の算出.

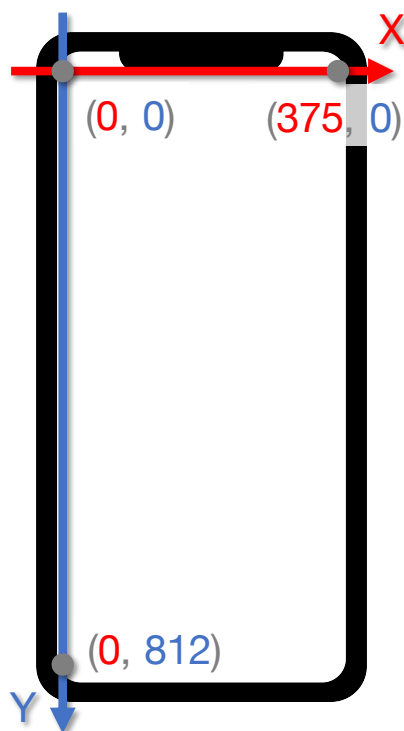


図 3.4 スクリーン平面の座標系および iPhone X における代表的な点の座標 (単位は pt).

3.3 予備実験 1：実装した視線推定アルゴリズムの精度

3.2 節にて説明した視線推定アルゴリズムを用いて、ARKit を利用した視線推定の精度を評価するための予備実験を行った。

3.3.1 参加者および実験環境

本実験の参加者は、大学生および大学院生の 8 名 (P1–P8, 19–25 歳, 平均年齢 22.3 歳) であった。内訳として、男性が 6 名, 女性が 2 名, 3 名が左利きであり, 5 名が右利きであった。また, 視力補正については眼鏡着用者, コンタクトレンズ着用者および裸眼の参加者がそれぞれ 4 名ずつであった。

全ての参加者は屋内の蛍光灯下にて実験に参加した。参加者は着席した状態にて, 実験用のアプリケーションがインストールされたスマートフォンを把持した。参加者は, スマートフォンのインカメラに顔が写る限り, スマートフォンの持ち方および姿勢を自由に変更して良いものとした。

3.3.2 タスク

本実験におけるタスクは, 画面内に指示された点を凝視することである。参加者が画面内に表示されたボタンをタップすることによってデータの収集が開始される。参加者が約 1 秒間見つめる間に, 実験用アプリケーションは連続した 50 フレームの視線情報を収集する。図 3.5 (a) に実験中のアプリケーションの画面を示す。凝視する点は中央部の円および, その中心にある + によって指示される。中央部の円は半径が 25 pt である。下部の四角はデータ収集を開始するためのボタンであり, 上部のプログレスバーは実験の進行度合いを示している。

指示される点は, 画面内の 9 点 (= 3 (上部, 中央, 下部) × 3 (左側, 中央, 右側)) であり (図 3.5 (b)), ランダムな順にて 3 度ずつ指示される。参加者は始めに眼鏡あるいはコンタクトレンズを着用した状態にてタスクを実施し, 続いて休憩を挟んだのちに眼鏡あるいはコンタクトレンズを取り外した状態にてタスクを実施した。なお, 実験中は常に実験用アプリケーションが視線情報を記録しており, 顔がインカメラの撮影範囲外に出てしまった場合には, データの収集を中断して, 顔の位置を撮影範囲内に戻すように指示した。

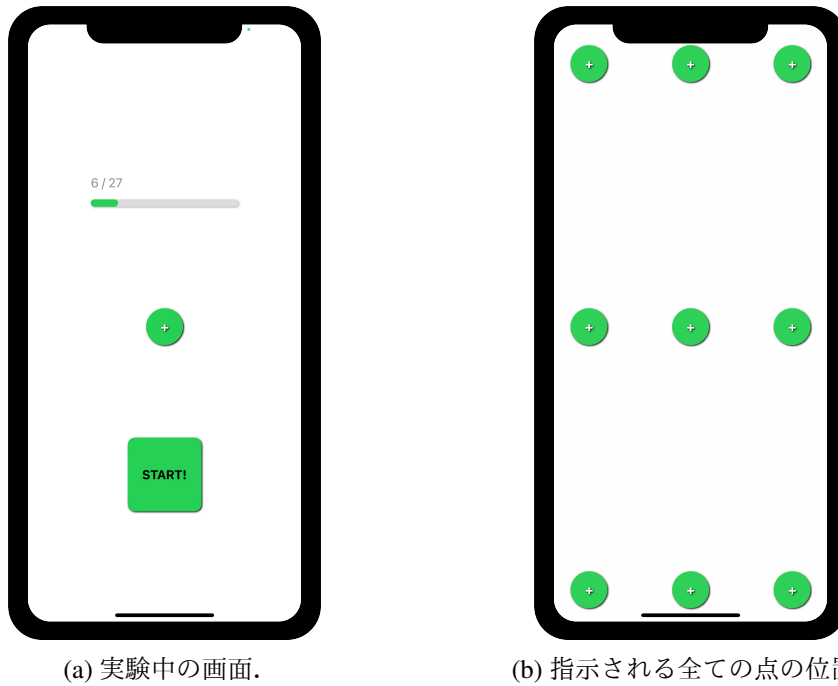


図 3.5 予備実験 1 に用いたアプリケーション画面のスクリーンショット。

3.3.3 結果と考察

本実験を通して 21,600 フレーム (= 50 フレーム × 9 点 × 3 回 × 8 人 × 2 条件) の視線情報を記録した。結果として、推定した視点とターゲット間に平均で 492.4 pt の距離があることが分かった。表 3.1 に観測された視点とターゲットの誤差距離、および軸方向の誤差距離の平均値、最小値、中央値、および最大値を示す。

シャピロ・ウィルク検定によると、誤差距離および軸方向の誤差距離の分布はいずれも正規分布に従わない ($p < 0.05$) ことが分かった。x 軸方向の誤差距離の中央値が 67.2 pt、y 軸方向の誤差距離の中央値が 380.3 pt であることから、y 軸方向の推定精度は特に悪いことが明らかである。

表 3.1 観測された視点とターゲットの誤差距離、および軸方向の誤差距離 (単位は pt)。

	平均値	最小値	中央値	最大値
誤差距離	492.4	2.1	392.2	1523.1
x 軸方向	85.9	0.0	67.2	400.0
y 軸方向	468.7	0.0	380.3	1522.5

続いて、表 3.2 に、視力補正条件毎に、観測された視点とターゲットの誤差距離、および軸方向の誤差距離の平均値、最小値、中央値、および最大値を示す。シャピロ・ウィルク検

定によると、いずれの分布も正規分布に従わない ($p < 0.05$) ことが分かった。Wilcoxon の符号順位検定によると、コンタクト条件とコンタクト着用者の裸眼条件では、誤差距離および両軸方向の誤差距離について中央値に有意差がある ($p < 0.01$) ことが分かった。同様に、Wilcoxon の符号順位検定によると、眼鏡条件と眼鏡着用者の裸眼条件では、誤差距離および両軸方向の誤差距離について中央値に有意差がある ($p < 0.01$) ことが分かった。また、コンタクト条件および眼鏡条件について Mann-Whitney の U 検定を行ったところ、誤差距離および両軸方向の誤差距離について中央値に有意差がある ($p < 0.01$) ことが分かった。これはすなわち視力の補正方法によって、実装した視線推定の精度に影響が生じている可能性があることを示唆している。特に眼鏡条件の y 軸方向については、誤差距離の中央値が 683.3 pt であり、全ての視力補正条件および全ての方向の中で最も悪い精度である。これは、眼鏡のレンズによって目の位置認識に誤差が発生したためであると考えられる。

表 3.2 視力補正および軸の方向毎に観測された視点とターゲットの差異 (単位は pt)。

視力補正		平均値	最小値	中央値	最大値
コンタクト	誤差距離	403.6	15.2	444.6	930.6
	x 軸方向	114.4	0.0	84.5	400.0
	y 軸方向	371.2	0.4	426.8	927.0
裸眼 (コンタクト着用者)	誤差距離	323.3	14.8	326.3	678.7
	x 軸方向	71.4	0.0	60.3	258.2
	y 軸方向	304.6	0.2	312.3	669.0
眼鏡	誤差距離	739.0	7.7	686.9	1523.1
	x 軸方向	72.4	0.0	53.6	366.7
	y 軸方向	719.4	0.2	683.3	1522.5
裸眼 (眼鏡着用者)	誤差距離	503.8	4.1	426.8	1412.1
	x 軸方向	85.3	0.1	70.0	397.0
	y 軸方向	479.5	0.0	394.8	1389.9

ターゲットと推定された視点の差異は大きいですが、本研究にて提案する *FlickSight* は、視線移動の開始位置を指定しないため、視線の移動を取得することができれば良い。すなわち、ユーザが任意の点を凝視している際に、視線推定システムは凝視していることが分かれば十分であり、どの点を凝視しているかを判別する必要はない。そこで、各点が指示された際に収集した 50 フレーム分の視線情報の分散について調査した。表 3.3 に、各点が指示された際に収集した 50 フレーム分の視線情報の分散の統計量を示す。x 軸方向の分散は平均で 15.63 pt、y 軸方向の分散は平均で 29.77 pt であることが分かった。本実験にて凝視する点の半径が 25 pt だったことから、凝視していることを検出できると考えられる。

表 3.3 各点が指示された際に収集した 50 フレーム分の視線情報の分散の統計量 (単位は pt).

方向	平均値	最小値	中央値	最大値
x	15.63	1.62	10.26	99.29
y	29.77	2.82	15.01	380.92

第4章 *FlickSight* の設計

本論文にて提案する *FlickSight* はフリックと視線移動を組み合わせることにより、スマートフォンの操作を拡張するための入力手法である。ユーザはフリックと視線移動を同時に行うことによって、*FlickSight* に割り当てられた操作が可能となる。本章ではまず、*FlickSight* の設計方針および種類について説明する。続いて、*FlickSight* が通常のフリックと競合しないように設計するために行なった予備実験2について述べ、最後に *FlickSight* が採用する視線移動の距離および移動方向を設定するために行なった予備実験3について述べる。

4.1 *FlickSight*

FlickSight はフリックと視線移動を同時に実行する入力手法である。フリックとは、スマートフォンのタッチスクリーンへタッチダウンした後、任意の方向に移動した後にタッチアップするタッチジェスチャであり、スクロールやパンを高速に実行する際に用いられる [Inc20b]。また、コンピュータ操作に用いる視線移動は、直線的で単純なジェスチャ（例：[MLGH10]）や直線的あるいは曲線的な視線移動を組み合わせた複雑なジェスチャが提案されている（例：[DHI17]）。本研究では、ユーザにとって記憶しやすく、かつフリックの高速性を維持するために、直線的な視線移動を採用する。すなわち、*FlickSight* の持つ入力語彙は、フリックの方向数 × 視線移動の方向数となる。*FlickSight* の設計方針として、以下の2通りが考えられる。

1. フリックの方向と視線移動の方向を任意の数に設定（図4.1）
2. フリックの方向を基準とした視線移動の方向を利用（図4.2）

また *FlickSight* は、いずれの設計においても、視線移動のないフリックを単なるフリックとして利用することが可能である。

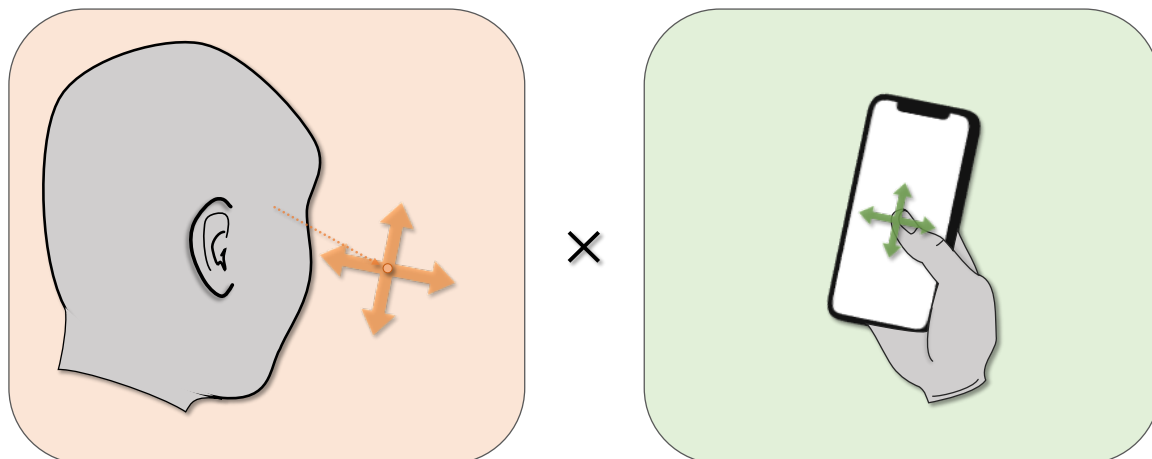


図 4.1 フリックの方向と視線移動の方向を任意の数に設定した入力手法。
4 方向（上下左右）への視線移動と 4 方向（上下左右）へのフリックを組み合わせる例。

4.1.1 フリックの方向と視線移動の方向を任意の数に設定

フリックの方向と視線移動の方向を任意の数に設定する *FlickSight* は、分割数によってその入力語彙が異なる。例えば、フリックの方向を 4 方向（上下左右）、視線移動の方向を 4 方向（上下左右）に設定することにより（図 4.1）、*FlickSight* は 4 （上下左右へのフリック） \times 5 （上下左右への視線移動 + 視線移動なし） $= 20$ 種類の入力語彙を用意することができる。これらの *FlickSight* は、*Flick*-(フリックの方向)-*Sight*-(視線移動の方向)の規則に従って命名する。例えば、上方向のフリックと上方向の視線移動を組み合わせた *FlickSight* は、*Flick-Up-Sight-Up* と呼び、右方向のフリックと左方向の視線移動を組み合わせた *FlickSight* は、*Flick-Right-Sight-Left* と呼ぶ。これらの名前一覧を表 4.1 に示す。

表 4.1 フリックの方向と視線移動の方向を上下左右の 4 方向に設定した *FlickSight* の名前一覧.

		フリック	
		上	下
視線移動	上	<i>Flick-Up-Sight-Up</i>	<i>Flick-Down-Sight-Up</i>
	下	<i>Flick-Up-Sight-Down</i>	<i>Flick-Down-Sight-Down</i>
	左	<i>Flick-Up-Sight-Left</i>	<i>Flick-Down-Sight-Left</i>
	右	<i>Flick-Up-Sight-Right</i>	<i>Flick-Down-Sight-Right</i>
	なし	<i>Flick-Up</i>	<i>Flick-Down</i>
		フリック	
		左	右
視線移動	上	<i>Flick-Left-Sight-Up</i>	<i>Flick-Right-Sight-Up</i>
	下	<i>Flick-Left-Sight-Down</i>	<i>Flick-Right-Sight-Down</i>
	左	<i>Flick-Left-Sight-Left</i>	<i>Flick-Right-Sight-Left</i>
	右	<i>Flick-Left-Sight-Right</i>	<i>Flick-Right-Sight-Right</i>
	なし	<i>Flick-Left</i>	<i>Flick-Right</i>

この方針にて設計された *FlickSight* はその豊富な入力語彙を活かして、システムショートカットとして利用することや、操作の多いアプリケーションなどにおけるメニュー選択に割り当てることができると考えられる。また、これらの *FlickSight* をフリックキーボードに適用することによって、キーの数を削減し、表示領域を小さくすることができる可能性がある。

上記の組み合わせに対して、視線移動の方向を 8 方向（上下左右+斜め）に設定することにより、入力語彙を 36 種類に拡張することも可能である。ただし方向数を増やすことは、ユーザにとって操作が困難になる可能性がある。例えばフリックのみであったとしても、32 方向へのフリックを意図的に使い分けることは容易ではないだろう。その細かな方向の使い分けを、フリックと視線移動の両方に求めることは、ユーザにとって負担となり得る。

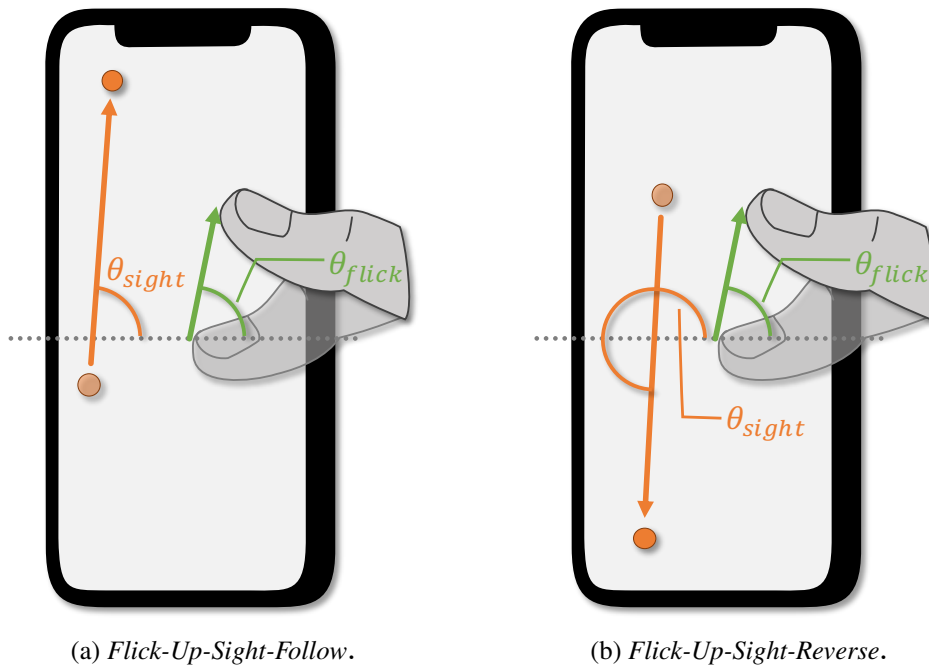


図 4.2 *FlickSight-Follow* および, *FlickSight-Reverse* のうち, 上方向へのフリックを基準とする入力の例.

4.1.2 フリックの方向を基準とした視線移動の方向を利用

フリックの方向と視線移動の方向を任意の数に設定する方針は, ユーザが指と視線を異なる方向に動かすことから混乱の原因となる恐れがある. そこで, 指と視線を動かす方向に関連を持たせるために設計した, フリックの方向を基準とした視線移動の方向を利用する *FlickSight* を説明する. フリックの方向を基準とした視線移動の方向を利用する *FlickSight* は二種類あり, それらを *FlickSight-Follow*, および *FlickSight-Reverse* と名付けた. *FlickSight-Follow* は, フリックと, フリックと同じ方向への視線移動を組み合わせた *FlickSight* である. 図 4.2 に示す θ_{flick} および θ_{sight} が, $|\theta_{\text{sight}} - \theta_{\text{flick}}| < 90^\circ$ を満たすものがこれに当てはまる. *FlickSight-Follow* は, 2本の指を同じ方向にフリックするジェスチャと類似した入力手法と見なすことができる. しかしながら, スマートフォンには2本の指を同じ方向にフリックすることによる操作が割り当てられていることは稀である. そのため *FlickSight-Follow* の利用例としては, ラップトップ PC に搭載されているタッチパッドにて実行される3本指によるフリックと同様に, 起動中のアプリケーション一覧を表示するなどの操作を割り当てることが考えられる.

FlickSight-Reverse は, フリックと, フリックと逆方向への視線移動を組み合わせた *FlickSight* である. 図 4.2 に示す θ_{flick} および θ_{sight} が, $|\theta_{\text{sight}} - \theta_{\text{flick}}| > 90^\circ$ を満たすものがこれに当てはまる. *FlickSight-Reverse* は, 2本の指をつまむように動かす, ピンチイン/アウトに似た入力手法と見なすことができ, 画面に表示されている内容の拡大縮小などの操作を割り当てる

ことが考えられる。

これらの視線移動によって、*FlickSight* は一方向のフリックに対して、順方向、逆方向と2種類の入力語彙を拡張することができる。これらの *FlickSight* は、通常の（視線移動のない）フリックと競合しないことが想定される。すなわち、一般的に用いられる上下左右方向へのフリックに対して適用することにより、4（上下左右へのフリック）×3（順方向、逆方向への視線移動+視線移動なし）= 12 種類へと入力語彙を拡張することができる。表 4.2 に、フリックの方向を基準とした視線移動の方向を利用する入力手法の名前と方向の例を示す。

表 4.2 フリックの方向を基準とした視線移動の方向を利用する入力手法の方向と名前の一覧。

名前	フリックの方向	視線移動の方向
<i>Flick-Up-Sight-Follow</i>		上
<i>Flick-Up</i>	上	移動なし
<i>Flick-Up-Sight-Reverse</i>		下
<i>Flick-Right-Sight-Follow</i>		右
<i>Flick-Right</i>	右	移動なし
<i>Flick-Right-Sight-Reverse</i>		左
<i>Flick-Down-Sight-Follow</i>		下
<i>Flick-Down</i>	下	移動なし
<i>Flick-Down-Sight-Reverse</i>		上
<i>Flick-Left-Sight-Follow</i>		左
<i>Flick-Left</i>	左	移動なし
<i>Flick-Left-Sight-Reverse</i>		右

4.2 予備実験 2：フリック中に発生する視線移動の観察

FlickSight が通常のフリック操作と共存するためには、日常的なフリック操作中に発生しづらい視線移動を採用する必要がある。そのため、ユーザがフリックやスワイプを行う際の視線移動の観察を目的とする実験を行った。

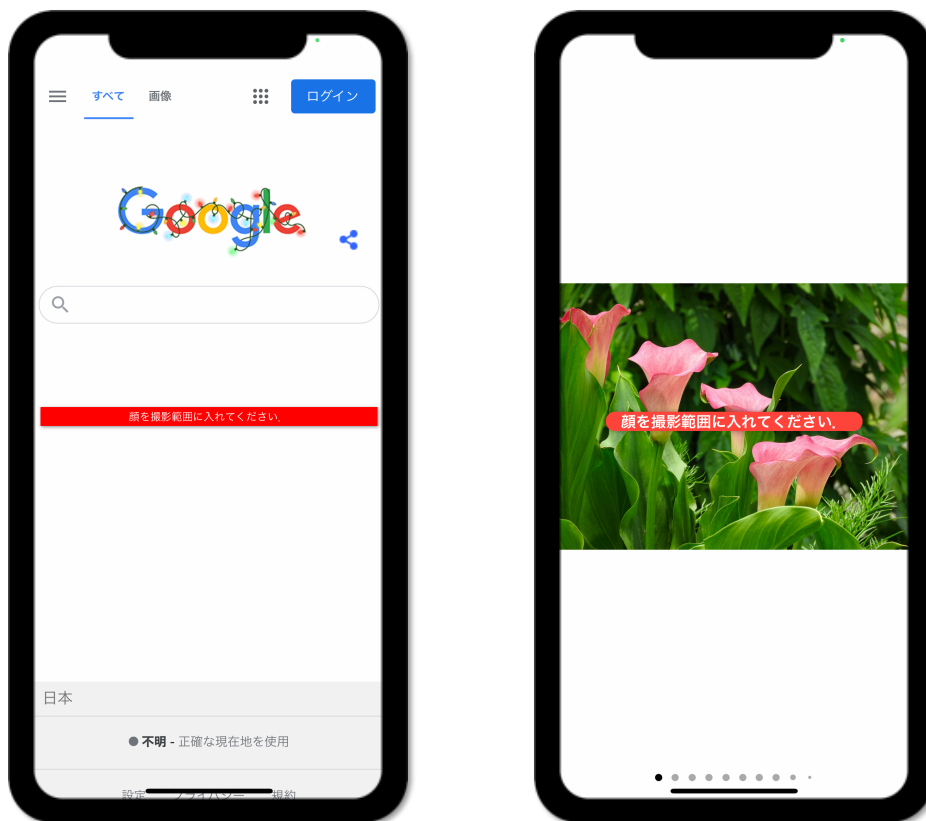
4.2.1 参加者および実験環境

本予備実験の参加者は、著者を含む大学生および大学院生の 12 名（P1–P12, 19–25 歳，平均年齢 21.8 歳）であった。男性が 9 名，女性が 3 名，3 名が左利きであり，9 名が右利きであった。また，視力補正については眼鏡着用者，コンタクトレンズ着用者および裸眼の参加者がそれぞれ 4 名ずつであった。

全ての参加者は屋内の蛍光灯下にて実験に参加した。参加者は着席した状態にて、実験用のアプリケーションがインストールされたスマートフォンを把持した。参加者は、スマートフォンのインカメラに顔が写る限り、スマートフォンの持ち方および姿勢を自由に変更して良いものとした。

4.2.2 タスクと実験手順

本予備実験のタスクは指示されたアプリケーションを3分間自由に操作することである。一つのアプリケーションを操作するタスクを完了したのちに、十分な休憩を挟み、次のアプリケーションを操作するタスクを開始する。本予備実験の対象となるアプリケーションは、ウェブブラウザおよびフォトビューアである。ウェブブラウザは主に縦方向、フォトビューアは主に横方向のフリックによって操作され、本予備実験の目的に適していると考えられる。図4.3に実験に用いたアプリケーション画面のスクリーンショットを示す。中央に表示されているのは、顔がスマートフォンに搭載されたインカメラの撮影範囲内がないという警告である。



(a) ウェブブラウザアプリケーションの画面。 (b) フォトビューアアプリケーションの画面。

図 4.3 予備実験 2 に用いたアプリケーション画面のスクリーンショット。

4.2.3 結果

本予備実験により合計で2,629回のフリック、および154,715フレームの視線データを記録した。ウェブブラウザを対象とした実験では、合計で1,309回のフリック、および67,960フレームの視線データを記録した。またフォトビューアを対象とした実験では、合計で1,320回のフリック、および86,755フレームの視線データを記録した。

フリック

実験によって記録した2,629回のフリックを分析した。表4.3に、操作したアプリケーション毎のフリック持続時間の統計量を示す。操作したアプリケーション毎のフリックに要する時間は、シャピロ・ウィルク検定に基づき、正規分布に従わない ($p < 0.01$) ことが分かった。また、Brunner-Munzel検定により、ウェブブラウザを使用する際のフリックに要する時間の中央値と、フォトビューアを使用する際のフリックに要する時間の中央値に有意な差がある ($p < 0.01$) ことが明らかになった。これは、ウェブブラウザを操作する際のスクロールに用いられるパンジェスチャの影響であると考えられる。そこで、フォトビューアにて計測されたフリックの98.9%に当たる1,306回のフリックが680ms以下であったことから、以降では要する時間が680ms以下のフリックについて議論する。なお、要する時間が680ms以下のフリックは、ウェブブラウザにて計測されたフリックの89.9%に当たる1,177回のフリックを含み、両アプリケーション合計で2,483回のフリックを含む。

表4.3 アプリケーション毎のフリック持続時間の統計量 (単位はms)。

操作対象	平均値	最小値	中央値	最大値
ウェブブラウザ	284	3	135	3970
フォトビューア	93	12	66	2610

視線移動

フリック中に記録された視線情報を用いて、フリック中の視線移動について分析した。合計2,483回のフリックのうち、2,446回のフリックは少なくとも2フレームの視線情報収集に成功していた。これらのフリック中に観察された視線情報について、フリック開始時の視点を視線移動の開始位置、フリック終了時の視点を視線移動の終了位置と定義することによって、視線移動の直線距離、および総移動距離を求めた。表4.4に、フリック中に観測された視線移動の直線距離、総移動距離の平均値、最小値、中央値、および最大値を示す。

表 4.4 フリック中に観測された視線移動距離に関する統計量.

	平均値	最小値	中央値	最大値
直線距離 (pt)	82.0	0.6	45.9	2457.6
総移動距離 (pt)	134.9	0.9	72.4	2457.6

フリック中に観察された視線移動の直線距離を、本予備実験に用いた iPhone X の大きさ（横幅 × 長さ = 375 pt × 812 pt）と比較した。図 4.4 に、フリック中に観察された視線移動の直線距離の対数分布を示す。なお、図中の赤線は iPhone X の横幅（375 pt）を示しており、黄線は iPhone X の長さ（812 pt）を示している。総移動距離を比較した場合、全体の 92.0% に当たる 2,255 回の視線移動は 375 pt よりも短く、全体の 98.4% に当たる 2,409 回の視線移動は 812 pt よりも短いことが分かった。また直線距離を比較した場合、全体の 97.0% に当たる 2,379 回の視線移動は 375 pt よりも短く、全体の 99.7% に当たる 2,442 回の視線移動は 812 pt よりも短いことが分かった。つまり、フリック中に視線移動が発生したと認識するための距離を画面の横幅や長さよりも長くすることによって、通常のフリックとの競合を回避することができると思われる。

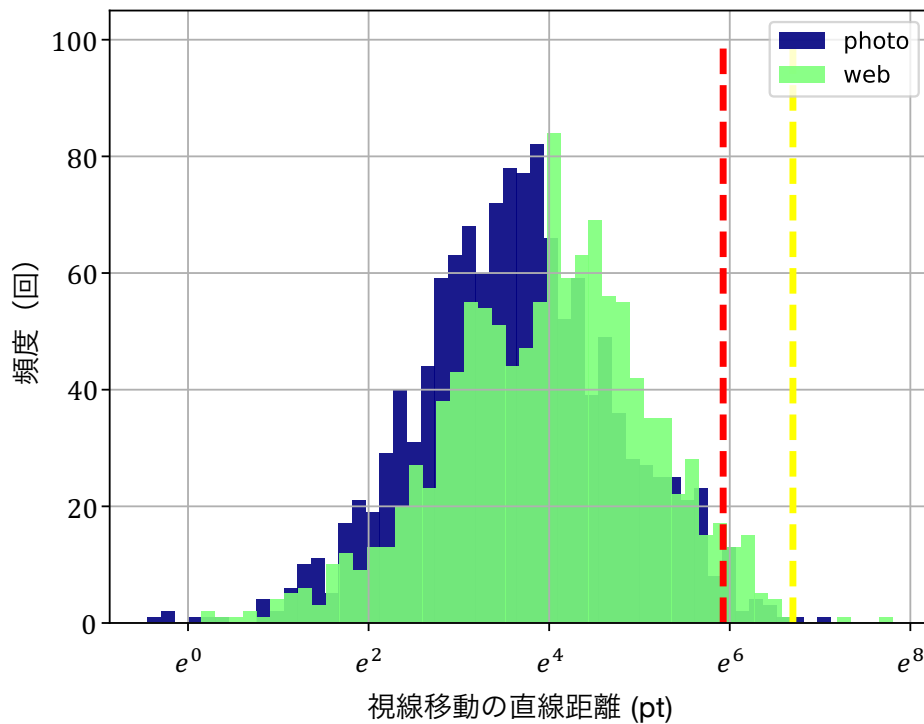


図 4.4 予備実験 2 にてフリック中に観察された視線移動の直線距離の対数分布。赤線は iPhone X の横幅（375 pt）を示しており、黄線は iPhone X の長さ（812 pt）を示している。

4.3 予備実験3：意図した直線的な視線移動の観察

予備実験2の結果から、フリック中にiPhone Xの横幅や長さを超えるような距離の長い視線移動が発生することは稀であることが分かった。つまり、フリック中に視線移動が発生したと認識するための距離を長くすることによって、通常のフリックとの競合を回避することができると考えられる。しかしながら、*FlickSight*に採用する視線移動に必要な距離が長くなれば長くなるほど、ユーザの目に負担がかかり、実行の高速さや容易さといった直線的な視線移動の持つ利点が薄れてしまう可能性がある。そのため、指示された際のユーザの直線的な視線移動を観察する実験を行い、*FlickSight*に採用する視線移動として適切な移動量、および移動方向を調査した。

4.3.1 参加者および実験環境

本予備実験の参加者は、予備実験2と同様である。

全ての参加者は屋内の蛍光灯下にて実験に参加した。参加者は着席した状態にて、実験用のアプリケーションがインストールされたスマートフォンを把持した。参加者は、スマートフォンのインカメラに顔が写る限り、スマートフォンの持ち方および姿勢を自由に変更して良いものとした。

4.3.2 タスクと実験手順

本予備実験におけるタスクは、画面内に表示されたStartボタンをタップし、3秒のカウントダウンののち、矢印によって指示された方向に視線を動かすことである。1セッションの間に8方向(図4.5)がランダムな順にて1度ずつ表示される。図4.6は、予備実験3に用いたアプリケーション画面のスクリーンショットである。視線移動の方向は画面中央に示されており、図4.6は下方向への視線移動の指示である。画面上部のプログレスバーは実験の進行度合いを示している。カウントダウンは音によって指示され、参加者が視線移動をするタイミング、すなわちカウントダウン終了時には、カウントダウンと異なる音が鳴る。全参加者は始めに、タイミングを合わせることを、および視線移動の方向を確認することを目的として、訓練セッションに取り組んだ。その後、休憩を挟んで5セッションを実施した。

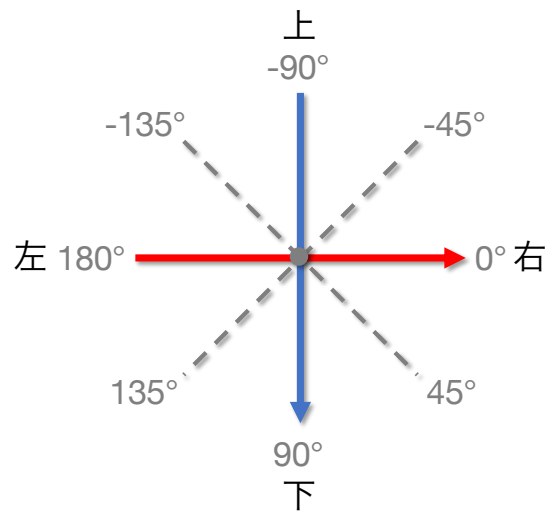


図 4.5 予備実験 3 にて指示される視線移動の方向と角度の対応.

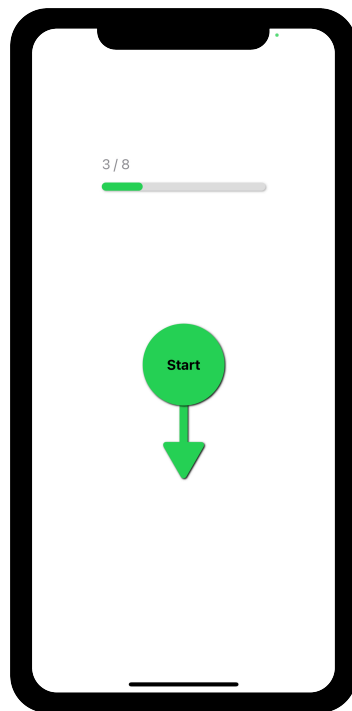


図 4.6 予備実験 3 に用いたアプリケーション画面のスクリーンショット.

4.3.3 視線移動の検出

本予備実験により，合計で 167,774 フレームの視線情報を収集した．本予備実験の目的は，直線的な視線移動による操作時における視線移動の距離および，視線移動の方向の正確性について調査することである．そのため，収集した視線情報から視線移動を検出する必要がある．

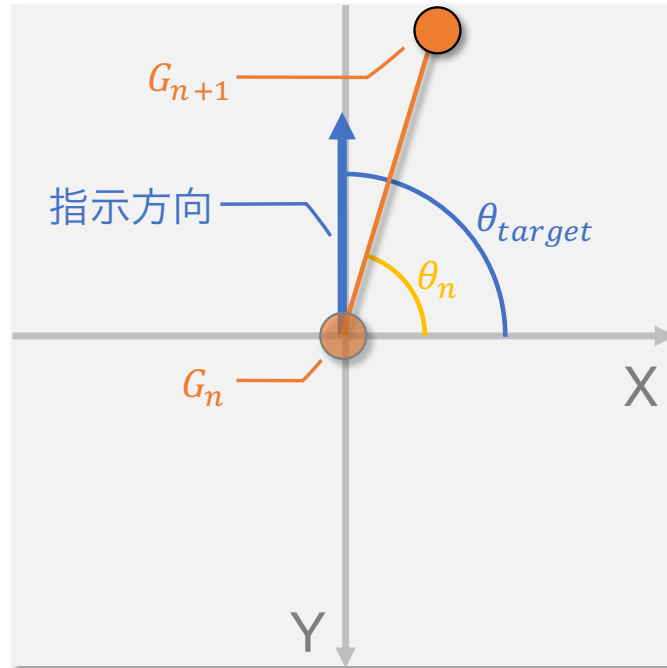


図 4.7 予備実験 3 における視線移動検出の概略図．橙色の点が視点を示しており，青色の矢印が指示された視線移動の方向である． $|\theta_{\text{target}} - \theta_n| \leq 45^\circ$ を満たすときに視線移動が検出される．

以下では，収集した視線情報から視線移動を検出する方法について説明する．収集した視線情報を，参加者，セッションおよび指示された視線移動の方向毎に分割する．分割した視線情報を時系列順に並べ，任意のフレームにて記録された視線位置 G_n に対する次のフレームにて記録された視線位置 G_{n+1} の角度 θ_n (図 4.7) を算出する ($n \in \mathbb{N}, 0 \leq n$)．算出した角度を時系列順に確認し，指示された方向 θ_{target} との差異が $\pm 45^\circ$ 以内である場合に視線移動が開始したことを検出する．視線移動が検出されると，以下の条件のいずれかを満たすまでの一連の視線情報を一つの視線移動として認識する．視線移動の終了を検出するための条件は以下の 2 つである．

1. $|\theta_k - \theta_{\text{target}}| > 90^\circ$ を満たす $G_k (k \in \mathbb{N}, n < k)$ が確認される (図 4.8(a)).
2. $l (l \in \mathbb{N})$ フレームの視線を含む視線移動について， $|\theta_m - \theta_{\text{target}}| < 45^\circ$ を満たす $m (m \in \mathbb{N}, 0 \leq m \leq l)$ が $l/2$ 個未満になる (図 4.8(b)).

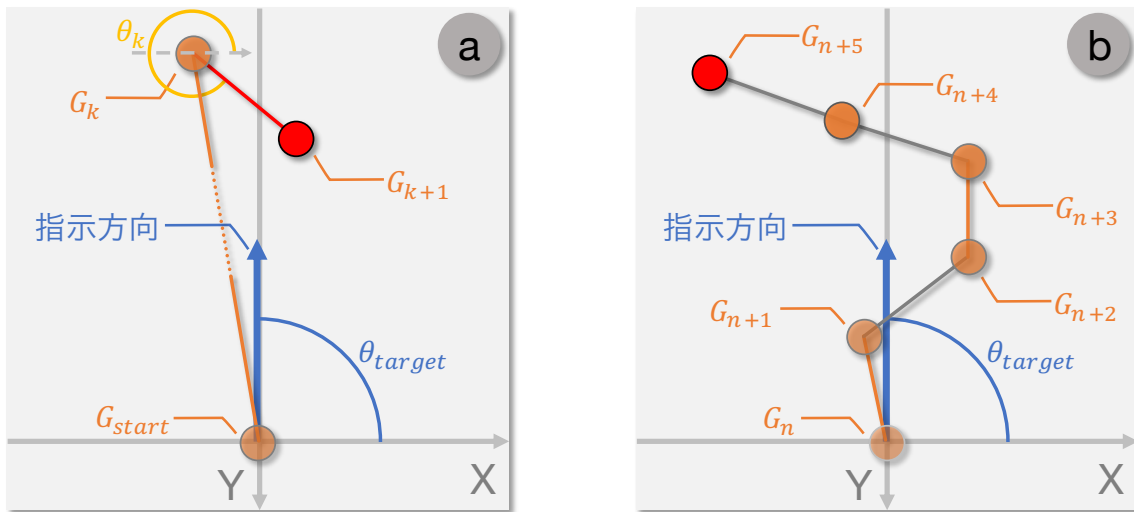


図 4.8 予備実験 3 における視線移動終了の条件. 円は視点を示しており, 青色の矢印が指示された視線移動の方向である. 円同士を結ぶ直線は, 橙色が θ_{target} との差異が $\pm 45^\circ$ 以内の視線移動を示し, 赤色は θ_{target} との差異が $\pm 90^\circ$ より大きい視線移動を, 灰色はそれ以外の視線移動を示す. (a) $|\theta_k - \theta_{target}| > 90^\circ$ を満たす $G_k (k \in \mathbb{N}, n < k)$ が確認された場合. (b) 6 フレームの視線情報を含む視線移動について, $|\theta_m - \theta_{target}| < 45^\circ$ を満たす $m (m \in \mathbb{N}, 0 \leq m \leq 6)$ が 3 個未満になった場合.

上記に基づき視線移動の検出を行うことによって, 参加者が行った視線移動の候補となる視線移動を取得した. これらの候補のうち, 視線移動開始の時間がカウントダウン終了の時間ともっとも近いものを参加者が行った視線移動とした. 結果として, 480 回 (12 人 \times 5 セッション \times 8 方向) の視線移動を検出した.

4.3.4 視線移動の分析および議論

検出した視線移動に対して, 直線距離, および角度の誤差についての分析を行う. なお視線移動の直線距離とは, 視線移動の開始点と終了点を直線で結んだ際の距離を意味し, 視線移動の角度とは, 視線移動の開始点に対する終了点の方向を意味する. 表 4.5 に, 検出した視線移動の直線距離, および角度の誤差に関する統計量を示す. なおシャピロ・ウィルク検定に基づくと, 直線距離の分布は正規性に従わず ($p < 0.01$), 角度の誤差の分布は正規性に従う ($p > 0.01$) ことが分かった. 視線移動の距離, および角度に対して, セッション (Session), 指示した視線移動の方向 (Direction) が与える影響について調査した.

表 4.5 予備実験 3 にて収集した視線情報から検出した視線移動に関する統計量.

	平均値	最小値	中央値	最大値
直線距離 (pt)	448.5	0.3	85.1	4630.2
角度の誤差 (度)	-1.7	-67.6	-3.2	76.0

直線距離

視線移動の直線距離とは、視線移動の開始点および終了点の座標を直線で結んだ際の距離を示す。これは、*FlickSight* における視線移動の有無を決定する際に必要である。Tukey-Kramer 法による多重比較の結果、いずれの *Session* 間、およびいずれの *Direction* 間にも有意差がないことが分かった。

FlickSight における視線移動の有無を決定するための閾値を設定する。そのために、4.2 節にて記録された視線移動の直線距離と本予備実験にて記録された視線移動の距離を比較する。図 4.9 に、予備実験 2 にて記録したフリック中の視線移動の識別率、および予備実験 3 にて検出した視線移動の識別率を示す。青色のグラフは、予備実験 2 にて記録したフリック中の視線移動を視線移動ではないと識別する確率を示しており、赤色のグラフは予備実験 3 にて検出した視線移動を視線移動であると識別する確率を示す。また、緑色のグラフはそれらの合計値を示す。このグラフにて、閾値が 218 pt の時にこれらの確率の和の最大値を得た。

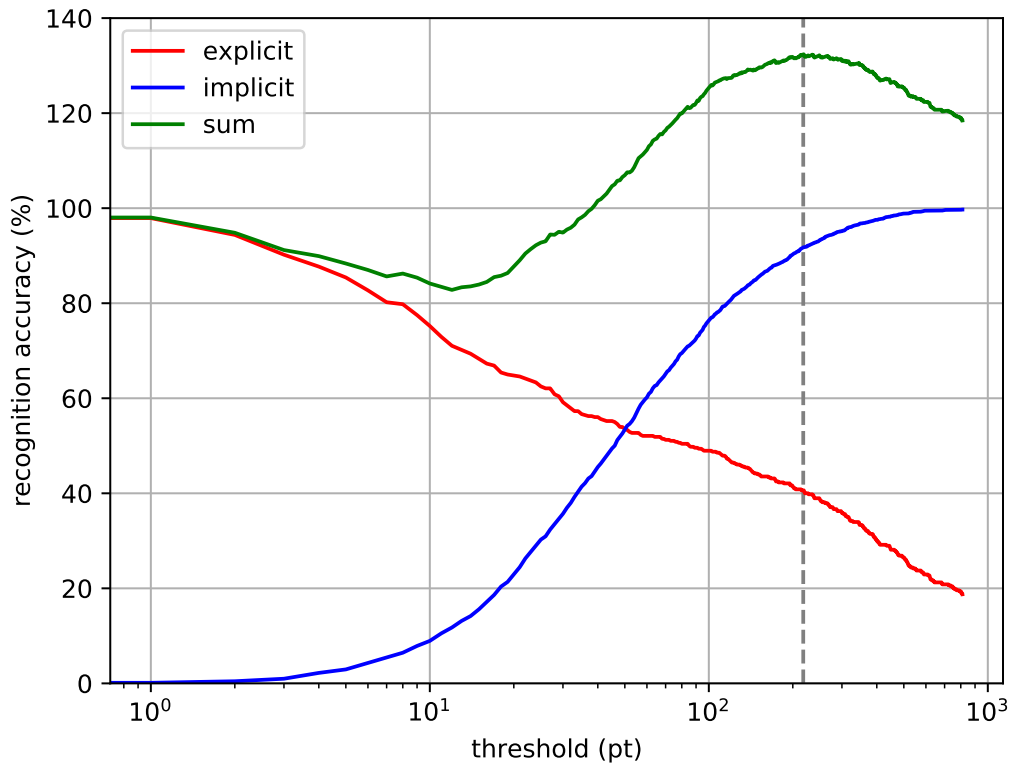


図 4.9 予備実験 2 にて記録したフリック中の視線移動を視線移動ではないと識別する確率 (青), 予備実験 3 にて検出した視線移動を視線移動であると識別する確率 (赤), およびそれらの合計 (緑). 灰色の点線は緑色のグラフの最大値を示しており, その時の閾値は 218pt であった.

角度の誤差

視線移動の角度の誤差とは, 指示した視線移動の方向と検出された視線移動の角度間の差異を示す. これは, *FlickSight* における視線移動の方向を識別するために必要である. Tukey-Kramer 法による多重比較の結果, いずれの *Session* 間にも有意差がないことが分かり, *Direction* による有意差があることが分かった. 図 4.10 に, *Direction* 毎の視線移動の角度の誤差, および角度の誤差に対する *Direction* の影響を示す. 有意差があった組み合わせは, -90° と 180° , 45° と 180° ($p < 0.01$), および 90° と 180° ($p < 0.05$) である.

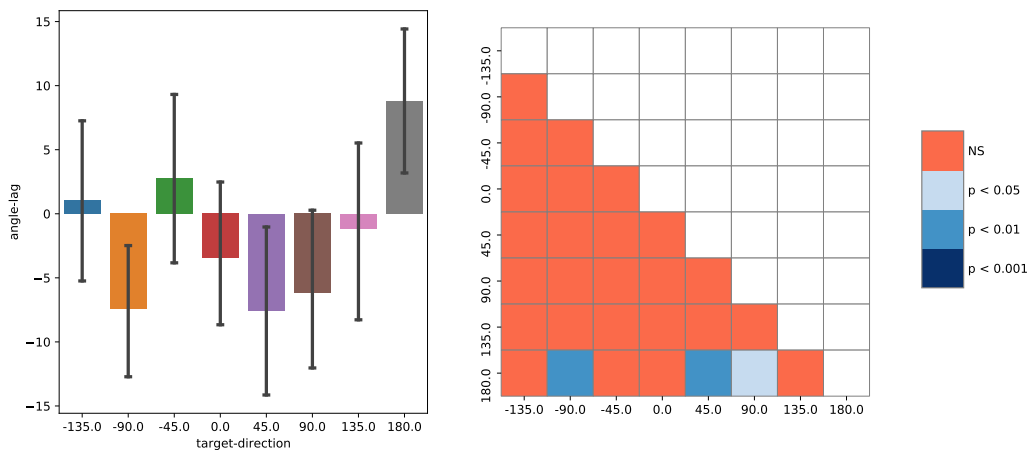


図 4.10 予備実験 3 にて検出した視線移動の角度の誤差に対する影響。
 左) *Direction* 毎の視線移動の角度の誤差のグラフ. 右) Tukey-Kramer 法による多重比較の結果.

また, *FlickSight* に採用する視線移動の方向を決定するために, 角度の誤差の絶対値に着目した. Tukey-Kramer 法による多重比較の結果, いずれの *Session* 間にも有意差がないことが分かり, *Direction* による有意差があることが分かった. 図 4.11 に, *Direction* 毎の視線移動の角度の誤差, および角度の誤差に対する *Direction* の影響を示す. 有意差があった組み合わせは, -90° と 135° ($p < 0.01$), 0° と 135° , -90° と 45° ($p < 0.05$) である. 視線移動を 8 方向に分類する場合, 隣接する方向との角度差は $45^\circ (= 360^\circ / 8)$ であるため, 任意の方向への視線移動として許容される角度の誤差は $\pm 22.5^\circ (= 45^\circ / 2)$ となる. しかしながら, 図 4.11 (左) に赤色の点線にて示した通り, 45° および 135° への視線移動は, 角度誤差の絶対値の平均が 22.5° を超えている. これは, 被験者の多くがスマートフォンを顔より低い位置にて把持していたことの影響があると考えられる. 顔より低い位置にスマートフォンがある場合, 視線移動開始時の視点が視界の中の低い位置になってしまい, 下方向 (90°) および斜め下方向 ($45^\circ, 135^\circ$) への十分な視線移動の距離を用意できなかったために, 参加者にとって方向の区別が困難であった可能性がある. すなわち, 顔より高い位置にスマートフォンを持つ参加者がいた場合, 斜め上方向 ($-45^\circ, -135^\circ$) について同様の結果が生じる可能性がある. これを踏まえて, *FlickSight* に採用する視線移動の方向は, 最大 4 方向とする.

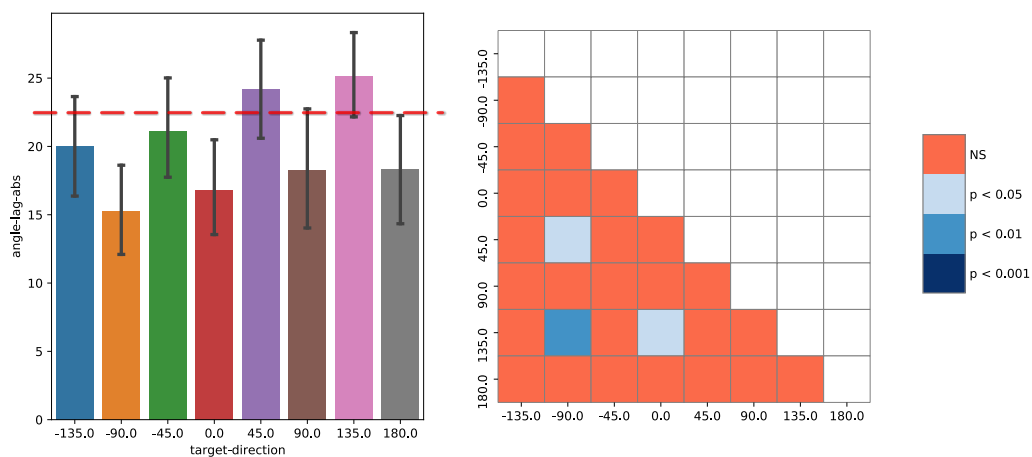


図 4.11 予備実験 3 にて検出した視線移動の角度の誤差の絶対値に対する影響。
 左) *Direction* 毎の視線移動の角度の誤差の絶対値のグラフ. 右) Tukey-Kramer 法
 による多重比較の結果.

第5章 *FlickSight* の評価実験

第4章では、*FlickSight* を設計するために実施した予備実験について述べた。本章では、その結果を基に実装した認識アルゴリズムを説明したのちに、実装した認識アルゴリズムを用いて行った、*FlickSight* の実現可能性を評価する実験について説明する。

5.1 *FlickSight* 認識アルゴリズムの実装

FlickSight の認識アルゴリズムは、フリックに関する情報および視線に関する情報を入力として受け取り、認識結果を出力する。このアルゴリズムは、Swift¹を利用して iPhone X 上に実装した。

5.1.1 入力

FlickSight の認識アルゴリズムは、フリックを検出した際に、フリックに関する情報および視線に関する情報を入力として受け取る。フリックに関する情報として、iPhone X のタッチスクリーンへのタッチを開始した座標 (F_{begin})、およびタッチを終了した座標 (F_{end}) を受け取る。視線に関する情報は、第3章に示した視線推定アルゴリズムを用いて取得し、タッチを開始した際の視線座標 (S_{begin}) を視線移動の開始点、タッチを終了した際の視線座標 (S_{end}) を視線移動の終了点として受け取る。図5.1に、視線移動の開始点および終了点を算出する概略図を示す。この図における S_{begin} は G_1 であり、 S_{end} は G_n である。

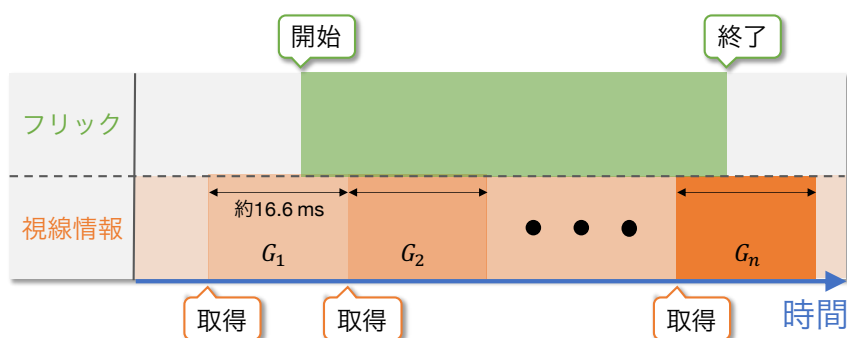


図 5.1 フリックと視線情報の取得.

¹<https://www.apple.com/swift/>

5.1.2 FlickSight の分類

FlickSight はフリックの方向および視線移動の方向によって分類される．そのため，認識アルゴリズムはそれぞれの方向を分類する．図 5.2 に，フリックの方向および視線移動の方向を分類するための変数を示す．フリックの方向は， F_{begin} と F_{end} から角度 (θ_{flick}) を算出し，その角度によって上下左右の 4 方向に分類する．視線移動の方向は，まず S_{begin} と S_{end} の間の距離 (d_{sight}) を算出し，閾値 d_{th} と比較することによって視線移動の有無を判定する．なお，閾値 d_{th} は，第 4 章にて述べた予備実験に基づき，218 pt に設定した． $d_{sight} > d_{th}$ を満たし，視線移動があると判定された場合， S_{begin} と S_{end} から角度 (θ_{sight}) を算出し，その角度によって上下左右の 4 方向に分類する．

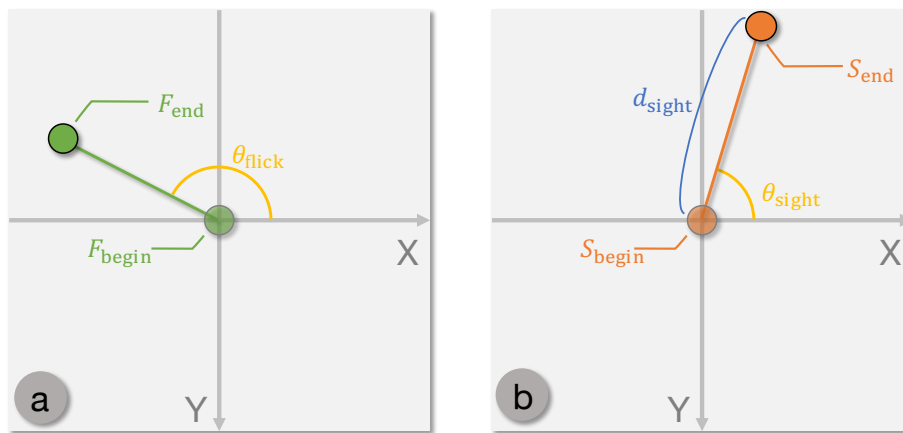


図 5.2 ジェスチャ方向の分類．(a) フリック方向の分類．(b) 視線移動方向の分類．

5.2 評価実験

設計したジェスチャは，視線移動とフリックの方向が異なる場合があり，ユーザにとって実行が困難である可能性がある．また，実行時間の短いフリックと同時に視線移動を実行することが困難である可能性がある．そのため，*FlickSight* の実現可能性を調査するための評価実験を行った．

5.2.1 参加者および実験環境

本実験の参加者は，大学生および大学院生の 8 名 (P1-P8, 21-25 歳，平均年齢 23.0 歳) であった．男性が 6 名，女性が 2 名，6 名が右利きであり，2 名が左利きであった．視力補正については，裸眼が 1 名，眼鏡着用者が 4 名，コンタクトレンズ着用者が 3 名であった．

全ての参加者は屋内の蛍光灯下にて実験に参加した．参加者は着席した状態にて，実験用のアプリケーションがインストールされたスマートフォンを把持した．参加者は，スマート

フォンのインカメラに顔が写る限り，スマートフォンの持ち方および姿勢を自由に変更して良いものとした。

5.2.2 タスクと実験手順

本実験のタスクは，指示された *FlickSight* を実行することである．1セッションの間に，20種類（ $= 4_{flicks} \times 5_{sights}$ ）の *FlickSight* がランダムな順にて一度ずつ表示される．参加者は訓練セッションの後に本番セッションを実施し，これを5回繰り返した．図5.3に実験時の指示を行うアプリケーションのスクリーンショットを示す．フリックの方向は画面中央にある緑色の四角上に矢印にて表示され，視線移動の方向は画面中央にある橙色の四角上に表示される．矢印は，視線を移動させる方向を示しており，円は，視線を移動させないことを示している．画面上部のプログレスバーは実験の進行度合いを示している．*FlickSight* が認識されると，フリックの方向および視線移動の方向が判定され，両方正解の場合には正解を示す音が，両方あるいはいずれかが間違っていた場合には不正解の音が鳴ることによって，参加者に正誤をフィードバックする．なお，訓練セッションのみ図5.3下部に示すように，認識結果を表示し，不正解の理由をフィードバックした．参加者はこの表示を基に視線移動の距離，および方向を調整する．

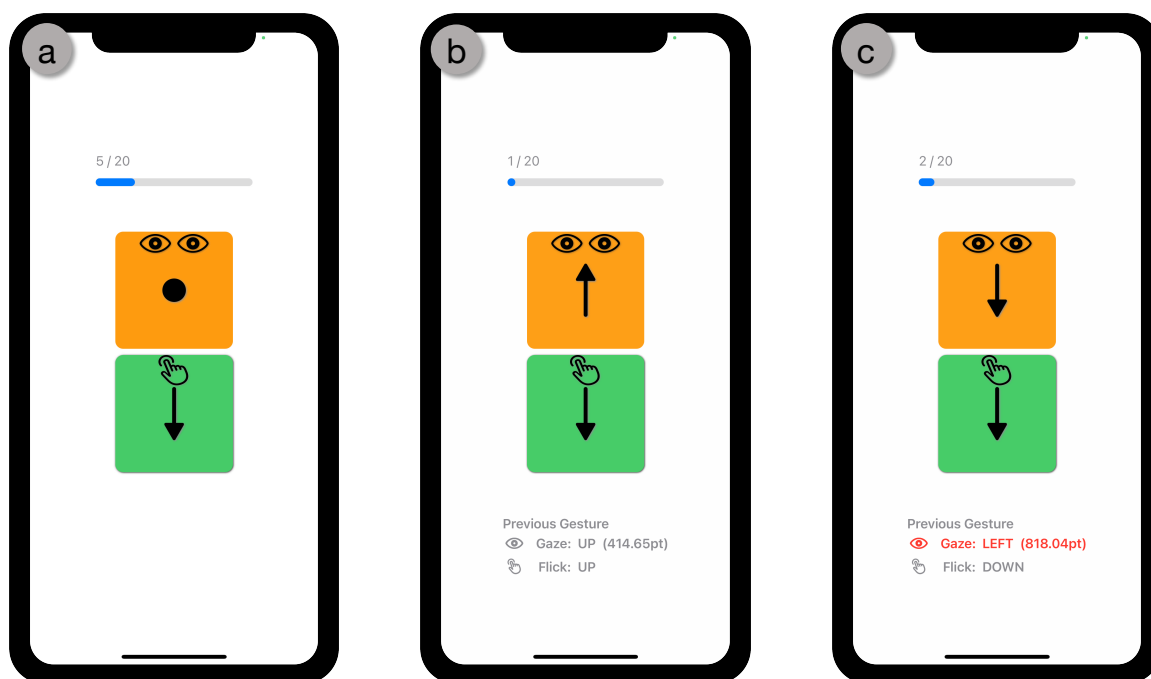


図 5.3 実験時に *FlickSight* の指示を行うアプリケーションの表示例．(a) 本番セッション．(b) 訓練セッション（直前の *FlickSight* が正解だった場合）．(c) 訓練セッション（直前の *FlickSight* が不正解だった場合）．

5.2.3 結果

本番セッションを通して合計 47,783 フレームの視線情報および 800 回の *FlickSight* を観測した。図 5.4 は、各参加者のセッション毎の成功率および平均の成功率のグラフである。フリックと視線移動の方向が両方正しかった場合を成功とした場合、成功した回数は合計 552 回であり、全体で 69.0%の成功率を得た。失敗した 248 回のうち、240 回は、フリックの方向が合っていたものの視線移動の方向を誤っていた。残りの 8 回はフリックの方向を誤っており、そのうち 4 回はフリックと視線移動の両方が誤っていた。また、フリックの方向を誤った 8 回のうち、3 回は視線移動の方向として指示された方向へフリックを行っており、視線移動の方向を誤った 244 回のうち、21 回はフリックの方向へ視線移動を行っていた（表 5.1(b)）。また、それ以外に視線移動の方向を誤った原因を表 5.1 に示す。なお、以降では視線移動の方向を *Up*（上方向）、*Down*（下方向）、*Left*（左方向）、*Right*（右方向）、*Keep*（移動なし）と表記する。

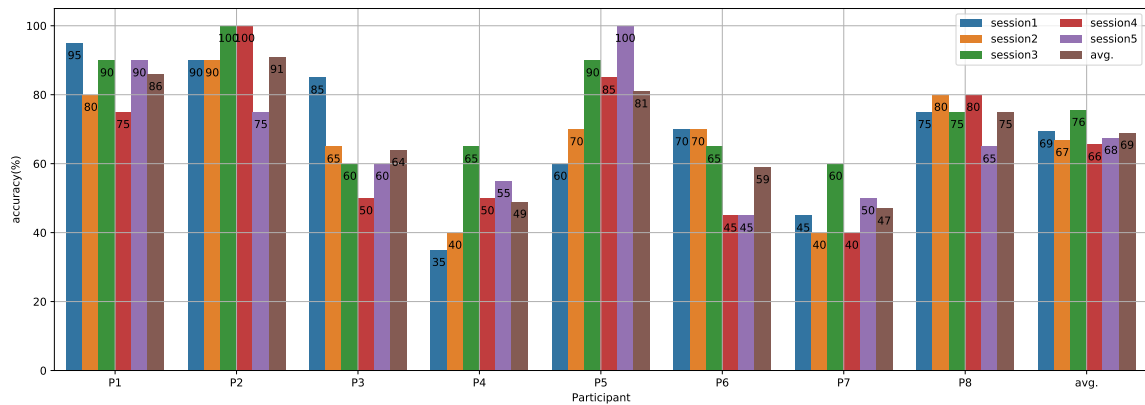


図 5.4 各参加者のセッション毎の成功率および平均の成功率。

表 5.1 視線移動の方向を誤った原因と回数. ○は, 行名の条件を満たすことを示し, ×は行名の条件を満たさないことを示す. また, *None*はその条件を考慮しないことを意味する. なお, 表中の d_{sight} , および θ_{sight} は図 5.2 にて定義したものと同様であり, θ_{target} は, 指示された視線移動の角度を指す.

	(a)	(b)	(c)	(d)	(e)	(f)	合計
フリックに要する時間 ≥ 16.7 ms	×	○	○	○	○	○	
$d_{\text{sight}} > 218$ pt	<i>None</i>	○	○	○	×	×	
指示方向が <i>Keep</i>	<i>None</i>	<i>None</i>	○	×	×	×	
$ \theta_{\text{sight}} - \theta_{\text{target}} < 45^\circ$	<i>None</i>	<i>None</i>	×	×	○	×	
指示されたフリック方向への視線移動	<i>None</i>	○	×	×	<i>None</i>	<i>None</i>	
回数	77	21	9	18	66	53	244
割合 (%)	9.6	2.6	1.1	2.3	8.3	6.6	30.5

フリックに要する時間

図 5.5 に, 観測した 800 回のフリックに要した時間の対数分布を示す. 灰色の点線は 16.7 ms を示しており, 全体の 16.7%にあたる 132 回のフリックは要する時間が 16.7 ms 未満であった. ARKit の更新速度は約 16.7 ms であり, この 132 回のフリック中には ARKit による視線情報の更新が行われなかった可能性がある. 実際にこれらのうち 121 回のフリック中には ARKit による視線情報の更新が行われなかった. 指示された視線移動方向の内訳は, *Keep* の場合が 44 回, *Left* の場合が 24 回, *Right* の場合が 20 回, *Down* の場合が 18 回, *Up* の場合が 15 回であった. フリック中に視線情報の更新が行われない場合, 視線移動の距離は 0 pt となるため *Keep* と分類される. そのため, *Keep* 以外の方向が指示されていた場合には, *Keep* として誤分類されていた (表 5.1(a)).

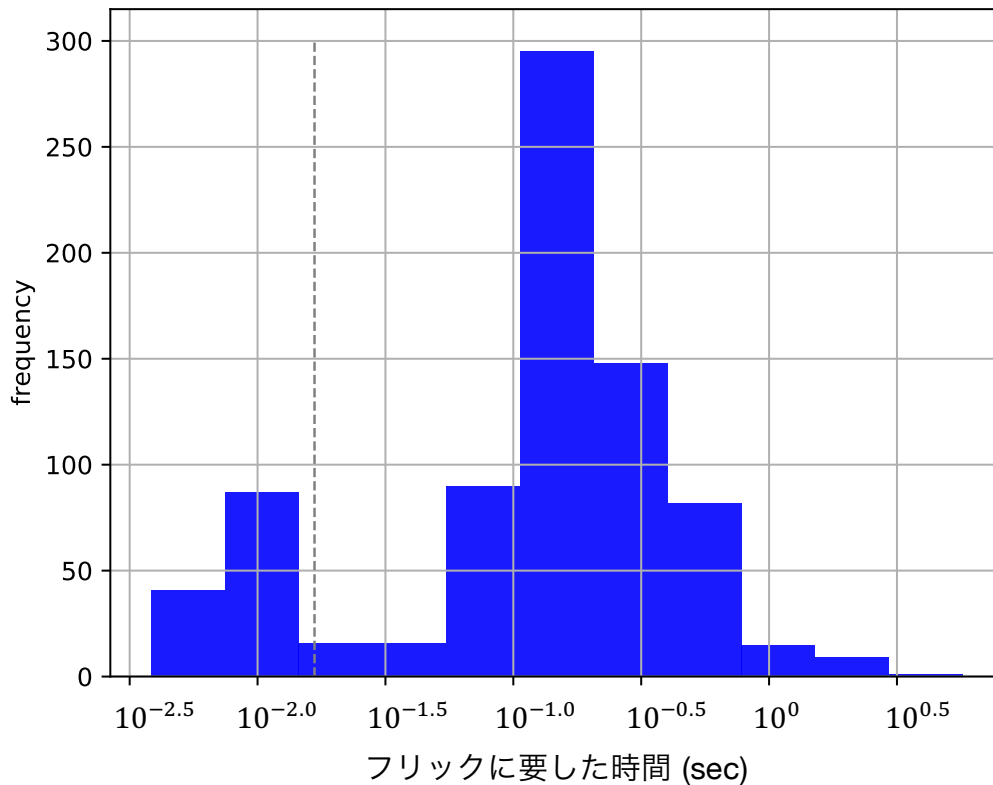


図 5.5 フリックに要した時間の対数分布.

ここで、指示された視線移動の方向毎のフリックに要する平均時間（対数表記）、および Tukey-Kramer 法による多重比較の結果を図 5.6 に示す。Tukey-Kramer 法による多重比較の結果、*Keep* と *Down* ($p < 0.001$), *Keep* と *Right*, および *Keep* と *Up* ($p < 0.01$) には有意差があり、*Keep* が指示された場合、フリックに要する時間が有意に短いことが分かった。これは、フリックと同時に視線移動を要求されることによってフリックの時間が長くなる可能性を示唆している。また、予備実験 2 にて観測したフリックに要する時間の対数分布を図 5.7 に示す。予備実験 2 にて観測されたフリックのうち、要する時間が 16.7 ms を下回ったフリックは全体の 0.8%にあたる 20 回 (2,483 回) であり、本実験よりも発生確率が低い。この原因として、実験用アプリケーションのインターフェイスがフリックキーボードのキーに似たデザインをしており、被験者がページめくりやスクロール時のフリックよりも高速なフリックの実行を求められているように感じた可能性が考えられる。

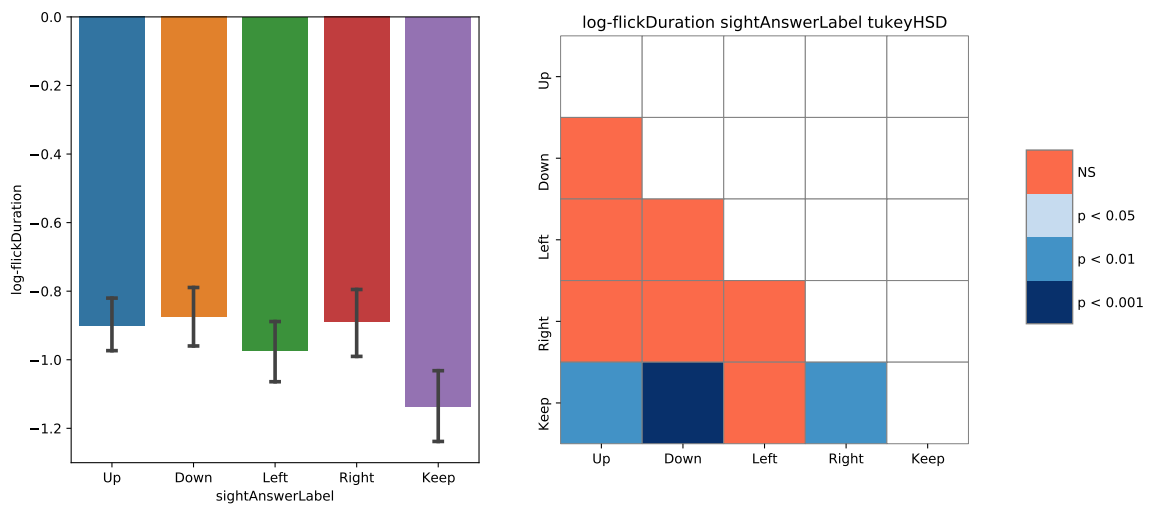


図 5.6 指示された視線移動の方向毎のフリックに要した時間 (対数表記).

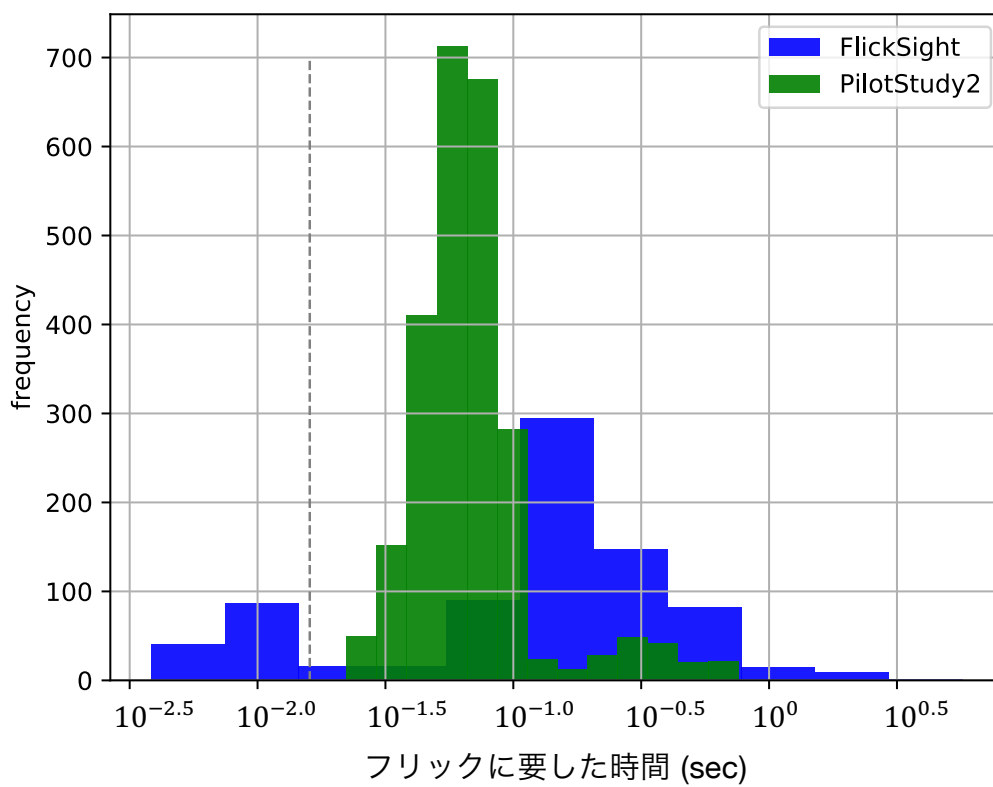


図 5.7 フリックに要した時間の対数分布. (青) 本実験にて観測されたフリック. (緑) 予備実験 2 にて観測したフリック.

視線移動の距離

図 5.8 に、視線移動方向の混同行列を示す。視線移動の方向が *Keep* と誤分類された 196 回のジェスチャのうち、77 回（表 5.1(a)）は先述したフリックに要する時間の影響を受けており、視線移動は *Keep* と誤分類されていた。残りの 119 回のうち 66 回は、視線移動の角度は正しいが移動距離が足りない失敗であった（表 5.1(e)）。指示された視線移動方向の内訳は、*Down* の場合 33 回、*Up* の場合 16 回、*Left* の場合 8 回、*Right* の場合 9 回であった。

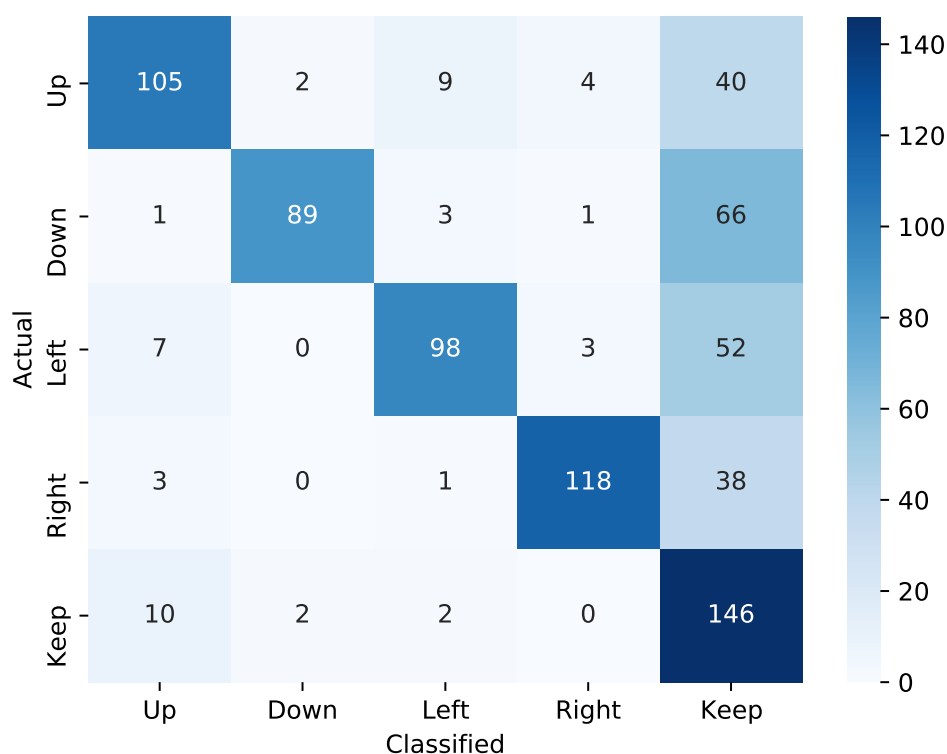


図 5.8 視線移動の方向毎の混同行列。

ここで、指示した視線移動の方向毎の移動距離を図 5.9 に示す。Tukey-Kramer 法によると、下方向が指示された場合の視線移動距離は下方向以外を指示された場合の視線移動距離と比較して有意差がある ($p < 0.001$) ことがわかった。また、上方向と右方向 ($p < 0.01$) および上方向と左方向 ($p < 0.05$) にも有意差が存在する。これは、多くの参加者が顔よりも低い位置にてスマートフォンを把持していたため、視界の下方向に十分な視線移動距離を用意できなかったことが原因であると考えられる。

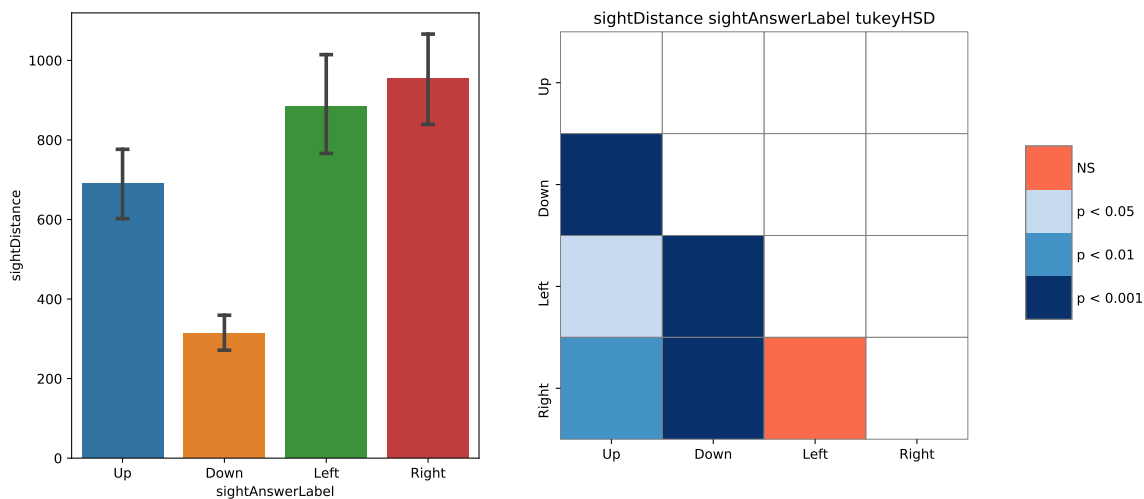


図 5.9 視線移動の方向毎の視線移動距離に対する影響。
 左) 視線移動の方向毎の距離のグラフ. 右) Tukey-Kramer 法による多重比較の結果.

視線移動の方向

視線移動の指示方向が *Keep* 以外の際に、フリックの方向以外に動いたと判定された 18 回 (表 5.1(d)) のケースについて、指示された視線移動と観測された視線移動の角度の誤差を調査した結果を表 5.2 に示す。P1 および P3 は、同じ視線移動の指示方向を同じように間違えていることが確認できる。さらに、本実験における視線移動の指示方向 θ_{target} と、実際の視線移動の方向 θ_{sight} の差の許容範囲は $\pm 45^\circ$ であり、わずかに異なってしまった場合が多いことが分かる。これは、参加者によって得意な視線移動方向と不得意な視線移動方向があることを示唆している。また P6 および P7 の結果にて、 180° に近い角度差が確認できる。これらは、フリックが完了する前に視線移動が完了してしまったために、元に戻ろうとする視線移動がフリック中に発生してしまったことが原因である可能性がある。

瞬き

実装した視線推定アルゴリズムは、瞳孔がまぶたに隠れることにより、一様の点を注視していても、視線推定の結果に影響が生じる可能性がある。視線移動の指示方向が *Keep* の際に、フリックの方向以外に動いたと判定された 9 回 (表 5.1(c)) のケースについて、判定された視線移動の方向、および視線移動開始時と終了時の両目の瞬き度合いを確認した。表 5.3 に、視線移動の指示方向が *Keep* の際に、フリックの方向以外に動いたと判定された場合の瞬き度合いを示す。なお、ARKit によって取得される瞬きの度合いは 1 以下の数値で表現され、0 は開いた状態を、1 は完全に閉じた状態を意味する。始めに、*Up* と分類された 6 つのケースに

表 5.2 視線移動の指示方向が *Keep* 以外の際に，フリックの方向以外に動いたと判定された視線移動の角度の誤差

参加者	セッション	指示方向	分類結果	指示方向と実際の視線移動の角度差 (度)
P1	2	<i>Left</i>	<i>Up</i>	47.2
	4	<i>Left</i>	<i>Up</i>	47.6
	4	<i>Left</i>	<i>Up</i>	45.9
	4	<i>Left</i>	<i>Up</i>	67.0
P3	1	<i>Right</i>	<i>Up</i>	-96.1
	2	<i>Up</i>	<i>Left</i>	-53.4
	3	<i>Up</i>	<i>Left</i>	-46.5
	4	<i>Up</i>	<i>Left</i>	-50.9
P4	2	<i>Up</i>	<i>Right</i>	78.6
P5	2	<i>Down</i>	<i>Right</i>	-48.9
P6	3	<i>Up</i>	<i>Down</i>	178.2
	4	<i>Up</i>	<i>Left</i>	-50.0
	5	<i>Left</i>	<i>Up</i>	56.4
P7	2	<i>Left</i>	<i>Right</i>	-163.6
	3	<i>Left</i>	<i>Right</i>	-168.9
	3	<i>Up</i>	<i>Left</i>	-102.9
	4	<i>Right</i>	<i>Up</i>	-74.1
	5	<i>Up</i>	<i>Left</i>	-106.8

表 5.3 視線移動の指示方向が *Keep* の際に、フリックの方向以外に動いたと判定された場合の瞬き度合い。

参加者 (平均)	セッション	視線移動の 分類結果	視線移動開始時		視線移動終了時	
			瞬き度合い	平均との差	瞬き度合い	平均との差
P3 (0.095)	1	<i>Left</i>	0.135	0.041	0.137	0.042
P5 (0.101)	2	<i>Down</i>	0.001	-0.099	0.164	0.063
	3	<i>Up</i>	0.243	0.143	0.103	0.003
P6 (0.024)	4	<i>Up</i>	0.149	0.125	0.000	-0.024
	5	<i>Up</i>	0.095	0.072	0.000	-0.024
P7 (0.077)	4	<i>Left</i>	0.000	-0.077	0.069	-0.008
	2	<i>Up</i>	0.194	0.115	0.019	-0.060
P8 (0.079)	4	<i>Up</i>	0.888	0.808	0.000	-0.079
	5	<i>Up</i>	0.178	0.099	0.022	-0.058

ついて説明する。これらは、視線移動が開始した際と終了した際の瞬き度合いを比較すると、視線移動が開始した際の瞬き度合いの方が少なくとも 0.095 高いことが確認できる。これらのケースでは視線移動開始時に、少なくとも 1 割目を閉じていることを意味している。特に、P8 session 4 のケースでは、視線移動開始時に 9 割近く目を閉じている。これらの目の閉じ具合によって、視線位置が誤って低く推定されたために、通常状態に戻った際に視線が上方向へ移動したように分類されてしまった可能性がある。同様に、*Down* と分類されているケースでは、視線移動終了時のほうが目の閉じ具合が 0.163 高いことが確認できる。この目の閉じ具合によって、視線位置が誤って低く推定され、視線移動開始時と比較して下方向へ移動したように分類されてしまった可能性がある。

5.2.4 フリックに要する時間が短い場合の補正による成功率の向上

視線移動の方向が *Keep* と分類された 342 回の視線移動は、フリックに要する時間が短いために視線移動がないと判定されている可能性がある。すなわち、フリックに要する時間が視線移動を行うには不十分である可能性がある。そのため、これらのジェスチャについて、フリック前後の視線情報を追加することによる正答率の変化を調査した。結果として、フリック開始前の視線情報は変化させず、フリックが完了した時点から 4 フレーム（約 66.8 ms）後の視線情報を視線移動終了点とすることによって、真陽性率（*Keep* 以外の方向が指示された場合に正しい方向への視線移動が発生したと認識される確率）と真陰性率（*Keep* が指示された場合に *Keep* と認識される確率）の和が最大になることが分かった。その際の真陽性率は 66.3% (130/196)，真陰性率は 87.0% (127/146) である。

そのため、本実験にて記録された *FlickSight* のうち、フリックに要する時間が 131.7 ms 未満の *FlickSight* について、フリックが完了した時点から 4 フレーム後の視線情報を視線移動終了

点とする補正を行った。なお、フリックに要する時間が 131.7 ms 未満の *FlickSight* は、*Keep* 以外が指示されたが *Keep* と判定された 196 回のうち 147 回 (75.0%) の *FlickSight* と、*Keep* 以外が指示され、正しい方向への視線移動があったと判定された 410 回のうち 104 回 (25.4%) の *FlickSight* を含む。この補正によって、P1 を除いた全ての参加者は平均成功率が向上し、特に補正前に成功率が平均以下であった P3, P4, P6, P7 については、それぞれ 20.0%, 29.0%, 11.0%, 22.0% の成功率向上が確認され、全体を通した成功率は 80.8% に向上する。なお、P1 は補正前後にて平均成功率が変化しなかった。この補正を行った場合の全ての *FlickSight* の混同行列を図 5.10 に示す。頻繁に発生する視線移動の誤認識は、*Down* と *Keep*、および *Keep* と *Up* であるが、これらはスマートフォンを持つ位置および瞬きの影響である可能性がある。

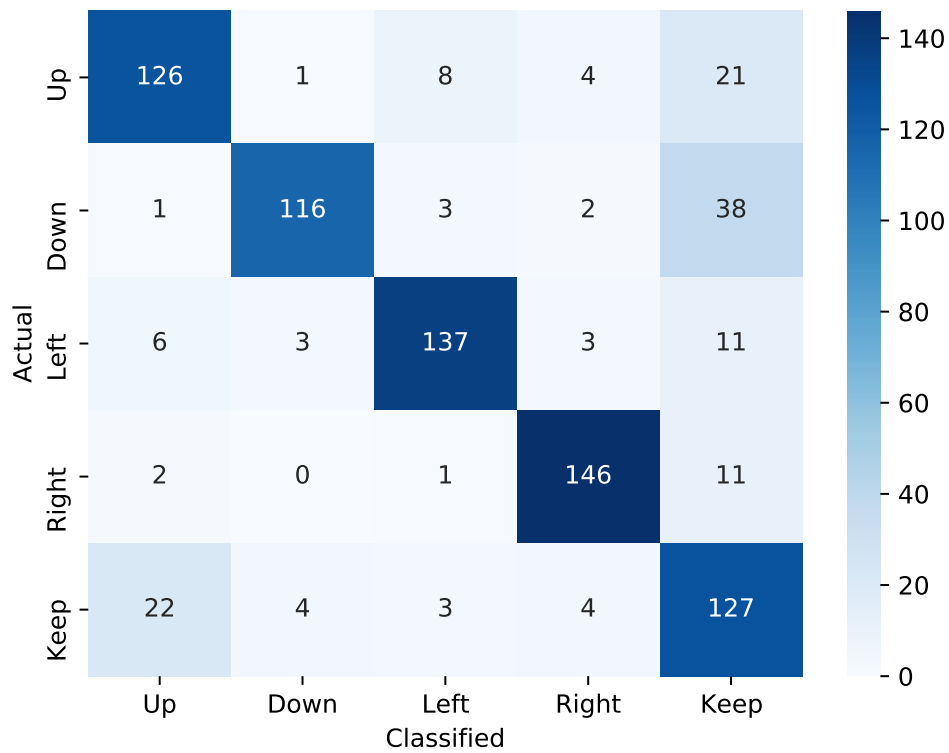


図 5.10 補正を行った際の視線移動の方向毎の混同行列。

第6章 *FlickSight* を用いた操作例

本研究にて提案する *FlickSight* はスマートフォンに対する入力手法であり，主に一本指による操作の拡張を行うことができると考えられる．本章では，*FlickSight* による具体的な操作の例を示す．

6.1 2本の指を用いる操作への割り当て

スマートフォンを片手にて操作する際には，親指のみによる操作が好まれている [NBW14, KBCV06]．しかしながら，2本の指を用いたタッチジェスチャにはピンチインやピンチアウトなどがあり，これらのタッチジェスチャには画面の縮小や拡大が割り当てられることが多く，地図アプリケーションを操作する際などに用いられている．そこで，*FlickSight* を用いることによって，2本の指を用いたタッチジェスチャに割り当てられた操作を行うことが考えられる．これは，*FlickSight* による視線移動を2本目の指の代替として用いるような操作である．例えば，*Flick-Down-Sight-Reverse* によって画面の拡大を行い，*Flick-Up-Sight-Reverse* によって画面の縮小を行うことができると考えられる．図6.1に，*Flick-Up-Sight-Reverse* によって画像の縮小を行っている例を示す．

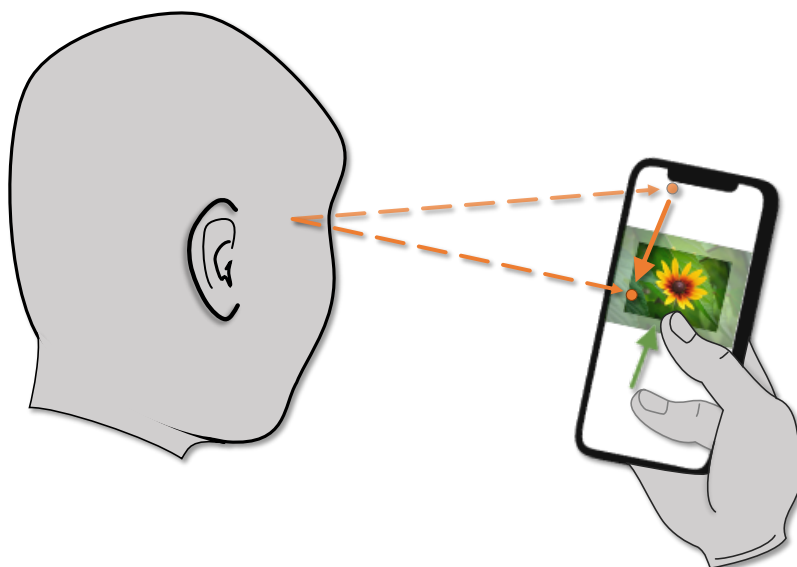


図 6.1 *Flick-Up-Sight-Reverse* によって画像の縮小を行っている例．

6.2 ベゼルや物理ボタンを用いる操作への割り当て

スマートフォンを操作する際には、しばしばベゼルや物理ボタンを用いることが求められる。その際にユーザの把持姿勢によっては、ベゼルや物理ボタンに指が届かないために把持姿勢を調整する必要がある可能性がある。しかし、頻繁なスマートフォンの把持姿勢変更は把持の安定を損なうため、ユーザの快適な操作を妨げることが報告されている [ERGT17, ERGT18]。加えて、スマートフォンの把持姿勢変更は、スマートフォンの落下につながる危険性がある。そこで、*FlickSight* を用いることによって、ベゼルや物理ボタンによる入力に割り当てられた操作を行うことが考えられる。

ベゼルを用いる操作の代替を行う場合、タッチパッドにて利用される3本以上の指を同じ方向にフリックするタッチジェスチャのように *FlickSight-Follow* を実行することが考えられる。例えば、*Flick-Up-Sight-Follow* によって起動中のアプリケーションを確認する画面へ遷移する(図6.2)ことができ、*Flick-Down-Sight-Follow* によってコントロールセンターを表示することができる。

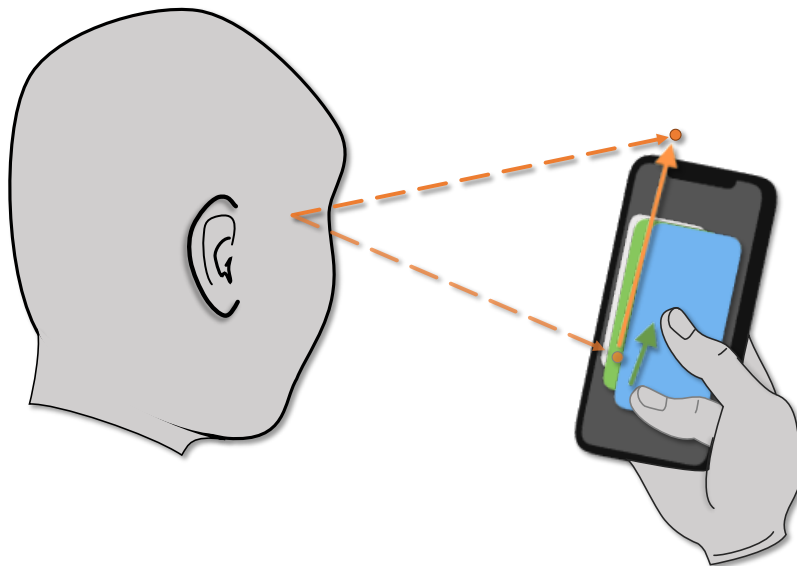


図 6.2 *Flick-Up-Sight-Follow* によって起動中のアプリケーションを確認する画面へ遷移する例。

物理ボタンを用いる操作の代替を行う場合、スマートフォンに搭載された物理ボタンを押すようなイメージでフリックを行い、視線移動によってその詳細を指定するように *FlickSight* を実行することが考えられる。例えば、*Flick-Left-Sight-Left* によって画面の点灯消灯を切り替えることができ、*Flick-Right-Sight-Up* によって音量をあげ、*Flick-Right-Sight-Down* によって音量を下げるができる。また、図 6.3 のように *Flick-Left-Sight-Right* によってスクリーンショットの撮影を行うことも考えられる。

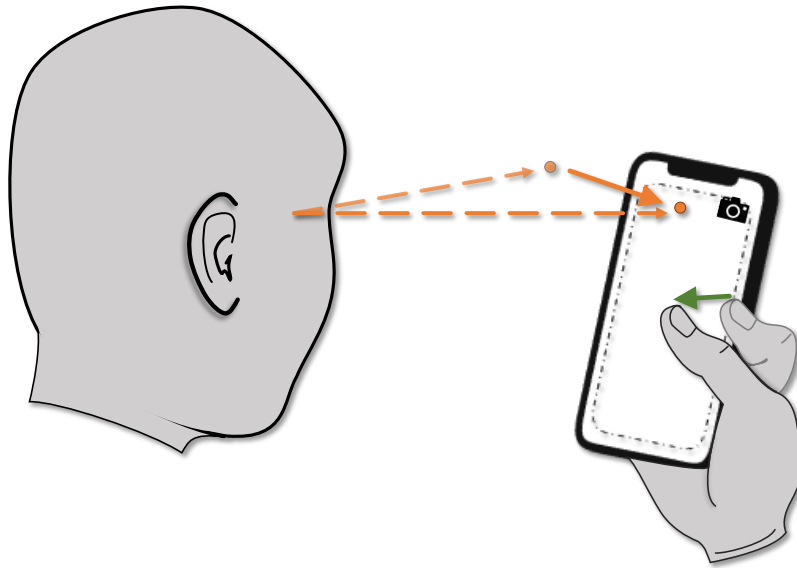


図 6.3 *Flick-Left-Sight-Right* によってスクリーンショットの撮影を行う例.

6.3 画面内における指の届きづらい範囲にて実行される操作への割り当て

スマートフォンを片手にて操作する際には、指が届きづらい範囲の操作を要求されることがある。このようなケースにおいてユーザは、もう一方の手を用いる、あるいは把持姿勢を変更することなどによって操作を完了する。指の届きづらい範囲における操作の例として、画面上部にて下方向へスワイプすることによるモーダルウィンドウの立ち下げや、画面左上にあるボタンを押すことによる直前の画面への遷移がある。例えば、地図にてある地点を検索した際にその地点、あるいは建物の詳細がモーダルウィンドウに表示される。しかし、モーダルウィンドウによって地図の一部、あるいは全部が隠れてしまい、地図を確認するためにモーダルウィンドウを閉じることがある。また、フォトギャラリーにて個別の写真を選択表示している状態から、フォトギャラリー画面へ遷移することがある。このような場面において、*FlickSight* を利用することによって、これらの操作を片手のみ、かつ把持姿勢を変更することなく実行できると考えられる。例えば地図にてある地点を検索し、モーダルウィンドウが表示されている状態にて *Flick-Down-Sight-Down* を実行することによって、モーダルを立ち下げ地図を表示することができる。また、フォトギャラリーにて個別の写真を選択表示している状態にて、*Flick-Left-Sight-Follow* を実行することによってフォトギャラリー画面へ遷移することができる。

6.4 フリックの拡張

スマートフォンにて Web ページを閲覧する際、多くのページにて縦方向のスクロールによる閲覧が要求される。このスクロールは主にフリックによって行われ、その距離や速度によってスクロールの距離が変わる。そこで、*FlickSight-Follow* を用いることによって、フリックの効果を強めることができると考えられる。例えば、上方向へのスクロール時に *Flick-Up-Sight-Follow* を実行することによって、上方向へのスクロールの距離を伸ばすことができる。同様に、下方向へのスクロール時に *Flick-Down-Sight-Follow* を実行することによって、下方向へのスクロールの距離を伸ばすことができる。

6.5 メニューの多いアプリケーションにおけるショートカット

スマートフォンにてスプレッドシートを利用する際、操作するためのメニューによって表の一部が隠れてしまうことがある。そこで、*FlickSight* をショートカットとして利用することが考えられる。例えば、*Flick-Right-Sight-Up* によって選択したセルのコピー、*Flick-Right-Sight-Right* によって選択したセルのカット、*Flick-Right-Sight-Down* によって選択したセルのペーストが可能になる。また、*Flick-Left-Sight-Right* によって列の挿入、*Flick-Left-Sight-Left* によって列の削除、*Flick-Up-Sight-Down* によって行の挿入、*Flick-Up-Sight-Up* によって行の削除が可能になる。

第7章 今後の課題

本章では、予備実験、*FlickSight* の設計、認識アルゴリズムの実装、および評価実験を通して明らかになった事項について議論し、今後の課題について述べる。

7.1 *FlickSight* の視線移動距離

第5章における評価実験時に用いた *FlickSight* の認識アルゴリズムは、視線移動があると判定するための閾値を第4章における予備実験に基づいて 218 pt に設定した。本節ではこの閾値が適切であったかどうかについて議論する。

第5章における評価実験では、フリックに要する時間が短い際に補正を加えることによって成功率が 11.8% 向上し、全体で 80.8% の成功率が確認できた。これは、フリックに要する時間が短く、閾値以上の視線移動をするのに十分な時間を確保できない場合があったことを示唆している。また、補正後の視線移動の距離を調査した結果、*Keep* 以外が指示され、指示された方向への視線移動が検出された 525 回の視線移動のうち 85.7% にあたる 450 回は実験に用いた iPhone X の横幅である 375 pt よりも距離が長く、59.0% にあたる 310 回は iPhone X の縦の長さである 812 pt よりも距離が長いことがわかった。この実験は、視線移動があると判定するための閾値が 218 pt であること、および 218 pt とは画面横幅のおよそ 6 割程度であることを示した上で行ったにも関わらず、実際には画面の横幅よりも長い視線移動が発生していることが確認できる。これは、ユーザが視線移動の距離を高速かつ正確に調整することが困難であることを示す結果であると考えられる。

また、補正によって *Keep* の成功率は 91.3% (146/160) から 79.4% (127/160) へと低下した。これは、補正によってフリック完了後の視線移動を取り入れたため、フリックが完了した後に視線を自由に動かした際の視線移動が反映されている可能性がある。また、フリックによるデバイスの振動によって視点推定結果に閾値以上の誤差が生じてしまった可能性も考えられる。

これらの誤認識に対処するために閾値を長く設定することができるが、距離の長い視線移動はユーザにとって負担となる可能性がある。一方で閾値を短く設定すると、誤認識の問題に加え、意図しない *FlickSight* を回避するためにユーザがストレスを感じる可能性がある。そのため、複数の閾値によって成功率、視線を動かす負担、あるいは視線を留めておくストレスについて調査することが必要であると考えられる。

7.2 指と視線を意図的に異なる方向へ動かすことによる負担

第5章における評価実験において、「指と視線を違う方向に動かす際に混乱することがあった」という参加者があった。第5章における評価実験では、フリックの方向として上下左右の4方向、視線移動の方向として上下左右に視線移動なしを加えた5種類を用意した。これは第4章に示したフリックの方向と視線移動の方向を任意の数に設定した入力手法に基づいて設計されている。しかし、指と視線をランダムな組み合わせにて異なる方向へ動かすことは、ユーザの混乱を招く可能性があるため、視線移動の方向を限り、フリックを基準とした方向にすることが考えられる。これは、第4章に示したフリックの方向を基準とした視線移動の方向を利用する入力手法に基づいて設計することが考えられる。

また運動器である手とは異なり、目はあくまで感覚器であり、「見る」こと以外を目的として視線を動かすことは稀である。しかし *FlickSight* は、「見る」こと以外を目的として意図的に視線を動かすことを要求する。そのため、視線を意図的に動かすことによる疲労感、あるいは公共の場で行うことに対する抵抗感などについての調査が必要であると考えられる。

7.3 実験環境および実験参加者の偏り

第5章における評価実験の参加者は、全員が21歳以上25歳以下であり、日常的にスマートフォンを利用していた。参加者全員が若年であったことから、視線移動を明確に行えた可能性があり、また、スマートフォンの操作に慣れていたため、フリックに要する時間をわずかに調整することができた可能性がある。しかし、スマートフォンのユーザは幅広い年齢層および使用歴に分布している。例えば高齢者は、若者に比べて視線移動が困難である可能性があり、スマートフォン初心者はフリックに要する時間を調整することが難しいかもしれない。そのため、さまざまな属性を持つ参加者を増やして追加の実験を行う必要があると考えられる。

また、第3章と第4章における予備実験、および第5章における評価実験において、参加者は常に着席していた。しかし日常的な環境においてユーザは、立った状態や寝転がった状態、あるいは歩いている状態などにおいてもスマートフォンを利用する。それらの環境では、着席している場合と比較してスマートフォンの安定度が異なることが報告されている [ERGT17, ERGT18, IOES20]。第3章にて説明した視線推定アルゴリズムでは、拡張現実空間におけるスマートフォン自身の位置が視線推定精度に影響する。そのため、ユーザの姿勢が変化することによって視線推定精度に影響がおよぶ可能性がある。また、実装した視線推定アルゴリズムは、視線推定のためにユーザの顔全体がスマートフォンインカメラの撮影範囲内に存在する必要がある。しかしながら、日常的な環境におけるスマートフォン使用時に顔全体がインカメラに映り込んでいる可能性は29%に過ぎないことが報告されている [KAB18]。そのため、顔全体がインカメラに映り込んでいない状態のフリックを通常のフリックとすることができる可能性がある。一方で、*FlickSight* の日常的な環境における試験を行うためには、異なる視線推定方法を実装する必要があると考えられる。

第8章 結論

本論文において、フリックと視線移動を組み合わせたスマートフォン向けの新たな入力手法である *FlickSight* を提案した。ユーザは、視線移動による入力を2本目の指と見なすことにより、片手操作時であってもピンチインなどのマルチタッチジェスチャの代替となる操作を実行できる。また、把持姿勢を変更しないまま物理ボタンやベゼルを利用する操作を実行できる。

これを実現するために、スマートフォンによる視線推定システムを実装し、その性能を調査するための予備実験を行った。結果として、視線移動を検出するには十分な精度をもつことが分かった。続いて、*FlickSight* について説明し、入力手法を設計するために2つの予備実験を行った。1つ目の予備実験では、日常的な操作時のフリック中に発生する視線移動の距離について調査を行った。結果として、実験に用いた iPhone X の長さを超えるような距離の長い視線移動が発生したのは、実験中に観察された 2,446 回のフリックのうち 7.8% にあたる 191 回のみであった。2つ目の予備実験では、直線的な視線移動を指示された場合に参加者が実行した視線移動の距離および角度について調査を行った。2つの予備実験から、*FlickSight* が通常のフリックと競合しないための閾値として 218 pt が適切であることが分かった。また、視線移動の方向を上下左右および斜めの 8 方向に設定した場合、特に斜め方向についてユーザが正確に視線移動を行えない可能性が示唆されたため、最大の方向数を上下左右の 4 方向に設定した。これらの結果を元に、*FlickSight* の認識アルゴリズムを実装し、*FlickSight* の実現可能性を調査するための評価実験を行った。結果として、平均で 69.0% の成功率を得たが、下方向への視線移動を視線移動なし、および視線移動なしを上方向への視線移動とする誤認識が頻繁に確認された。これらは、顔とスマートフォンの相対的な位置関係、および瞬きが影響していると考えられる。またフリックに要する時間が短い場合に、フリック後の視線移動を考慮する補正を加えることによって 80.8% の成功率を得た。今後の課題として、視線移動を操作に組み込むことによる視線の振る舞いへの影響を調査すること、および疲労感や公共の場にて行うことに対する抵抗感などについての調査が必要であると考えられる。

謝辞

本論文を執筆するにあたり，高橋伸准教授，志築文太郎教授には多大なご助力を賜り，感謝致します．特に高橋伸准教授には論文の執筆方法だけでなく，研究生活においても丁寧かつ熱心なご指導を賜りました．ここに深い感謝の意を表します．

インタラクティブプログラミング研究室の皆様には，研究における様々なご支援を頂きました．特に UBIQUITOUS チームの皆様にはチームゼミをはじめ，論文執筆を行った際には多くのご助言や添削など，多くのご支援を頂きました．感謝いたします．さらに，研究生活においてメンターとして指導して頂いた昨年度の修了生である関森 広大さんと，私の携わった全ての研究において多大なるご助力を頂いた大和 優輝さんに深く感謝いたします．

また，私の生活を支えて下さった友人，学生生活にてお世話になった皆様にお礼申し上げます．最後に，私の成長を見守り，学生生活を支えてくださった家族に深く感謝いたします．

参考文献

- [BG10] Andreas Bulling and Hans Gellersen. Toward Mobile Eye-Based Human-Computer Interaction. *IEEE Pervasive Computing*, Vol. 9, No. 4, pp. 8–12, 2010.
- [BLC⁺12] Sebastian Boring, David Ledo, Xiang ‘Anthony’ Chen, Nicolai Marquardt, Anthony Tang, and Saul Greenberg. The Fat Thumb: Using the Thumb’s Contact Size for Single-Handed Mobile Interaction. In *Proceedings of the 14th International Conference on Human-Computer Interaction with Mobile Devices and Services, MobileHCI ’12*, pp. 39–48, New York, NY, USA, 2012. Association for Computing Machinery.
- [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 Symposium on Eye Tracking Research and Applications, ETRA ’12*, pp. 83–90, New York, NY, USA, 2012. Association for Computing Machinery.
- [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. Association for Computing Machinery.
- [DDLS07] Heiko Drewes, Alexander De Luca, and Albrecht Schmidt. Eye-Gaze Interaction for Mobile Phones. In *Proceedings of the 4th International Conference on Mobile Technology, Applications, and Systems and the 1st International Symposium on Computer Human Interaction in Mobile Technology, Mobility ’07*, pp. 364–371, New York, NY, USA, 2007. Association for Computing Machinery.
- [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. 1–13, New York, NY, USA, 2017. Association for Computing Machinery.
- [DS07] Heiko Drewes and Albrecht Schmidt. Interacting with the Computer Using Gaze Gestures. In Cécilia Baranauskas, Philippe Palanque, Julio Abascal, and Simone Diniz Junqueira Barbosa, editors, *Human-Computer Interaction – INTERACT 2007*, pp. 475–488, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

- [ERGT17] Rachel Eardley, Anne Roudaut, Steve Gill, and Stephen J. Thompson. Understanding Grip Shifts: How Form Factors Impact Hand Movements on Mobile Phones. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, pp. 4680–4691, New York, NY, USA, 2017. Association for Computing Machinery.
- [ERGT18] Rachel Eardley, Anne Roudaut, Steve Gill, and Stephen J. Thompson. Investigating How Smartphone Movement is Affected by Body Posture. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, pp. 1–8, New York, NY, USA, 2018. Association for Computing Machinery.
- [EWA14] Hanene Elleuch, Ali Wali, and Adel M. Alimi. Smart Tablet Monitoring by a Real-Time Head Movement and Eye Gestures Recognition System. In *2014 International Conference on Future Internet of Things and Cloud*, pp. 393–398, 2014.
- [EWA16] Hanene Elleuch, Ali Wali, and Adel M. Alimi. A Real-Time Eye Gesture Recognition System Based on Fuzzy Inference System for Mobile Devices Monitoring. In Jacques Blanc-Talon, Cosimo Distanto, Wilfried Philips, Dan Popescu, and Paul Scheunders, editors, *Advanced Concepts for Intelligent Vision Systems*, pp. 172–180, Cham, 2016. Springer International Publishing.
- [EWSA17] Hanene Elleuch, Ali Wali, Anis Samet, and Adel M. Alimi. Interacting with Mobile Devices by Fusion Eye and Hand Gestures Recognition Systems Based on Decision Tree Approach. In Antanas Verikas, Petia Radeva, Dmitry P. Nikolaev, Wei Zhang, and Jianhong Zhou, editors, *Ninth International Conference on Machine Vision (ICMV 2016)*, Vol. 10341, pp. 62–66. International Society for Optics and Photonics, SPIE, 2017.
- [GVA20] Robert Greinacher and Jan-Niklas Voigt-Antons. Accuracy Assessment of ARKit 2 Based Gaze Estimation. In *International Conference on Human-Computer Interaction*, pp. 439–449. Springer, 2020.
- [HAKI17] Khalad Hasan, David Ahlström, Junhyeok Kim, and Pourang Irani. AirPanes: Two-Handed Around-Device Interaction for Pane Switching on Smartphones. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, pp. 679–691, New York, NY, USA, 2017. Association for Computing Machinery.
- [HIST19] Kyohei Hakka, Toshiya Isomoto, Buntarou Shizuki, and Shin Takahashi. Bounded Swipe: Swipe Gesture Inside a Target. In *Proceedings of the 31st Australian Conference on Human-Computer-Interaction*, OZCHI '19, pp. 312–316, New York, NY, USA, 2019. Association for Computing Machinery.

- [HKAI16] Khalad Hasan, Junhyeok Kim, David Ahlström, and Pourang Irani. Thumbs-Up: 3D Spatial Thumb-Reachable Space for One-Handed Thumb Interaction on Smartphones. In *Proceedings of the 2016 Symposium on Spatial User Interaction, SUI '16*, pp. 103–106, New York, NY, USA, 2016. Association for Computing Machinery.
- [HL12] Seongkook Heo and Geehyuk Lee. *ForceDrag*: Using Pressure as a Touch Input Modifier. In *Proceedings of the 24th Australian Computer-Human Interaction Conference, OzCHI '12*, pp. 204–207, New York, NY, USA, 2012. Association for Computing Machinery.
- [HL13] Seongkook Heo and Geehyuk Lee. Ta-Tap: Consecutive Distant Tap Operations for One-Handed Touch Screen Use. In *Proceedings of the Adjunct Publication of the 26th Annual ACM Symposium on User Interface Software and Technology, UIST '13 Adjunct*, pp. 91–92, New York, NY, USA, 2013. Association for Computing Machinery.
- [HLNL17] Michael Xuelin Huang, Jiajia Li, Grace Ngai, and Hong Va Leong. ScreenGlint: Practical, In-Situ Gaze Estimation on Smartphones. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, CHI '17*, pp. 2546–2557, New York, NY, USA, 2017. Association for Computing Machinery.
- [HSH11] Chris Harrison, Julia Schwarz, and Scott E. Hudson. TapSense: Enhancing Finger Interaction on Touch Surfaces. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology, UIST '11*, pp. 627–636, New York, NY, USA, 2011. Association for Computing Machinery.
- [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. 1–7, New York, NY, USA, 2018. Association for Computing Machinery.
- [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. Association for Computing Machinery.
- [Inc20a] Apple Inc. ARKit-Docummentation-Apple Developer. <https://developer.apple.com/documentation/arkit>, 2020. (accessed 2020-12-25).
- [Inc20b] Apple Inc. Gestures-User Interaction-iOS-Human Interface Guidelines-Apple Developer. <https://developer.apple.com/design/>

human-interface-guidelines/ios/user-interaction/gestures/, 2020. (accessed 2020-12-25).

- [Inc20c] Apple Inc. World Tracking-ARKit-Documentation-Apple Developer. https://developer.apple.com/documentation/arkit/world_tracking, 2020. (accessed 2020-12-25).
- [IOES20] Kaori Ikematsu, Haruna Oshima, Rachel Eardley, and Itiro Sii. Investigating How Smartphone Movement is Affected by Lying Down Body Posture. *Proceedings of ACM Human Computer Interaction*, Vol. 4, No. ISS, pp. 1–17, November 2020.
- [IY20a] Kaori Ikematsu and Shota Yamanaka. ScraTouch: Extending Interaction Technique Using Fingernail on Unmodified Capacitive Touch Surfaces. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, Vol. 4, No. 3, pp. 1–19, September 2020.
- [IY20b] Kaori Ikematsu and Shota Yamanaka. ScraTouch: Extending Touch Interaction Technique Using Fingernail on Capacitive Touch Surfaces. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI EA '20, pp. 1–10, New York, NY, USA, 2020. Association for Computing Machinery.
- [Jac90] Robert J. K. Jacob. What You Look at is What You Get: Eye 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. Association for Computing Machinery.
- [Jac91] Robert J. K. Jacob. The Use of Eye Movements in Human-Computer Interaction Techniques: What You Look at is What You Get. *ACM Transactions on Information Systems (TOIS)*, Vol. 9, No. 2, pp. 152–169, April 1991.
- [JK03] 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. In J. Hyönä, R. Radach, and H. Deubel, editors, *The Mind's Eye*, pp. 573–605. North-Holland, Amsterdam, 2003.
- [JM16] Shahram Jalaliniya and Diako Mardanbegi. EyeGrip: Detecting Targets in a Series of Uni-Directional Moving Objects Using Optokinetic Nystagmus Eye Movements. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI' 16, pp. 5801–5811, New York, NY, USA, 2016. Association for Computing Machinery.

- [KAB18] Mohamed Khamis, Florian Alt, and Andreas Bulling. The Past, Present, and Future of Gaze-Enabled Handheld Mobile Devices: Survey and Lessons Learned. In *Proceedings of the 20th International Conference on Human-Computer Interaction with Mobile Devices and Services*, MobileHCI '18, pp. 1–17, New York, NY, USA, 2018. Association for Computing Machinery.
- [KAH⁺16] Mohamed Khamis, Florian Alt, Mariam Hassib, Emanuel von Zezschwitz, Regina Hasholzner, and Andreas Bulling. GazeTouchPass: Multimodal Authentication Using Gaze and Touch on Mobile Devices. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, CHI EA '16, pp. 2156–2164, New York, NY, USA, 2016. Association for Computing Machinery.
- [KB16] Dominik Kirst and Andreas Bulling. On the Verge: Voluntary Convergences for Accurate and Precise Timing of Gaze Input. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, CHI EA '16, pp. 1519–1525, New York, NY, USA, 2016. Association for Computing Machinery.
- [KBCV06] Amy K Karlson, Benjamin B Bederson, and Jose L Contreras-Vidal. Studies in One-Handed Mobile Design: Habit, Desire and Agility. In *Proceedings of 4th ERCIM Workshop User Interfaces All (UI4ALL)*, pp. 1–10. Citeseer, 2006.
- [KHBA17] Mohamed Khamis, Regina Hasholzner, Andreas Bulling, and Florian Alt. GTmoPass: Two-Factor Authentication on Public Displays Using Gaze-Touch Passwords and Personal Mobile Devices. In *Proceedings of the 6th ACM International Symposium on Pervasive Displays*, PerDis '17, pp. 1–9, New York, NY, USA, 2017. Association for Computing Machinery.
- [KHMS20] Chandan Kumar, Ramin Hedeshy, I. Scott MacKenzie, and Steffen Staab. TAGSwipe: Touch Assisted Gaze Swipe for Text Entry. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, pp. 1–12, New York, NY, USA, 2020. Association for Computing Machinery.
- [KHZ⁺17] Mohamed Khamis, Mariam Hassib, Emanuel von Zezschwitz, Andreas Bulling, and Florian Alt. GazeTouchPIN: Protecting Sensitive Data on Mobile Devices Using Secure Multimodal Authentication. In *Proceedings of the 19th ACM International Conference on Multimodal Interaction*, ICMI '17, pp. 446–450, New York, NY, USA, 2017. Association for Computing Machinery.
- [KKK⁺16] Kyle Krafka, Aditya Khosla, Petr Kellnhofer, Harini Kannan, Suchendra Bhandarkar, Wojciech Matusik, and Antonio Torralba. Eye Tracking for Everyone. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2176–2184, 2016.

- [LF13] Wing Ho Andy Li and Hongbo Fu. BezelCursor: Bezel-Initiated Cursor for One-Handed Target Acquisition on Mobile Touch Screens. In *SIGGRAPH Asia 2013 Symposium on Mobile Graphics and Interactive Applications*, SA '13, p. 1, New York, NY, USA, 2013. Association for Computing Machinery.
- [LKB⁺18] Huy Viet Le, Thomas Kosch, Patrick Bader, Sven Mayer, and Niels Henze. Palm-Touch: Using the Palm as an Additional Input Modality on Commodity Smartphones. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, pp. 1–13, New York, NY, USA, 2018. Association for Computing Machinery.
- [MGA⁺15] Alexander Mariakakis, Mayank Goel, Md Tanvir Islam Aumi, Shwetak N. Patel, and Jacob O. Wobbrock. SwitchBack: Using Focus and Saccade Tracking to Guide Users' Attention for Mobile Task Resumption. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pp. 2953–2962, New York, NY, USA, 2015. Association for Computing Machinery.
- [MLGH10] Emilie Møllenbach, Martin Lillholm, Alastair Gail, and John Paulin Hansen. Single Gaze Gestures. In *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications*, ETRA '10, pp. 177–180, New York, NY, USA, 2010. Association for Computing Machinery.
- [MLKI19] Päivi Majaranta, Jari Laitinen, Jari Kangas, and Poika Isokoski. Inducing Gaze Gestures by Static Illustrations. In *Proceedings of the 11th ACM Symposium on Eye Tracking Research & Applications*, ETRA '19, pp. 1–5, New York, NY, USA, 2019. Association for Computing Machinery.
- [MMIS09] David C. McCallum, Edward Mak, Pourang Irani, and Sriram Subramanian. PressureText: Pressure Input for Mobile Phone Text Entry. In *CHI '09 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '09, pp. 4519–4524, New York, NY, USA, 2009. Association for Computing Machinery.
- [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 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, pp. 2558–2570, New York, NY, USA, 2017. Association for Computing Machinery.
- [NBW14] Alexander Ng, Stephen A. Brewster, and John H. Williamson. Investigating the Effects of Encumbrance on One- and Two- Handed Interactions with Mobile Devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*,

- CHI '14, pp. 1981–1990, New York, NY, USA, 2014. Association for Computing Machinery.
- [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. Association for Computing Machinery.
- [NYS10] Takashi Nagamatsu, Michiya Yamamoto, and Hiroshi Sato. MobiGaze: Development of a Gaze Interface for Handheld Mobile Devices. In *CHI '10 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '10, pp. 3349–3354, New York, NY, USA, 2010. Association for Computing Machinery.
- [Pfe18] Thies Pfeiffer. Gaze-Based Assistive Technologies. In *Smart Technologies: Breakthroughs in Research and Practice*, pp. 44–66. IGI Global, 2018.
- [PG16] Ken Pfeuffer and Hans Gellersen. Gaze and Touch Interaction on Tablets. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, UIST '16, pp. 301–311, New York, NY, USA, 2016. Association for Computing Machinery.
- [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. Association for Computing Machinery.
- [PPJ03] Salil Prabhakar, Sharath Pankanti, and Anil K Jain. Biometric recognition: Security and privacy concerns. *IEEE security & privacy*, Vol. 1, No. 2, pp. 33–42, 2003.
- [RAM⁺19] Sheikh Rivu, Yasmeen Abdrabou, Thomas Mayer, Ken Pfeuffer, and Florian Alt. GazeButton: Enhancing Buttons with Eye Gaze Interactions. In *Proceedings of the 11th ACM Symposium on Eye Tracking Research & Applications*, ETRA '19, pp. 1–7, New York, NY, USA, 2019. Association for Computing Machinery.
- [RLG09] Anne Roudaut, Eric Lecolinet, and Yves Guiard. MicroRolls: Expanding Touch-Screen Input Vocabulary by Distinguishing Rolls vs. Slides of the Thumb. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pp. 927–936, New York, NY, USA, 2009. Association for Computing Machinery.
- [RT09] Volker Roth and Thea Turner. Bezel Swipe: Conflict-Free Scrolling and Multiple Selection on Mobile Touch Screen Devices. In *Proceedings of the SIGCHI Conference*

on Human Factors in Computing Systems, CHI '09, pp. 1523–1526, New York, NY, USA, 2009. Association for Computing Machinery.

- [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. Association for Computing Machinery.
- [SFS19] Yusuke Sei, Minto Funakoshi, and Buntarou Shizuki. Expanding One-Handed Touch Input Vocabulary Using Index Finger on and Above Back-of-Device. In *Proceedings of the 31st Australian Conference on Human-Computer-Interaction*, OZCHI '19, pp. 585–589, New York, NY, USA, 2019. Association for Computing Machinery.
- [SRC05] Katie A. Siek, Yvonne Rogers, and Kay H. Connelly. Fat Finger Worries: How Older and Younger Users Physically Interact with PDAs. In Maria Francesca Costabile and Fabio Paternò, editors, *Human-Computer Interaction - INTERACT 2005*, pp. 267–280, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.
- [SWRX16] Cheng Song, Aosen Wang, Kui Ren, and Wenyao Xu. EyeVeri: A secure and usable approach for smartphone user authentication. In *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, pp. 1–9, 2016.
- [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 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pp. 4179–4188, New York, NY, USA, 2015. Association for Computing Machinery.
- [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 Symposium on Eye-Tracking Research & Applications*, ETRA '10, pp. 93–96, New York, NY, USA, 2010. Association for Computing Machinery.
- [VB12] Vytautas Vaitukaitis and Andreas Bulling. Eye Gesture Recognition on Portable Devices. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, UbiComp '12, pp. 711–714, New York, NY, USA, 2012. Association for Computing Machinery.
- [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. Association for Computing Machinery.

- [VHCR20] Simon Voelker, Sebastian Hueber, Christian Corsten, and Christian Remy. Head-Reach: Using Head Tracking to Increase Reachability on Mobile Touch Devices. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, pp. 1–12, New York, NY, USA, 2020. Association for Computing Machinery.
- [VHH⁺20] Simon Voelker, Sebastian Hueber, Christian Holz, Christian Remy, and Nicolai Marquardt. GazeConduits: Calibration-Free Cross-Device Collaboration through Gaze and Touch. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, pp. 1–10, New York, NY, USA, 2020. Association for Computing Machinery.
- [WCRI09] Feng Wang, Xiang Cao, Xiangshi Ren, and Pourang Irani. Detecting and Leveraging Finger Orientation for Interaction with Direct-Touch Surfaces. In *Proceedings of the 22nd Annual ACM Symposium on User Interface Software and Technology*, UIST '09, pp. 23–32, New York, NY, USA, 2009. Association for Computing Machinery.
- [WEB18] Anton Wachner, Janick Edinger, and Christian Becker. Towards Gaze-Based Mobile Device Interaction for the Disabled. In *2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pp. 397–402, 2018.
- [WG20] Bryan Wang and Tovi Grossman. BlyncSync: Enabling Multimodal Smartwatch Gestures with Synchronous Touch and Blink. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, pp. 1–14, New York, NY, USA, 2020. Association for Computing Machinery.
- [ZKM17] Xiaoyi Zhang, Harish Kulkarni, and Meredith Ringel Morris. Smartphone-Based Gaze Gesture Communication for People with Motor Disabilities. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, pp. 2878–2889, New York, NY, USA, 2017. Association for Computing Machinery.
- [池松 19] 池松香, 山中祥太, 坪内孝太, 椎尾一郎. Copernican-Touch: タッチサーフェスを指へ向けて動かし接触させるタッチ入力拡張技法. コンピュータソフトウェア, Vol. 36, No. 3, pp. 20–32, 2019.
- [八箇 19] 八箇恭平, 磯本俊弥, 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, 2019.