

2020年度

筑波大学情報学群情報科学類

卒業研究論文

題目

スマートフォンの操作予測に向けた視線分析

主専攻 ソフトウェアサイエンス主専攻

著者 村岡 沙紀

指導教員 高橋 伸 川口 一画 志築 文太郎

要 旨

本研究では、スマートフォン操作時の自然な視線の振る舞いから、操作を予測することを目指して、スマートフォンの操作と視線の関係を明らかにした。そのために、まずスマートフォンの正面カメラとiOSのフレームワークであるARKitを使用した視線推定を行った。次に、スマートフォンの操作に関するデータとスマートフォンに向けられる視線データを記録する2つのアプリケーション、ホーム画面アプリケーションとマップアプリケーションを実装し、データの記録を行った。記録したデータを観察したところ、視線の移動距離と移動方向には、操作と関連性があることがわかった。その後、視線データから操作の予測を行えるかを検証するために、視線データを入力として、操作の領域と操作の種類を分類する機械学習モデルをランダムフォレストを用いて作成し、精度を調査した。その結果、操作領域分類では、画面を4分割した際の正解率が0.75、6分割が0.67となり、操作分類では、タップ、ドラッグ、ピンチの3種類の分類の場合に正解率が0.56となった。

目次

第1章	序論	1
1.1	本論文の構成	2
第2章	関連研究	3
2.1	視線ジェスチャを利用したモバイルデバイスの操作方法	3
2.2	視線の分析	3
第3章	視線による操作予測に基づくユーザ支援	5
3.1	視線による操作予測	5
3.2	操作予測に基づくユーザ支援	5
第4章	実装	8
4.1	視線推定	8
4.1.1	視線推定方法	8
4.1.2	視線観察アプリケーション	10
4.2	予備実験	11
4.2.1	タスク	11
4.2.2	結果と分析	12
4.3	アプリケーション	16
4.3.1	視線及び操作データの収集用アプリケーション	16
4.3.2	収集したデータの可視化用アプリケーション	18
第5章	分析	20
5.1	操作中における視線の記録	20
5.1.1	タスク	20
5.1.2	記録データ	21
5.2	キャリブレーション	22
5.3	データ観察	23
5.4	分類	25
5.4.1	データの前処理	26
	ラベル付け	27
	セグメンテーション	28
5.4.2	機械学習	30
5.4.3	分類結果	30

操作領域分類	30
操作分類	38
第 6 章 考察	42
6.1 用いた視線の振る舞い	42
6.2 操作領域分類	42
6.3 操作分類	42
6.4 ラウンド間の正解率	43
6.5 今後の展望	43
第 7 章 結論	45
謝辞	46
参考文献	47

目次

3.1 ユーザ支援例 1	6
3.2 ユーザ支援例 2	7
4.1 eyePoint の座標系と代表的な座標 (単位は pt)	9
4.2 視線観察アプリケーション	10
4.3 実験アプリケーションの例	12
4.4 瞬き発生時の eyePoint の変化	13
4.5 eyeBlink の指数加重移動平均	14
4.6 瞬きによるズレの補正	15
4.7 データ収集用アプリケーションの例	17
4.8 キャリブレーションのための赤い円の表示例	18
4.9 データ再生アプリケーション	19
5.1 アラートの表示例	21
5.2 一定のズレの修正	23
5.3 operationPoint と eyePoint の移動距離と移動方向のグラフ	25
5.4 片手の指の届く範囲	26
5.5 SEGMENTSIZ E と STEPSIZ E の例	27
5.6 画面の領域に対応するラベルの例	28
5.7 操作に対するラベル付けの例 (SPLITSIZ E=4)	28
5.8 データ例	29
5.9 SEGMENTSIZ E を 1 から 100 とした場合の正解率 (操作領域分類, SPLITSIZ E=4)	31
5.10 操作領域分類の正解率 (SEGMENTSIZ E=75, STEPSIZ E=1, SPLITSIZ E=4)	32
5.11 STEPSIZ E を 1 から 100 とした場合の正解率 (操作領域分類, SPLITSIZ E=4)	33
5.12 操作領域分類の正解率 (SEGMENTSIZ E=75, STEPSIZ E=4, SPLITSIZ E=4)	34
5.13 SPLITSIZ E=6 の場合の正解率 (操作領域分類, SEGMENTSIZ E=75, STEPSIZ E=1)	36
5.14 SPLITSIZ E=9 の場合の正解率 (操作領域分類, SEGMENTSIZ E=75, STEPSIZ E=1)	37
5.15 SEGMENTSIZ E を 1 から 100 とした場合の正解率 (操作分類)	38
5.16 操作分類の正解率 (SEGMENTSIZ E=27, STEPSIZ E=1)	39
5.17 STEPSIZ E を 1 から 100 とした場合の正解率 (操作分類, SEGMENTSIZ E=27)	40
5.18 操作分類の正解率 (ラベル数:6, SEGMENTSIZ E=27, STEPSIZ E=1)	41

第1章 序論

現在、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、片手によって把持可能な情報端末が広く普及している。その操作は多くの場合、手の指を用いたタッチスクリーンへのタッチによって行われる。しかし、スマートフォンのタッチ操作には問題がある。例えば、操作対象が指で隠れてしまうファットフィンガー問題 [1, 2] や、画面の拡大により片手の指で全体を操作することが難しいという指の到達性の問題 [3, 4] がある。そこで、インタフェースの表示を変更したり、カーソルを使用することによって、これらの問題に対応する操作方法も提案されている [5, 6, 7]。しかし、これらは通常の操作から提案された操作への切り替えのために、事前に決められた角度以上に端末を傾けること [6] や端末のベゼルからのスワイプ [5]、ユーザが押下圧を高めること [7]、などの普段用いないジェスチャをトリガとして行う必要がある。

また追加のモダリティとして、視線を用いてスマートフォンを操作したり、操作支援をすることが研究されている [8]。これは、商用スマートフォンに搭載されたカメラの機能が向上し、追加の機器なしで視線追跡できる場合が増えたことが一つの要因である。視線を用いることで、指で画面が隠れることがなく、指が届かないところの操作が容易になったり、手が塞がっている時の操作が可能になるという利点がある。視線を用いたコンピュータの操作や操作支援には、操作を意図した注視、あるいは瞬きを含む視線の動き（以降、視線ジェスチャ）を用いた操作のために視線を意図的に移動させる方法と、通常のスマートフォン操作時の自然な視線の振る舞いを用いる方法がある。本研究では、自然な視線の振る舞いを利用してスマートフォンの操作予測ができないかと考えた。自然な視線の振る舞いを利用するため、ユーザはスマートフォンの通常の操作だけで、これまでのスマートフォンにおける操作の問題を解消することができると考えられる。

自然な視線の振る舞いを利用した研究はいくつかある。その多くがPC操作時やビデオフレームを利用した研究である。これらにおける視線の振る舞いは、スマートフォンとは異なる可能性がある。またモバイルデバイスにおける自然な視線の振る舞いを利用した研究では、操作には着目した研究はない。そこで、スマートフォン操作時の視線の振る舞いを分析し、視線と操作の関係を明らかにすることで、自然な視線の振る舞いを利用したスマートフォン操作予測に貢献できると考えられる。

本研究の目的は、スマートフォンの操作と視線の関係を明らかにすることである。それによって、スマートフォン操作時の自然な視線の振る舞いから、操作を予測することが可能になると考えられる。そのために、スマートフォンの正面カメラを使用して視線推定を行い、スマートフォンの操作に関するデータとスマートフォンに向けられる視線のデータを記録し、分析を行う。

スマートフォン操作時の自然な視線の振る舞いから、操作を予測できれば、操作モードの自動的な切り替えや操作のレコメンドが可能になる。例えば、ユーザが指の届かない範囲を操作しようとすることを検出し、自動でカーソルを表示することができる。これによって、ユーザはトリガなしで新たな操作方法を実行することができ、操作ごとに対応するトリガを用意する必要がなく、その数による

制約がなくなるためより多くの種類の操作を行うことも可能になる。また、ユーザが次に実行しようとしている操作の種類を特定し、その操作のレコメンドを表示することができる。レコメンドの表示によって、スマートフォン初心者の操作を助けることが可能になる。

本研究では、記録したデータの観察から、視線の移動距離と移動方向には、操作と関連性があることがわかった。その後の視線データによって操作が分類できるかどうかを検証したところ、操作の領域の分類では、画面を4分割した際の正解率が0.75、6分割が0.67、9分割が0.49となった。操作の種類分類では、タップ、ドラッグ、ピンチの3種類で正解率が0.56となり、さらにドラッグを上下左右に分けた5種類で正解率が0.27となった。

1.1 本論文の構成

本章において、本論文の背景および研究目的とアプローチを述べた。第2章では、関連研究に対する本論文の位置付けを述べる。第3章では、スマートフォンの正面カメラを用いた視線推定の方法と、推定した視線座標を観察するためのアプリケーションの実装、推定した視線座標の精度を確認するための予備実験について述べる。第4章では、分析のための操作に関するデータと視線データを記録するための2つのアプリケーションと、記録したデータを観察するためのアプリケーションの実装について述べる。第5章では、記録したデータの観察・分析と、視線データによる操作の分類について述べる。第5章では、本研究にて行った視線データの観察・分析についての考察を示す。そして第7章にて、本研究のまとめを述べる。

第2章 関連研究

本研究では、スマートフォンの操作と自然な視線の振る舞いの関係を明らかにするために、スマートフォン操作時の視線データを分析する。そのため本研究に関連する研究として、「視線ジェスチャを利用したモバイルデバイスの操作方法」及び「視線の分析」がある。本章では、これらの研究に対する本研究の位置付けを示す。

2.1 視線ジェスチャを利用したモバイルデバイスの操作方法

モバイルデバイスにおいて視線を利用した操作方法は多数提案されており、主に視線を意図的に移動させる視線ジェスチャを用いている。Mortenら [9] は、眼球運動のみを使用してスマートフォンのインタラクションが可能であるかを調査した。彼らの結果から注視時間選択と視線ジェスチャ選択が、エラー率と完了時間でチルト操作に匹敵し、視線ジェスチャが注視時間選択よりもエラー率が低く、小さなターゲットにおいては完了時間も早いということが明らかになった。Stellmachら [10] は、チルトまたはタッチと視線でのパンを組み合わせた、視線位置でズームを行う手法である視線検出ピポットズームを提案した。彼らの実験ではユーザのフィードバックにおいてタッチジェスチャと組み合わせたピポットズームが高い可能性を示した。Miluzzoら [11] は、モバイル端末 Nokia N810 における EyePhone のプロトタイプ実装を提供した。これは、アプリケーションを起動することを目的として、ディスプレイの目の位置を追跡し、瞬きによって視線が向けられた位置のアプリケーションを起動した。Valitukaitisら [12] 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。Drewesら [13] は、モバイルデバイスを目で制御するための2つの手法、滞留時間ベースの手法と視線ジェスチャを比較した。彼らは視線ジェスチャが、滞留時間ベースの手法と比べて、正確なキャリブレーションなしで使用できることや実行できるジェスチャ数が非常に多いという利点を示した。これらは、スマートフォンの操作のために意図的に視線を移動させたり、瞬きを行う必要がある。本研究では、自然な視線の振る舞いに着目するという点でこれらの研究とは異なる。

2.2 視線の分析

視線の振る舞いを分析した研究の多くが、PC操作時の視線や画像・動画を見るとききの視線についての分析を行った研究である。まずPC操作時の視線の研究について説明する。Wiedbushら [14] は、マルチメディア学習環境におけるメタ認知判断をモデリングした。彼らの結果は、視線の振る舞いが実際のコンテンツの関連性ではなく、知覚されたコンテンツの関連性に基づいて異なることを示した。Blascheckら [15] は、人々がどのように、自然言語テキストとソースコードを読むのかを調査するため

に、眼球運動データを分析した。彼らの結果から、自然言語テキストはより直線的に読み取られ、ソースコードは重要な部分により集中することがわかった。Papoutsaki ら [16] は、視線とタイピングの関係を調査し、タッチタイピストと非タッチタイピストの違いに焦点を当てた。これらは、テキストキャレットと注視位置の距離を計算することで、92%の精度でタッチタイピストと非タッチタイピストを分類することに成功した。Huang ら [17] は、マウスクリックイベントを取り巻く注視行動のパターンを分析し、人間のストレスレベルを推定した。彼らの結果は、非侵襲的に収集されたデータを通じてユーザのストレスを検出することが可能であるだけでなく、提案された機能がさまざまなユーザー間で一般化可能であることも示した。Salja ら [18] は、学習障害のために読解力が異なる子供の凝視を分類した。これらは、学習障害のある子供の凝視時間が有意に長いことを発見した。Bednarik ら [19] は、視線を利用した、インタラクションの意図のタスクに依存しない予測のためのオフラインフレームワークを開発した。彼らの結果から、凝視とサッケードの位置と持続時間から計算された特徴量から、動画の視聴時にタスクが与えられているか、そうでないかを識別可能であることが示された。次に画像・動画を見るとき視線の研究について説明する。Gridinger ら [20] は、動画視聴時の視線を分析した。これらは、自由に動画を視聴した場合と、タスクを与えられた場合の視聴のスキャンパスに有意差があることを発見した。Castner ら [21] は、歯科 X 線写真を解釈する際の視線データを分析した。彼らの手法は、93%の精度で歯科の専門家と初心者とを識別することができ、これによってリアルタイムで学生の学習の進捗状況の評価の可能性を示した。

これらの研究は、スマートフォンよりも大きな画面を持つ PC を操作する際や画像・動画を見る際の視線を分析しており、スマートフォンを操作する際の視線の振る舞いとは異なる可能性がある。

本研究で着目する、モバイルデバイスにおける自然な視線の振る舞いの分析には、ユーザの関心や画面内のターゲットとの視線の関係についての分析がある。Mueller ら [22] は、10×9 mm 間隔で 30 箇所連続して出現するターゲットを凝視する時の視線データを収集し、ターゲット位置を視線から読み取ることができるかどうかを調査した。30 箇所の分類器の精度は、被験者間で広範囲に渡っていた。Kar ら [23] は、視線と視線ジェスチャ追跡から学習者の注意を推定するためのプラットフォームである Gestatten を提案した。彼らの結果では、平均主観スコアがアプリケーションで生成されたスコアと一致するため、注意深い学習者と注意力のない学習者の両方を識別できることが明らかになった。Lagun ら [24] は、携帯電話におけるブラウザのビューポート（Web ページの表示部分）を追跡することで、ユーザーの注意を正確に測定でき、クリックがない場合の検索満足度を適切に測定できるかどうかを調査した。彼らの結果から、注視時間とビューポート時間の間に強い相関関係があり、平均的なユーザーの注意は電話画面の上半分に集中していることが明らかになった。Jalaliniya ら [25] は、視界で水平または垂直に動くオブジェクトから、ユーザの関心のあるオブジェクトを自動検出する、眼球運動分析技術 EyeGrip を提案した。EyeGrip はスクロールタスク中に、ユーザにとって魅力的な画像を 87%の精度で検出した。Giannopoulos ら [26] は、小型ディスプレイにおけるマップの探索に視線の履歴を表示する構想 GeoGazeMarks を提案した。GeoGazeMarks は、標準のパン・ズームと比較してマップ探索タスクの効率と有効性が有意に上昇した。これらは、スマートフォンに向けられた視線を分析した研究であるが、操作との関係性は明らかになっていない。また、視線追跡には追加の装置や、スマートフォンや身体の固定を必要としており、実際の操作時の視線の振る舞いとは異なる可能性がある。本研究では、スマートフォンの正面カメラを利用することで、追加の装置なしで視線を推定し、スマートフォン操作と自然な視線の振る舞いの関係を明らかにする。

第3章 視線による操作予測に基づくユーザ支援

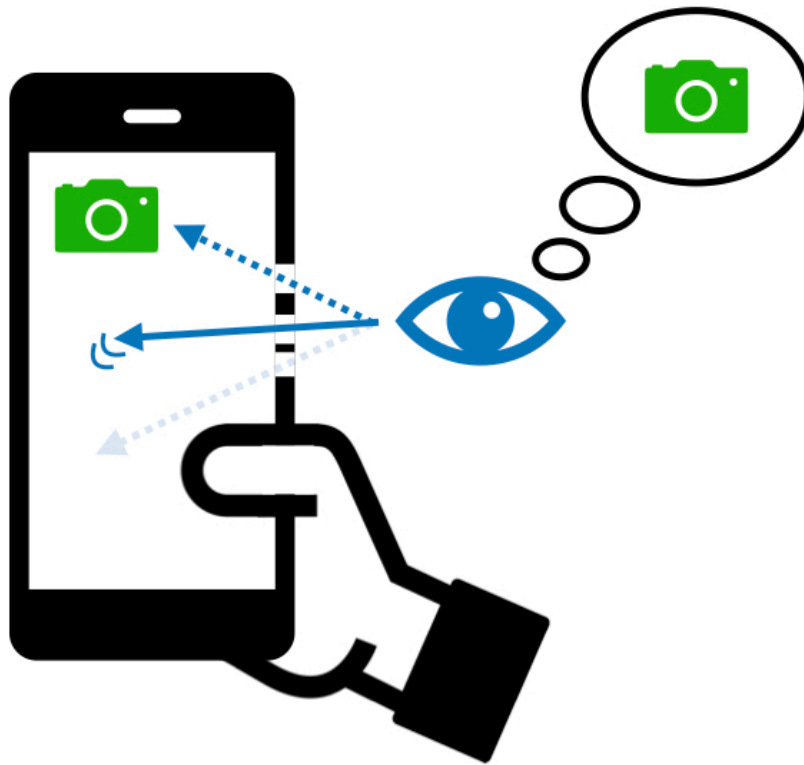
3.1 視線による操作予測

本研究では、新たなジェスチャを必要としないスマートフォン操作の支援のために、自然な視線の振る舞いに着目した。スマートフォンはその大部分をタッチスクリーンが占めており、操作時には操作対象や行われる操作の触覚フィードバックを得にくい。そのため、ユーザは視覚的なフィードバックに頼り、正確な操作を行うためには操作対象に視線を向ける必要がある。また、スマートフォンに向けられる視線は、操作位置や操作の種類によって異なると考えられる。例えば、ホーム画面からアプリを起動させる場合、ユーザはホーム画面からそのアプリを探し、見つけたのちにアプリアイコンをタップする。つまりユーザは操作したい対象を見て、それから操作を行う。このことから、その際の視線位置を用いることで、操作対象の位置を特定することが可能であると考えられる。また、マップアプリなどをスクロールする場合、ユーザはスクロールするために指でタッチする位置ではなくスクロールして見たい方向に視線を向ける。この際の視線の振る舞いはタップとは異なると考えられ、操作の種類の手予測も可能ではないかと考えられる。

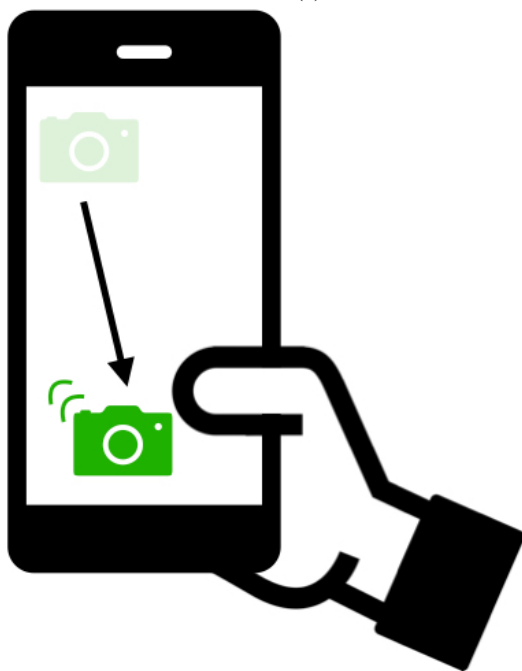
日常のスマートフォン操作では、様々な操作が連続して行われる。その際に視線は実際に操作を行う瞬間には次の操作のために移動している。そのため、操作中の視線ではなく操作前の視線による操作予測ができる可能性がある。このことは、5章におけるデータの観察で確かめる。

3.2 操作予測に基づくユーザ支援

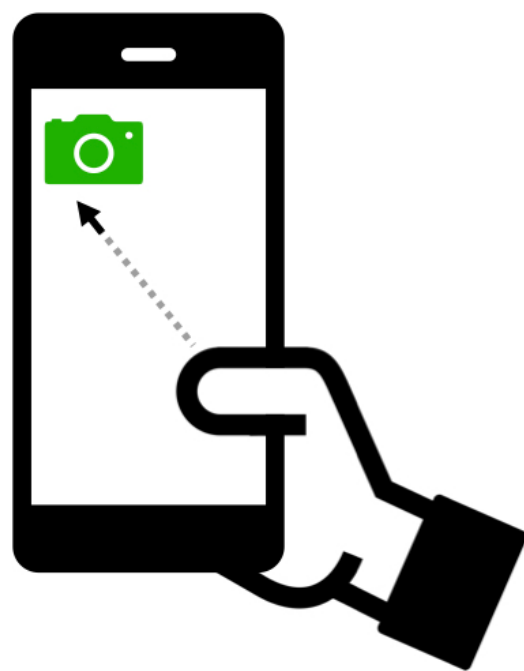
視線による操作予測が可能になれば、様々なユーザ支援が可能になると考えられる。例えば、様々な理由で片手が塞がりスマートフォンを片手のみで操作しなければならない場合は多い。しかし、片手の親指が届く範囲は限られるため画面全体を操作することは難しい。ここでユーザがスマートフォンの指の届かない範囲を操作しようとするを予測することで、2章で述べた片手操作手法が自動的に起動し、カーソルが表示されたり、画面を小さくしたりすることができる。他にもUI要素を移動する、または自動で実行することがなどが可能になる(図3.1)。また、スマートフォンの画面はサイズが限られているため、抽象化されたイラストで表現されたボタンが数多く存在する。初心者のユーザはそれらのボタンの機能を理解できない可能性がある。そこで、ユーザがあるボタンを操作しようとするを検出し、そのボタンの機能の説明を表示させることで、ユーザの円滑な操作を助けることができる(図3.2)。



(a) 指の届かない範囲を操作しようとしている場合

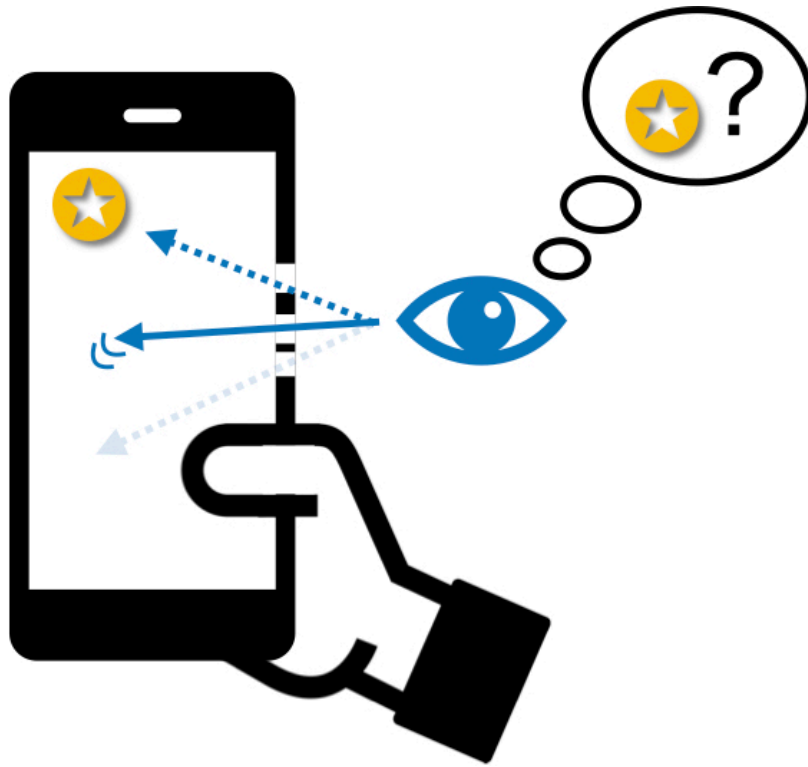


(b) UI 要素が指の届く範囲に移動してくる



(c) ターゲット選択のためのカーソルが表示される

図 3.1: ユーザ支援例 1



(a) ユーザがあるボタンに対して、どのような機能なのか分からない場合



(b) ボタンの説明が表示される

図 3.2: ユーザ支援例 2

第4章 実装

本研究では、スマートフォン操作時の視線データの記録・観察のためにアプリケーションを実装した。本章では、まず視線データ記録のためのスマートフォンの正面カメラを用いた視線推定方法について述べ、その精度を確認するための予備実験について述べる。次にその視線推定方法を用いて視線と操作のデータを収集するアプリケーション、収集したデータを観察するアプリケーションについて述べる。

4.1 視線推定

本研究では、スマートフォン操作時の視線データを記録するために、スマートフォンの正面カメラを用いてユーザの視点を推定した。視線推定には iOS における拡張現実を構築するためのフレームワークである ARKit¹を使用し、スマートフォンは、iPhone11²を用いた。本節では、ARKit を用いた視線推定方法を説明し、推定した視線位置を観察するためのアプリケーション、精度を確認するための予備実験について説明する。

4.1.1 視線推定方法

本研究では、ARKit を用いることにより、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。ARKit は、カメラを起動した際の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³。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 ARAnchor⁴が設置され、撮影範囲に存在する限り位置および姿勢の推定が可能となる。特に、人の顔には ARFaceAnchor⁵が設置され、拡張現実空間における頭の位置および向き、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。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、ユーザの目の位置と向きが必要である。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ため、目の位置および向きを利用するために、拡張現実空

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

²<https://www.apple.com/jp/iphone-11/specs/>

³https://developer.apple.com/documentation/arkit/world_tracking

⁴<https://developer.apple.com/documentation/arkit/anchor>

⁵<https://developer.apple.com/documentation/arkit/faceanchor>

間における座標に変換する。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。ここまでの操作により、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および向きと、ユーザの左右の目のそれぞれの視線ベクトルの始点および向きが算出される。多くの場合、左右の目のそれぞれ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。そのため本手法では、左右の目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。なお視点座標は、最後に、拡張現実空間における座標から、スクリーンの左上を原点とする座標に変換する。この座標系を図 4.1 に示す。

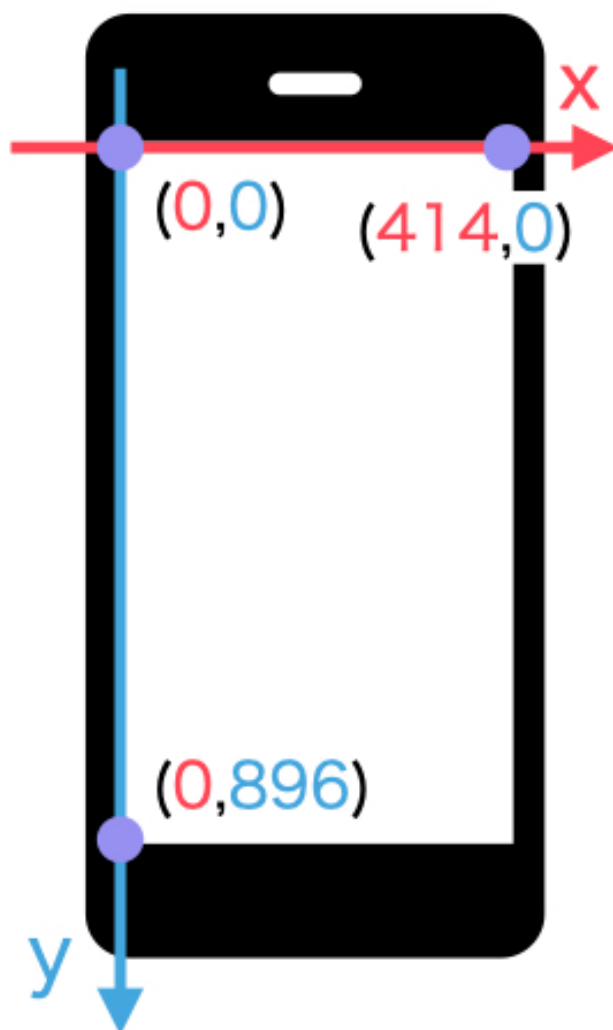


図 4.1: eyePoint の座標系と代表的な座標（単位は pt）

以下では、ここで推定した視点座標を eyePoint、その x 座標、y 座標をそれぞれ eyePoint-x、eyePoint-y とする。

4.1.2 視線観察アプリケーション

4.1.1 節で推定した視線の振る舞いを観察するために、スマートフォンに向けられた視線をリアルタイムで視覚化する視線観察アプリケーションを実装した。

視線観察アプリケーションは、スマートフォンを見ている際の、eyePoint を灰色の円で表示するアプリケーションである (図 4.2)。赤い円は、目で見えるターゲットとして実際に見ている位置と eyePoint とを比べるために用いた。このアプリケーションを用いて、推定した視線座標を観察したところ、視線座標には恒常的なズレと、瞬き時の瞬間的な上下のズレがあることが確認できた。

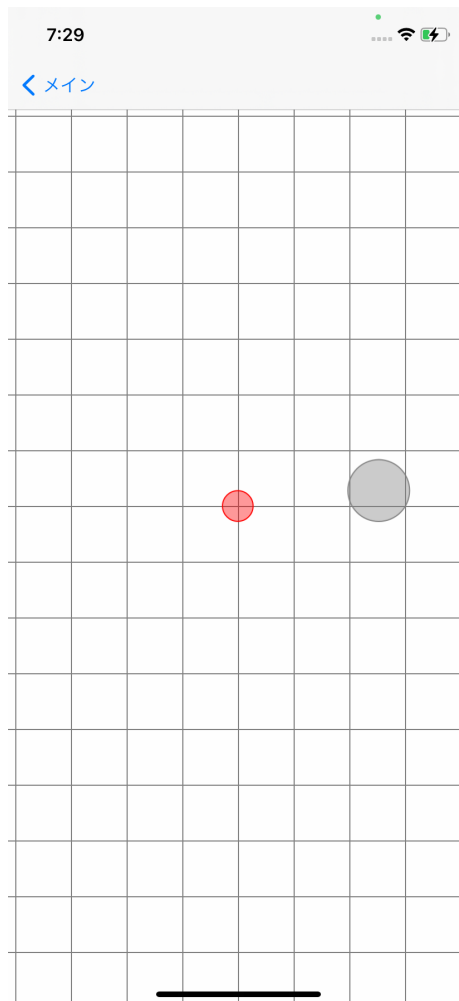


図 4.2: 視線観察アプリケーション

4.2 予備実験

4.1.2 節で視線の振る舞いを観察した結果、推定した視線位置と実際の視線位置には、恒常的なズレと瞬間的なズレがあることがわかった。瞬間的なズレは、瞬きに基づいており、恒常的なズレと同時に発生する。そこで推定した視線位置の精度を評価するために予備実験を行った。参加者は、著者1名で椅子に座った状態でスマートフォンを操作した。参加者は、普段スマートフォンを操作する方の手でスマートフォンを操作し、もう片方の手でスマートフォンを把持した。また、机に肘を付き、できるだけ安定した状態でスマートフォンを操作するように求められた。

4.2.1 タスク

本実験では、画面内に凝視する点が、灰色のボタンで表示されるアプリケーション（図 4.3a）を利用した。ボタンは押すと緑色に変わり、参加者はボタンを押してからボタンが消えるまで、ボタンを2秒間凝視するように求められた。その後、次のボタンが表示され、これを1分間繰り返し行う。ボタンのサイズは、Apple のヒューマンインタフェースガイドライン⁶にそって、60pt × 60pt とし、ボタン中央には注視点として+マークが表示される。ボタンが表示されるのは、画面内の9点で位置はランダムに表示される。表示位置を図 4.3b に示す。ボタンを凝視している間、ボタンの中心の座標 `targetPoint`（座標系は図 4.1 と同様）と `eyePoint` が記録される。このタスクを1回行うことを1ラウンドとし、参加者は3ラウンドを完了した。

⁶<https://developer.apple.com/design/human-interface-guidelines/ios/icons-and-images/app-icon/>

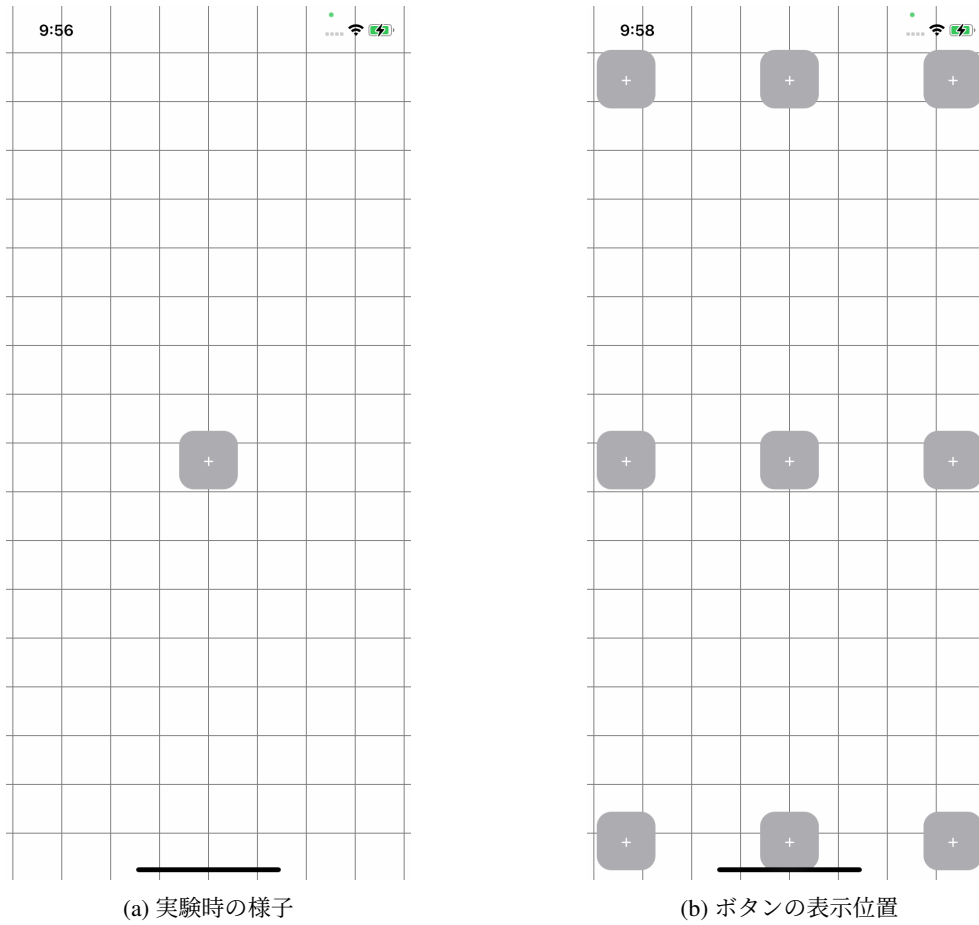
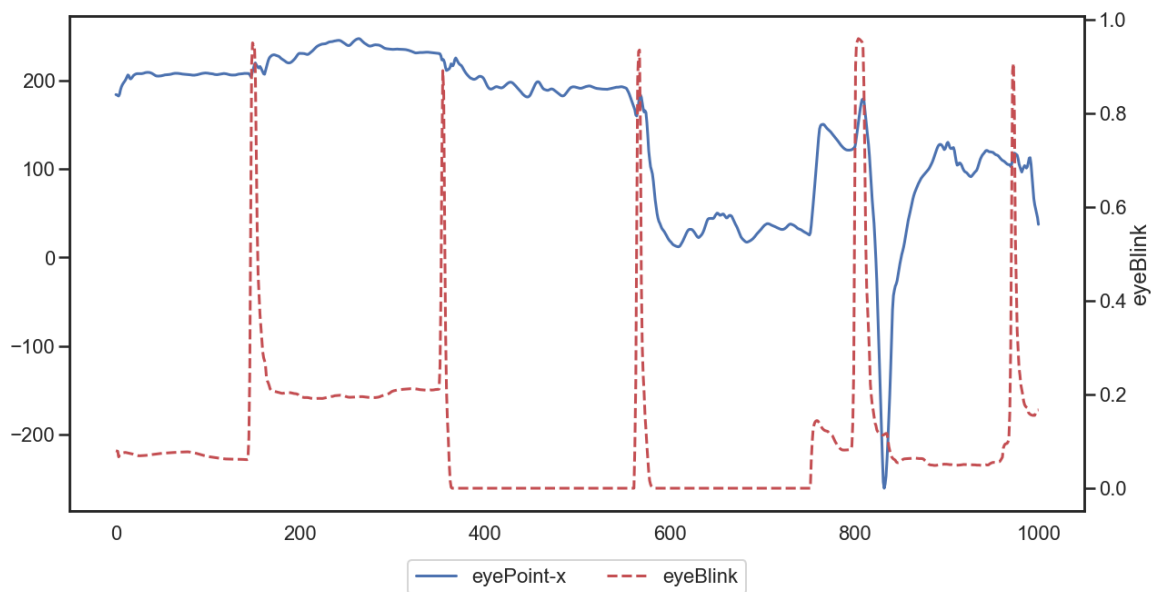


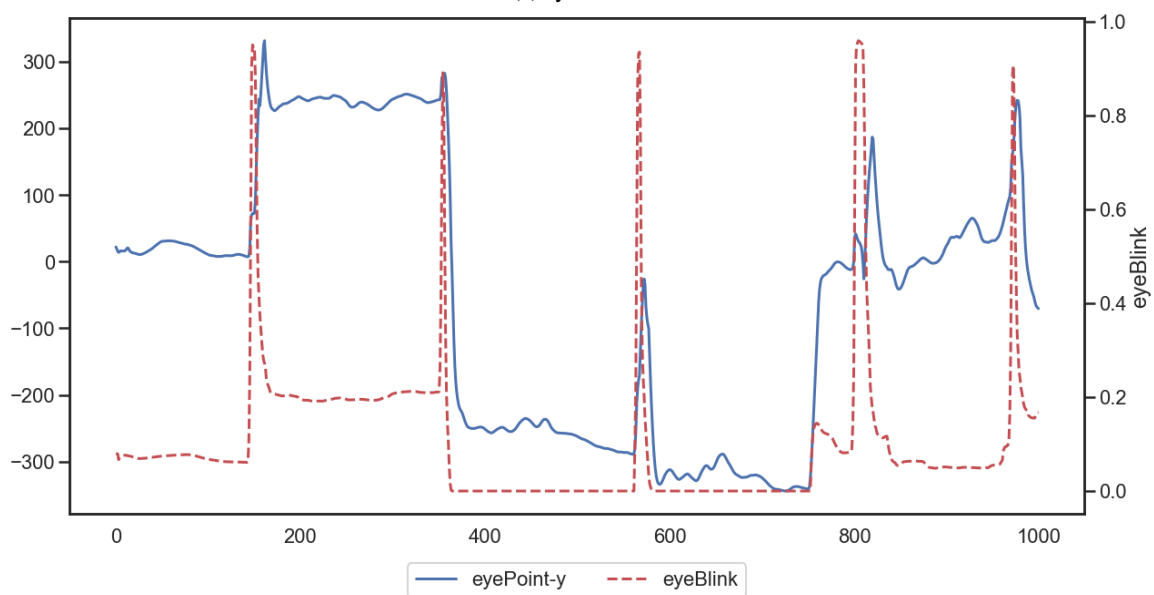
図 4.3: 実験アプリケーションの例

4.2.2 結果と分析

初めに、瞬きによる瞬間的なズレについての結果を示す。本研究では、ARKit を用いて、左右の目の閉じ具合の値 `leftEyeBlink` と `rightEyeBlink` も記録した。この値が大きいほど目を閉じていることを表し、瞬きが発生していると考えられる。そこで `leftEyeBlink` と `rightEyeBlink` の平均 `eyeBlink` を求め、この値と `eyePoint` の x 軸、y 軸の値をそれぞれグラフにした。これを図 4.4a と図 4.4b に示す。図は、青線が `eyePoint-x` または `eyePoint-y` を示し、赤の点線が `eyeBlink` を示している。`eyePoint` は、`eyePoint-x` が大きくなるほど画面右に、`eyePoint-y` が大きくなるほど下に移動したことを示す。図には、`eyeBlink` の値が大きく上昇する点が 5 箇所あり、`eyePoint-x` には変化が見られないが、`eyePoint-y` の値は大きく上昇することがわかった。つまり、瞬き時には `eyePoint` は実際の視線位置よりも画面下部方向にズレていることがわかった。



(a) eyePoint-x



(b) eyePoint-y の値

図 4.4: 瞬き発生時の eyePoint の変化 (青線が eyePoint, 赤の点線が eyeBlink, 横軸がフレーム, 縦軸はそれぞれの値を示す)

そこで、瞬きによるズレを除去するために、瞬き時の eyePoint-y の補正を行った。瞬き検出に用いる eyeBlink の閾値を決めるために、eyeBlink の外れ値検出を行った。外れ値検出には、指数加重移動平均を用い、その値と実際の eyeBlink の値の差が指数加重移動標準偏差の 4 倍よりも大きい場合に

外れ値として検出した（指数加重移動平均のフレーム数は、100）。この検出結果を図 4.5 に示す。図から eyeBlink が大きく上昇した点が検出できていることがわかる。ここで外れ値として検出された、eyeBlink の平均 0.71 を閾値 th_{blink} とする。eyeBlink が th_{blink} を連続して超える場合に瞬きをしていると判定した。瞬き時の視線位置のズレを観察したところ、eyeBlink が大きく上昇しているタイミングと eyePoint-y がズレているタイミングが一致していないことがわかった（図 4.6）。またズレているフレーム数が、瞬きごとに異なることもわかった。そこで、eyeBlink によって検出された瞬き時の前後 50 フレームのなかで、eyePoint-y の値が最も大きいフレームを検出し、そのフレームを修正フレームの中心 $f_{blinkCenter}$ とした。eyeBlink が th_{blink} を超えているフレーム数 BLINKSIZE を計算し、 $f_{blinkCenter}$ の前後、BLINKSIZE \times 2 フレーム分を修正対象とした。そして、修正対象の 1 フレーム前の座標と 1 フレーム後の座標が、一次式で繋がるように修正した（Algorithm4.1）。瞬きによる修正を行った結果を図 4.6 に示す。

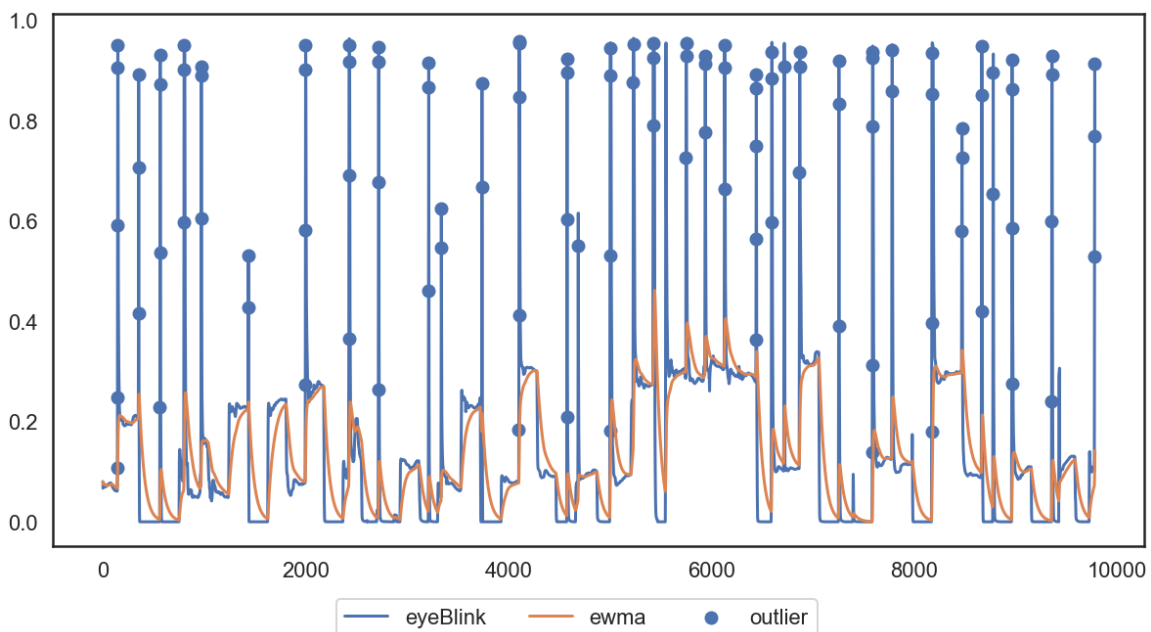


図 4.5: eyeBlink の指数加重移動平均（青線が eyeBlink, オレンジ線が eyeBlink の指数加重移動平均, 青点が外れ値を示す）

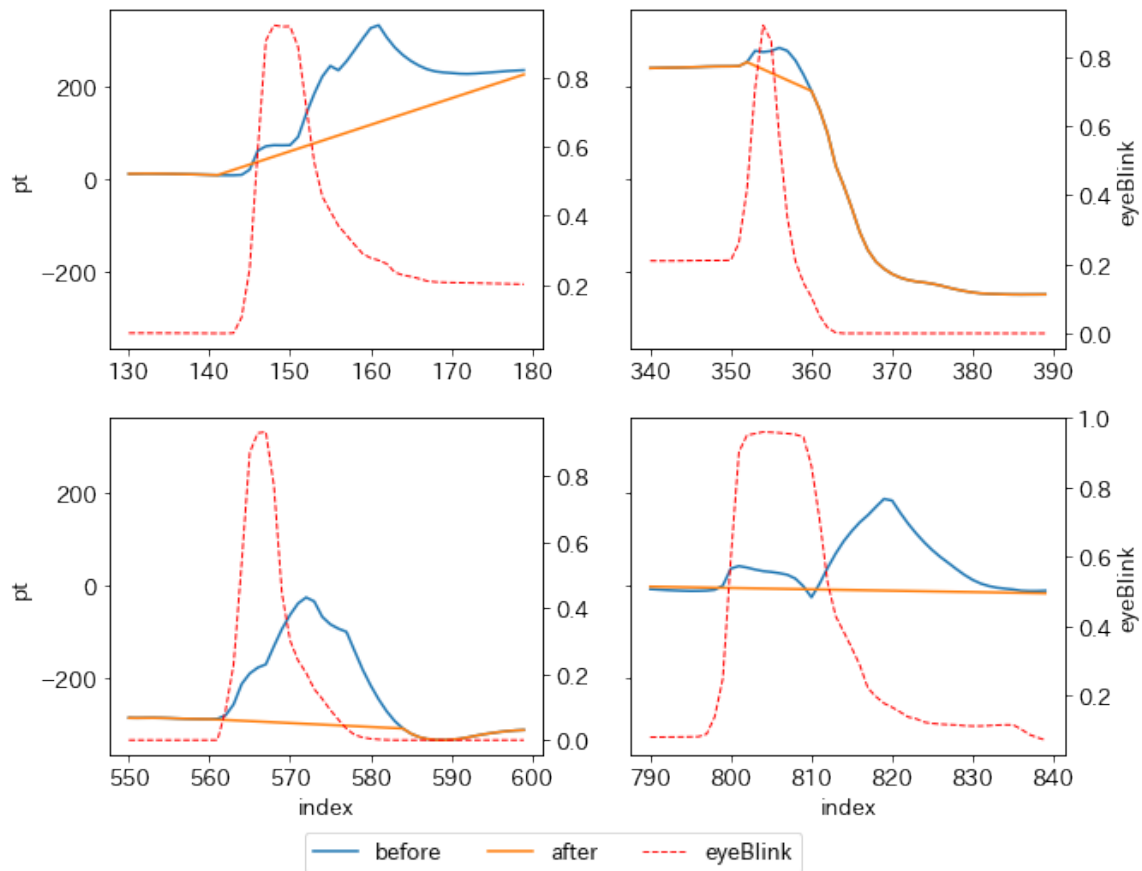


図 4.6: 瞬きによるズレの補正（青線が修正前，オレンジ線が修正後の eyePoint-y，赤い破線が eyeBlink を示す）

Algorithm 4.1 瞬きによるズレの補正

Require: $data$: データフレーム, $blinkStart$: 修正開始のインデックス, $blinkEnd$: 修正終了のインデックス

$d \leftarrow blinkStart - blinkEnd$

$a \leftarrow data[blinkStart]$ と $data[blinkEnd]$ の $eyePoint$ の差

$b \leftarrow data[blinkEnd]$ の $eyePoint * blinkStart$ と $data[blinkStart]$ の $eyePoint * blinkEnd$ の差

for $i \leftarrow blinkStart$ から $blinkEnd$ **do**

$data[data.index == i] \leftarrow (a * i + b) / d$

end for

続いて，Algorithm4.1で瞬きによる誤差を修正したデータを用いて，データ全体の $eyePoint$ と $targetPoint$ の差異を計算した．データごとの結果を表 4.1 に示す．平均値をみると x 軸，y 軸ともにズレが発生しており，特にデータ 3 の y 軸は最小値が 131 pt，最大値 1059 pt であることから，画面下部の

方に大きくズレが発生していることがわかる。このことから、推定した視線位置には一定のズレが発生しているため、分析の前にはキャリブレーションが必要となることがわかった。また、各ラウンドでズレも異なるため、ラウンドごとにキャリブレーションを行う。

表 4.1: targetPoint と eyePointAtPoint の差異 (単位は pt)

ラウンド番号	軸の方向	平均値	最小値	中央値	最大値
ラウンド 1	x 軸	69.14	-93.0	53.0	204
	y 軸	461.8	216.0	456.0	658
ラウンド 2	x 軸	59.23	-83.0	62.0	190.0
	y 軸	484.5	270.0	463.0	696.0
ラウンド 3	x 軸	10.65	-119.0	14.0	196
	y 軸	418.0	131.0	424.0	1059

これらを踏まえて、本研究では分析の前に瞬きによる瞬間的なズレの補正と恒常的なズレのキャリブレーションを行うこととした。

4.3 アプリケーション

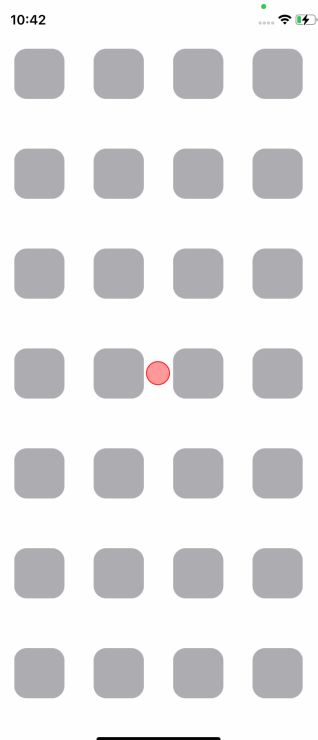
本研究では、2つのスマートフォン操作時の視線データを記録するアプリケーションと、収集したデータを視覚化するアプリケーションを実装した。

4.3.1 視線及び操作データの収集用アプリケーション

スマートフォン操作時の視線データを記録するために、日常のスマートフォンの操作を想定した2つのアプリケーションであるホーム画面アプリケーションとマップアプリケーションを実装した。

ホーム画面アプリケーションは、アプリケーションアイコンを想定したボタンを、iPhone11のホーム画面に倣って、4×7個配置したアプリケーションである(図4.7a)。ボタンのサイズは、4.2.1節と同様に、60pt×60ptとした。マップアプリケーションは、日常でよく使用されるアプリケーションであり、様々なタッチジェスチャ(タップ、スワイプ、ドラッグ)を行うアプリケーションであるので採用した。このアプリケーションは基本的なマップアプリケーションと同様に、マップの移動や拡大縮小、検索、マップタイプの変更、ピンの配置削除などの機能を実装した。

4.2.2節からデータを分析する前に、キャリブレーションが必要であることがわかった。そのため両方のアプリケーションにおいて、データの記録を開始すると恒常的なズレのキャリブレーションのための赤い円が中央に現れる。



(a) ホーム画面アプリケーション



(b) マップアプリケーション

図 4.8: キャリブレーションのための赤い円の表示例

4.3.2 収集したデータの可視化用アプリケーション

記録したデータを観察するためにアニメーションにして視覚化するデータ再生アプリケーションを実装した。

データ再生アプリケーションは、アプリケーションを用いて記録した推定した視線位置と操作のデータを視覚化するアプリケーションである (図 4.9)。灰色の円で視線位置を、赤い円でタッチジェスチャによる操作の座標を表示する。赤い円の下にはどんな操作を行ったかのラベルも一緒に表示される。灰色の円は、推定した視線位置の軌跡を 100 フレーム分表示し、観察しやすくするために、恒常的なズレをキャリブレーションした後の視線位置を表示する。また、データの観察のために再生・一時停止、データの早送り・戻しの機能を実装した。再生できるデータは、4.1.2 節の視線観察アプリケーション、ホーム画面アプリケーション、マップアプリケーションで記録したデータである。

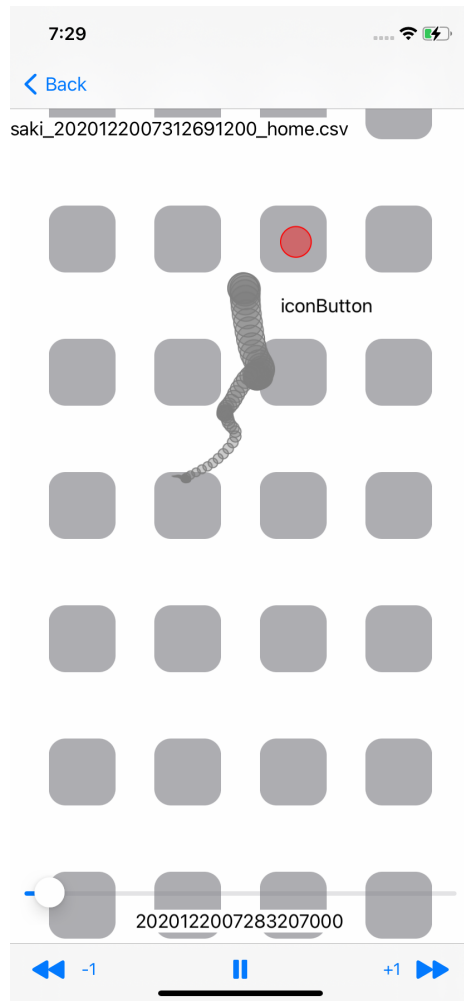


図 4.9: データ再生アプリケーション

第5章 分析

本研究では、スマートフォン操作と自然な視線の振る舞いの関係を明らかにすることを試みる。そのためにまずアプリケーション操作時のデータを記録し、推定した視線座標のキャリブレーションを行った。次に視線の振る舞いの特徴を見つけるためにデータの観察をした。その後、視線データから操作を分類する機械学習モデルを作成し、精度を調査した。

5.1 操作中における視線の記録

スマートフォン操作時の視線データと操作データを収集するために、4.3.1節で実装したホーム画面アプリケーションとマップアプリケーションを使用してスマートフォン操作時のデータを記録した。参加者は、著者1名で予備実験と同様の状態で行った。

5.1.1 タスク

ホーム画面アプリケーションとマップアプリケーションの両方で参加者は初めに、キャリブレーションのために画面中央に表示される赤い円のターゲットを見るように指示された(図4.8)。ターゲットは5秒間表示された。その後、それぞれのアプリケーションを3分間操作した。

ホーム画面アプリケーションでは、ランダムに1~10の数字がアイコンボタンに表示され、順番にボタンを押していくように指示された。最後のボタンが押されると0.5秒の遅延の後に新しく数字が表示される。マップアプリケーションでは、アプリケーションを自由に使って、自分の思い入れのある場所にピンを立てるように指示された。また両方のアプリケーションにおいて、視線の追跡が行えていない場合は、アラートが表示され、参加者はできるだけアラートが表示されないようにスマートフォンの角度やスマートフォンと顔の距離を調節するように求められた(図5.1)。両方のアプリケーションを操作するタスクを1ラウンドとし、参加者は6ラウンドを完了した。



(a) ホーム画面アプリケーション



(b) マップアプリケーション

図 5.1: アラートの表示例

5.1.2 記録データ

記録したデータは、アプリケーションの操作に関するデータである operationType (どんな操作を行ったか、タップ、ドラッグ、ピンチなど) と operationPoint (操作を行った座標)、そして視線に関するデータ (推定した視線座標 eyePoint, 目の閉じ具合 left/rightEyeBlink など) である。分析に使用するデータを表 5.1 に示す。データは、アプリケーション操作時に視線の追跡ができていない間のみ記録される。operationPoint は、ドラッグの場合にはジェスチャ開始から終了までの座標、ピンチの場合にはジェスチャ開始時の 2 本の指のタッチの重心の座標が記録される。また、操作を行っていない場合は例外として、operationType は none, operationPoint は (-1,-1) が記録される。operationPoint の座標系は eyePoint と同様である (図 4.1)。ラウンドごとの記録したデータのフレーム数を表 5.2 に示す。

表 5.1: 分析に使用するデータ

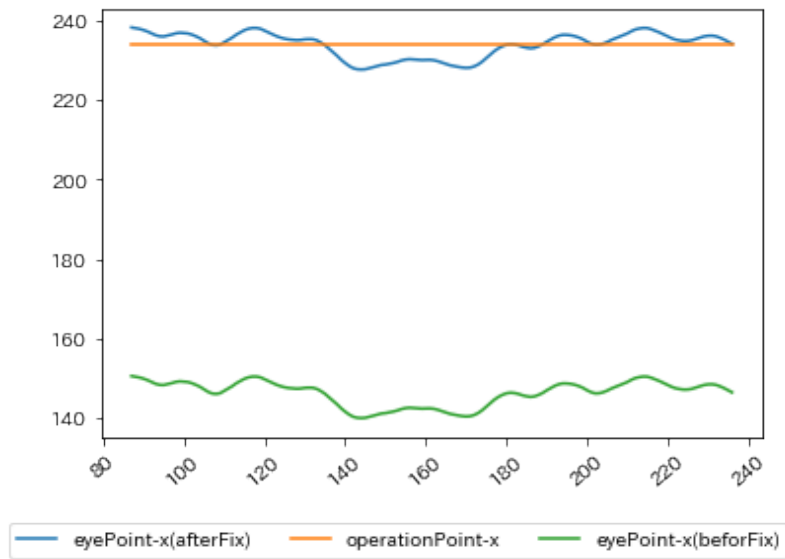
データ名	内容
eyePoint (x, y)	推定した視線座標
operationType	行った操作
operationPoint (x, y)	操作を行った座標
left・rightEyeBlink	目の閉じ具合 (左右)

表 5.2: 記録データのフレーム数

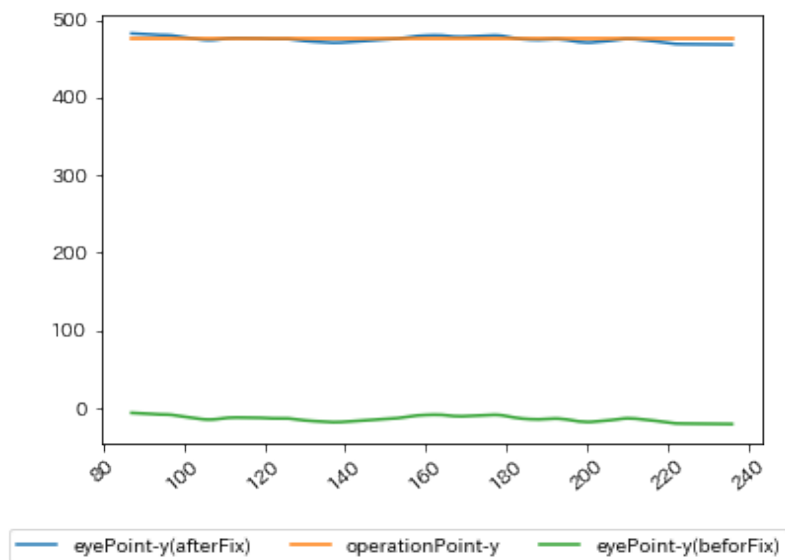
ラウンド番号	ホーム画面アプリケーション	マップアプリケーション
1	10068	8956
2	10817	10844
3	10733	10854
4	10774	10803
5	10576	10735
6	10870	7007

5.2 キャリブレーション

予備実験の結果から、推定した視線座標には、恒常的なズレと瞬間的なズレがあることがわかった。瞬間的なズレは、瞬きに基づいており、恒常的なズレと同時に発生する。そのため、まず瞬きによるズレの補正を行い、次に恒常的なズレのキャリブレーションを行う。瞬きによるズレの補正は、4.2.2 節と同様に行う。本節では、恒常的なズレのキャリブレーションについて説明する。ホーム画面アプリケーションとマップアプリケーションには、キャリブレーションのためのターゲットが表示される。初めに、このキャリブレーションのためのターゲットが表示されている間の、eyePoint の平均 (g_{ave}) を計算した。ただし最初の 150 フレームは視線が安定していないため、取り除いた。次に g_{ave} とターゲットの座標が一致するように、データ全体の座標を平行移動させた。キャリブレーションをした結果を図 5.2 に示す。図は、青線が修正前の eyePoint、オレンジ線が operationPoint、緑線が修正後の eyePoint を示す。eyePoint は修正後に operationPoint 付近に移動していることがわかる。



(a) eyePoint-x



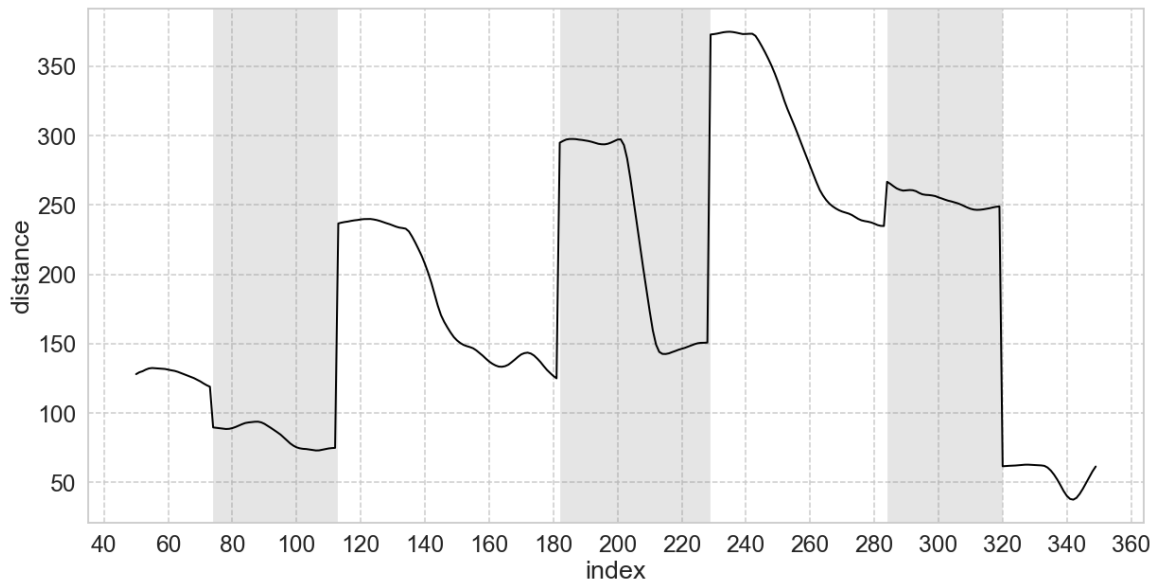
(b) eyePoint-y

図 5.2: 一定のズレの修正（青線が修正前の eyePoint, オレンジ線が operationPoint, 緑線が修正後の eyePoint を表す）

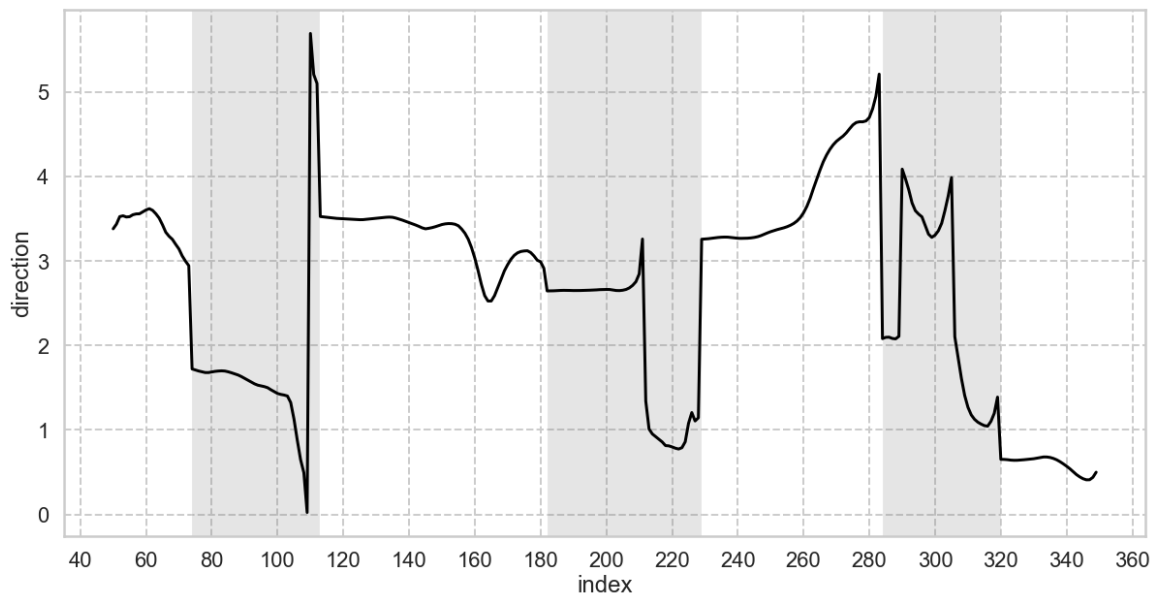
5.3 データ観察

視線データと操作の関係を見つけるために、4.3.2 節のデータ再生アプリケーションを利用して推定した視線座標と操作を観察した。データ再生アプリケーションからは、操作をする少し前から視線

座標が操作座標に近づいていくことが確認できた。また、操作を行う瞬間には、すでに視線は次の操作のために動き始めていることも観察できた。これらを検証するために、視線座標と操作座標の距離、それぞれの移動方向の角度をグラフにした（図 5.3a, 図 5.3b）。図 5.3a は、次の操作座標と現在の視線座標の距離を示し、数字が小さいほど操作座標と視線座標が近づいていることを示す。背景色の変わり目が操作された瞬間を示す。図 5.3a を見ると、次の操作の瞬間に近くに連れて、距離は短くなっており、次の操作座標へ視線座標が近づいていくことがわかる。図 5.3b は、現在の視線座標から次の操作座標への移動方向と、現在の視線座標から次のの視線座標への移動方向の差を示し、数字が小さいほど視線座標が操作座標の方向に移動し、正しい方向への移動であることを示す。図 5.3a と同様に、背景色の変わり目が操作された瞬間を示す。図 5.3b から、視線の移動方向は、操作の瞬間の少し前に最も小さくなることが多いことがわかる。これらから、操作と視線には、移動距離と移動方向において関係があることが明らかになった。



(a) 距離



(b) 角度

図 5.3: operationPoint と eyePoint の移動距離と移動方向のグラフ

5.4 分類

視線データから操作を予測できるかを調査するために、分割した画面の領域のどこで操作したかの分類（以下、操作領域分類）と、どんな操作を行ったかの分類（以下、操作分類）を行った。本節で

は、まずデータの前処理と機械学習の方法について説明し、最後に分類結果を明らかにする。

5.4.1 データの前処理

記録したデータを機械学習に用いるために、ラベル付け、セグメンテーションを行った。

データの前処理では、ラベル付けのための定数 $SPLITSIZE$ 、セグメンテーションのための2つの定数 $SEGMENTSIZ$ E、 $STEPSIZE$ を定めた。 $SPLITSIZE$ は、ラベル付けをする際に画面を何分割するかを示しており、4 (横×縦:2×2)、6分割 (2×3)、9分割 (3×3) を選択した。図 5.4 から、片手の指の届かない範囲の、 $SPLITSIZE=4$ で上下、 $SPLITSIZE=6$ 、9 で上下左右を分類できると考えられる。



図 5.4: 片手の指の届く範囲

$SEGMENTSIZ$ E は、1つの操作に対して分類の入力に使用する、視線の振る舞いのセグメントの長さを示す。 $STEPSIZE$ は、操作と関係のある視線の振る舞いのセグメントと操作までのフレーム数の許容範囲を示す。図 5.3b に対するセグメントは図 5.5 のようになる。操作直前の1つのセグメント (図 5.5 の segment1) のみを用いる場合は $STEPSIZE=1$ となる。 $STEPSIZE$ が増えるごとに、操作直前から1フレームずつ前にずらしたセグメント ($STEPSIZE=2$ では、segment1 と segment2, $STEPSIZE=3$ では、segment1, segment2 および segment3) を学習時の入力に用いる。

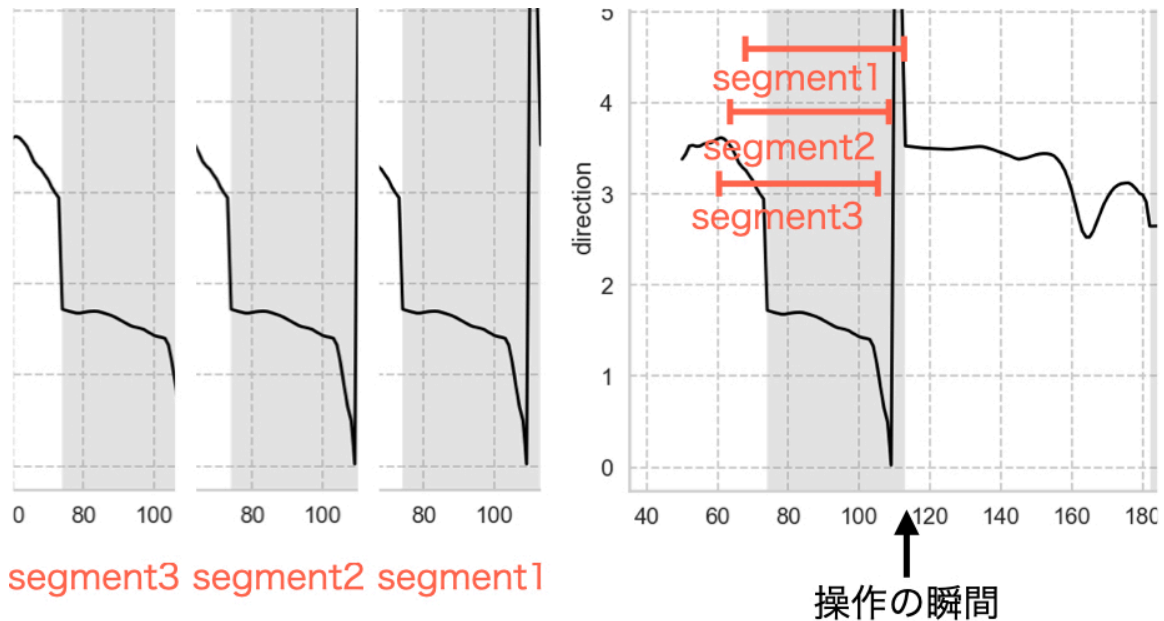


図 5.5: SEGMENTSIZESIZE と STEPSIZE の例

また、データ観察から視線の移動距離と移動方向が操作と関連していると考えられるため、前のフレームからの移動距離 $distance$ と移動方向 $direction$ を計算した。

ラベル付け

本研究では、操作領域分類に用いるデータに、 $area$ ラベル、 $next_area$ ラベル、をつけた。操作領域分類では、画面の分割数を $SPLITSIZE$ とすると現在のフレーム（以下、 f_{now} ）の $operationPoint$ に対応する $1 \sim SPLITSIZE$ までのラベル $area$ を付ける。（ $operationPoint$ が $(-1,-1)$ の場合は、例外として -1 でラベル付けする）。本研究では、操作をする前の視線の振る舞いから操作を分類するため、次の操作の $area$ ラベルを $next_area$ ラベルに付けた。 f_{now} の次のフレーム（以下、 f_{next} ）の $area$ ラベルが変わる場合（ラベル -1 は操作をしていないことを表すので除く）、 f_{now} のフレームの $next_area$ ラベルには、 f_{next} の $area$ ラベルが入る。画面の領域に対応するラベルは図 5.6a、図 5.6b、図 5.6c のようにした。

また、操作分類に用いるデータには、 $next_area$ ラベルのみをつけた。操作領域分類の $area$ ラベルの代わりに、どんな操作を行ったかを示すラベル $operationType$ を用いた。 $operationType$ は視線データ記録時に一緒に記録される。記録されるラベルは、 tap （タップ）、 $drag$ （ドラッグ）、 $pinch$ （ピンチ）である。

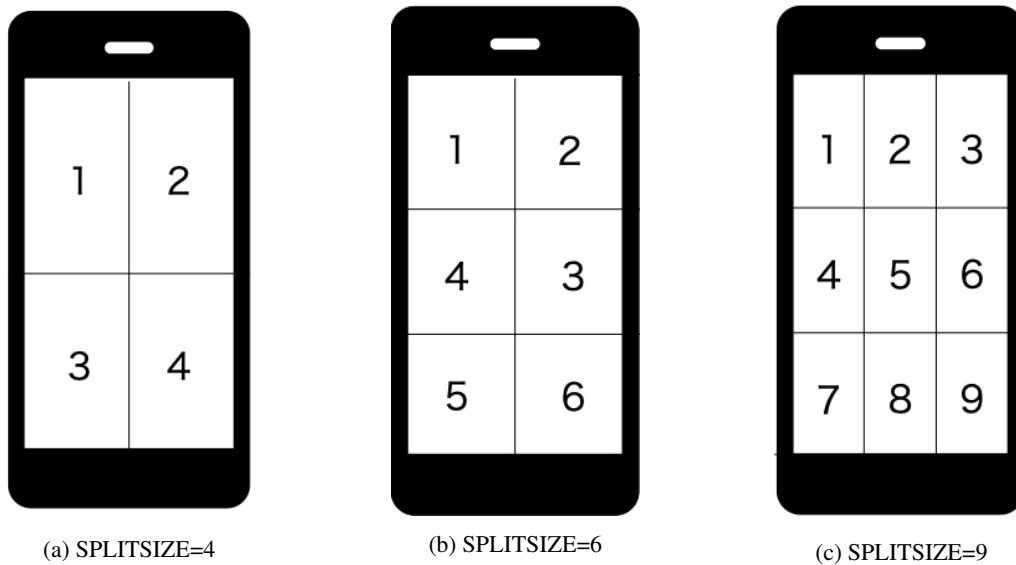
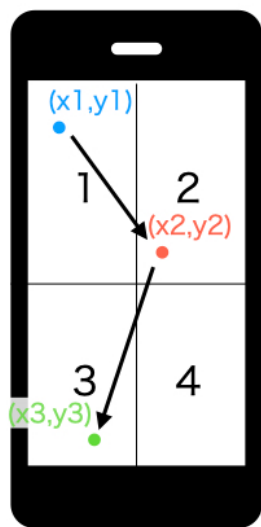


図 5.6: 画面の領域に対応するラベルの例



(a) 操作の例

targetPoint	area	next_area
(x1,y1)	1	-1
(x1,y1)	1	-1
(-1,-1)	-1	2
(x2,y2)	2	-1
(x2,y2)	2	-1
...
(x2,y2)	2	-1
(x2,y2)	2	3
(x3,y3)	3	-1

(b) データ例

図 5.7: 操作に対するラベル付けの例 (SPLITSIZE=4)

セグメンテーション

連続した一定数のフレームの centerEyeLookAtPoint を入力として分類をするために、記録したデータを時系列で分割するセグメンテーションを行った。入力 (以下, segx) に使う centerEyeLookAtPoint

のフレーム数が SEGMENTSIZ E, 操作直前とするフレーム数が STEPSIZ E となる。操作領域分類では, f_{now} の next_area が -1 以外の場合に, segx に f_{now} を含めた前の SEGMENTSIZ E 個のフレームの eyePoint, distance, direction を追加し, 正解データ (以下, segy) には f_{now} の next_area を追加する。これを f_{now} から 1 フレームずつ前にずらしながら, STEPSIZ E 回繰り返す。操作分類も操作領域分類と同様にセグメンテーションする。記録したデータが図 5.8 のような場合, SEGMENTSIZ E=5, STEPSIZ E=1 とした場合のセグメンテーションは以下ようになる。

eyePoint	distance	direction	next_area
(x1,y1)	dis1	dir1	-1
(x2,y2)	dis2	dir2	-1
(x3,y3)	dis3	dir3	-1
(x4,y4)	dis4	dir4	-1
(x5,y5)	dis5	dir5	-1
(x6,y6)	dis6	dir6	2
(x7,y7)	dis7	dir7	-1
(x8,y8)	dis8	dir8	-1
(x9,y9)	dis9	dir9	3

図 5.8: データ例

$$segmentx = [[[x2, y2, dis2, dir2], [x3, y3, dis3, dir3], \dots, [x6, y6, dis6, dir6]], \\ [[x5, y5, dis5, dir5], \dots, [x9, y9, dis9, dir9]]]$$

$$segmenty = [2, 3]$$

また, SEGMENTSIZ E=5, STEPSIZ E=2 とした場合のセグメンテーションは以下ようになる。

$$segmentx = [[[x1, y1, dis1, dir1], [x2, y2, dis2, dir2], \dots, [x5, y5, dis5, dir5]], \\ [[x2, y2, dis2, dir2], \dots, [x6, y6, dis6, dir6]], [[x4, y4, dis4, dir4], \dots, [x8, y8, dis8, dir8]], \\ [[x5, y5, dis5, dir5], \dots, [x9, y9, dis9, dir9]]]$$

$$segmenty = [2, 2, 3, 3]$$

両方の分類において, ラベルの数を揃えるために, 正解ラベルの中で最も数が少ないラベルの数に合わせて, ランダムな順番で数が多いラベルのセグメントを減らしていく。最後に学習用のセグメントの数を確保するために, 1 回のラウンドから作成したセグメントを 6 ラウンド分連結する。

5.4.2 機械学習

本研究では、視線データから操作の分類を行うために機械学習をした。機械学習には、Python の機械学習ライブラリである scikit-learn¹を用いた。scikit-learn で扱える様々な機械学習アルゴリズムで分類を行ったところ、ランダムフォレストの精度が高かったため、ランダムフォレストを採用した。また、各モデルごとに、ハイパパラメータを最適化するために、収集した学習データに対して交差検証を用いたグリッドサーチを行う。グリッドサーチは、指定されたハイパパラメータの総当たりの組み合わせのモデルを構築し、最も性能の良いパラメータの組を決定する。ハイパパラメータの最適化後、最も精度の良かったモデルを分類に用いる。用いたハイパパラメータを表に示す。

表 5.3: グリッドサーチに用いたハイパパラメータ

パラメータ名	最小値	最大値	探索幅
n_estimators	10	100	10
max_depth	1	20	1

パラメータ名	パラメータ
criterion	gini, entropy
random_state	777

5.4.3 分類結果

操作領域分類

ホーム画面アプリケーションのデータを用いて、操作領域分類をするモデルを作成した。そのために、はじめにデータの前処理のための定数 SEGMENTSIZ E の値を設定した。そのために、SEGMENTSIZ E を 1 から 100 とした時の分類モデルの正解率を計算した (SPLITSIZ E=4 とした)。正解率は、記録した 6 回ラウンドのデータを、5 分割で交差検証を行い、その平均値とする。この結果を図 5.9 に示す。SEGMENTSIZ E=75 の時に正解率が最も高くなった (正解率: 0.75)。よって操作領域分類では、SEGMENTSIZ E=75 を選択する。

¹<https://scikit-learn.org/stable/>

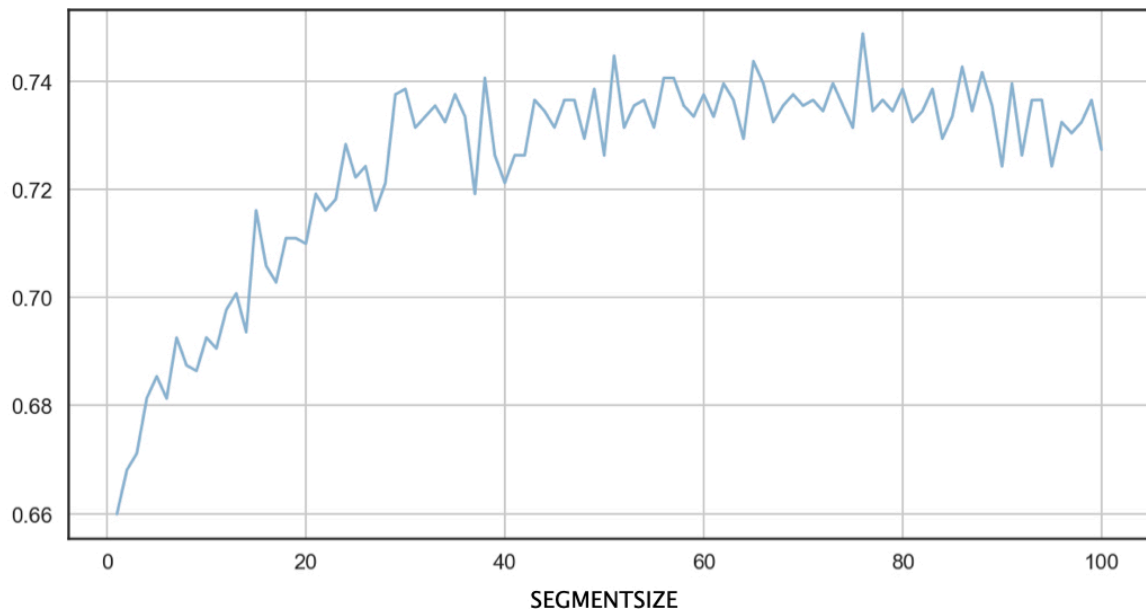
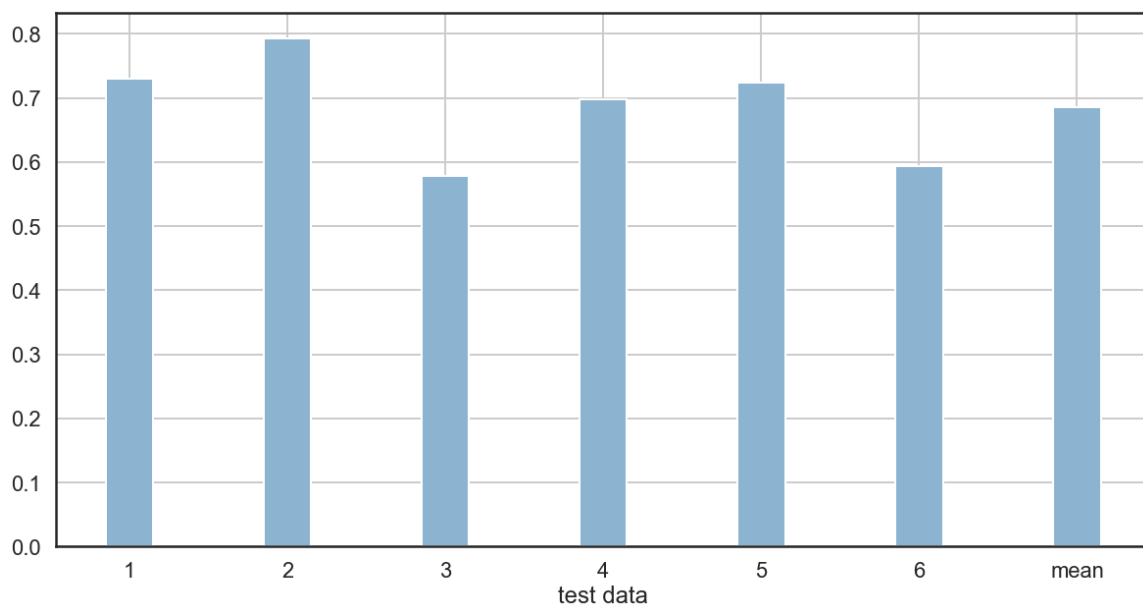
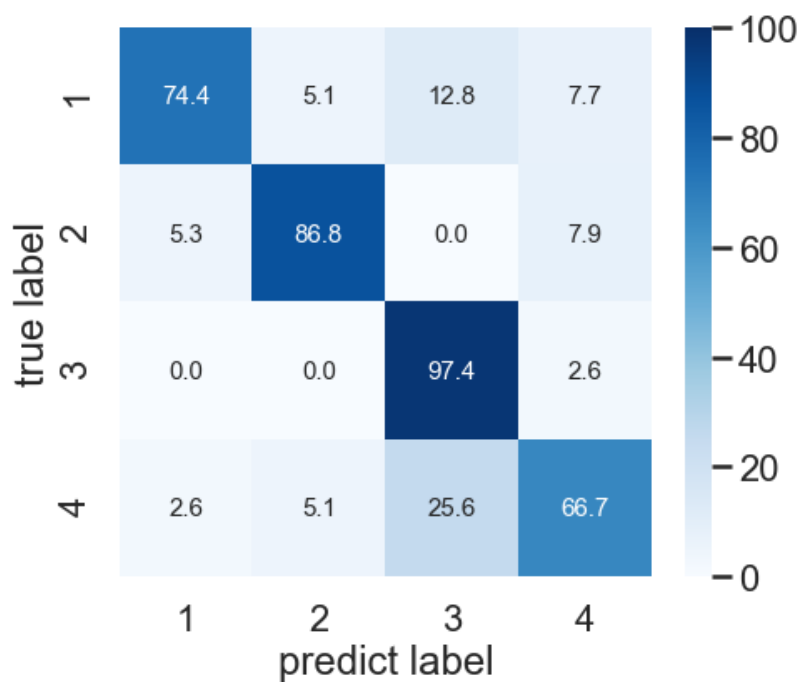


図 5.9: SEGMENTSIZESIZE を 1 から 100 とした場合の正解率（操作領域分類, SPLITSIZE=4）

次に、記録した各ラウンドの偏りによる影響を確認するために、記録した 6 回のラウンドのうち 5 回を訓練データとして選択し、残り 1 回のラウンドをテストデータとした場合の正解率を計算した。この結果を図 5.10a に示す。平均値は 0.69、標準偏差は 0.07 であった。また、正解ラベルに対する予測ラベルの正解率を計算した。図 5.10b に最も正解率が高かったラウンド 2 の結果を示す（正解率：0.79）。正解ラベル 4 の場合の正解率が最も低く、ラベル 3 との間違が多かったことがわかった。



(a) 記録した6回のラウンドのうち5回を訓練データとして選択し、残り1回のラウンドをテストデータとした場合の正解率



(b) 正解ラベルに対する予測ラベルの正解率 (ラウンド2)

図 5.10: 操作領域分類の正解率 (SEGMENTSIZ=75, STEPSIZE=1, SPLITSIZE=4)

図 5.9 を見ると、SEGMENTSIZ は 40 から 100 の間の正解率は一定の水準で推移していることが

わかる。これは各操作によって、影響を与えるフレームのセグメントに時間のズレがあるためであると考えられる。このことから、分類器の学習に使うセグメントに、一定のフレーム数ずらしたものを追加し、ある操作に対して入力を複数与えることで、正解率の向上が期待されることが考えられる。そこで、操作直前からずらすフレーム数を示す STEPSIZE を設定する。ある操作直前のフレーム `ope_frame` がある時、STEPSIZE=1 の場合は、そのフレームから SEGMENTSIZ E フレーム遡ったセグメントのみを入力として用いるが、STEPSIZE=2 の場合、さらに `ope_frame` よりも 1 つ前のフレームから、作成したセグメントも入力に追加する。つまり、1 回の操作に対して STEPSIZE 個のセグメントが入力として用いられる。適切な STEPSIZE の値を設定するため、STEPSIZE を 1 から 25 (=100-75) とした時の正解率を計算した。SEGMENTSIZ E は先ほど設定した 75 とする。この結果を図 5.11 に示す。正解率は STEPSIZE=4 の時に正解率は最も高く（正解率：0.74）となり、STEPSIZE が 4 を超えると正解率は低くなっていった。

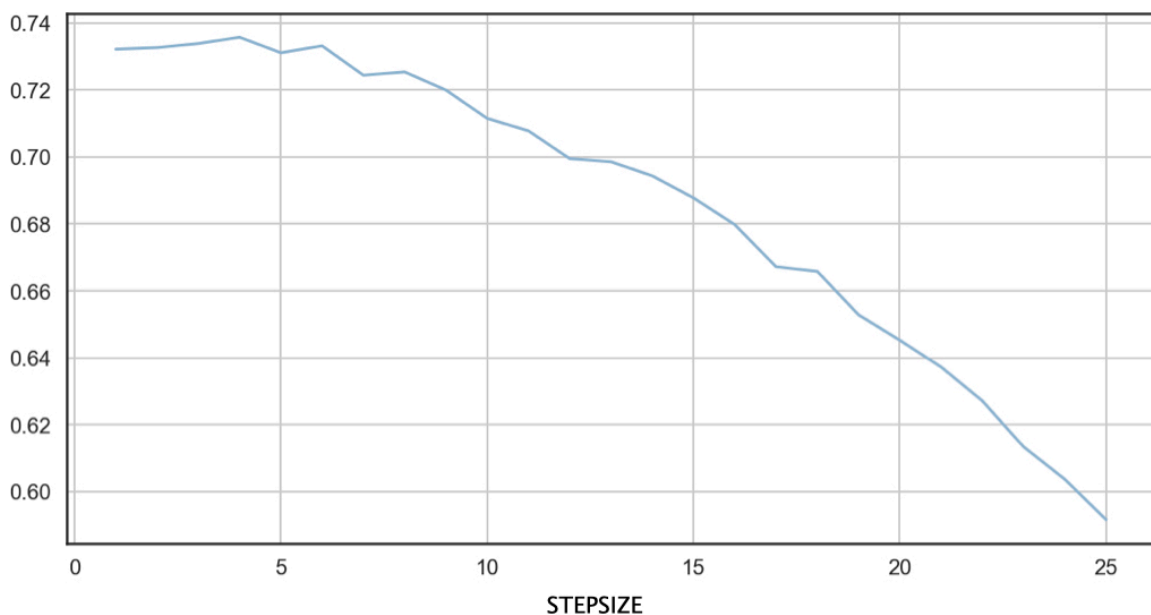
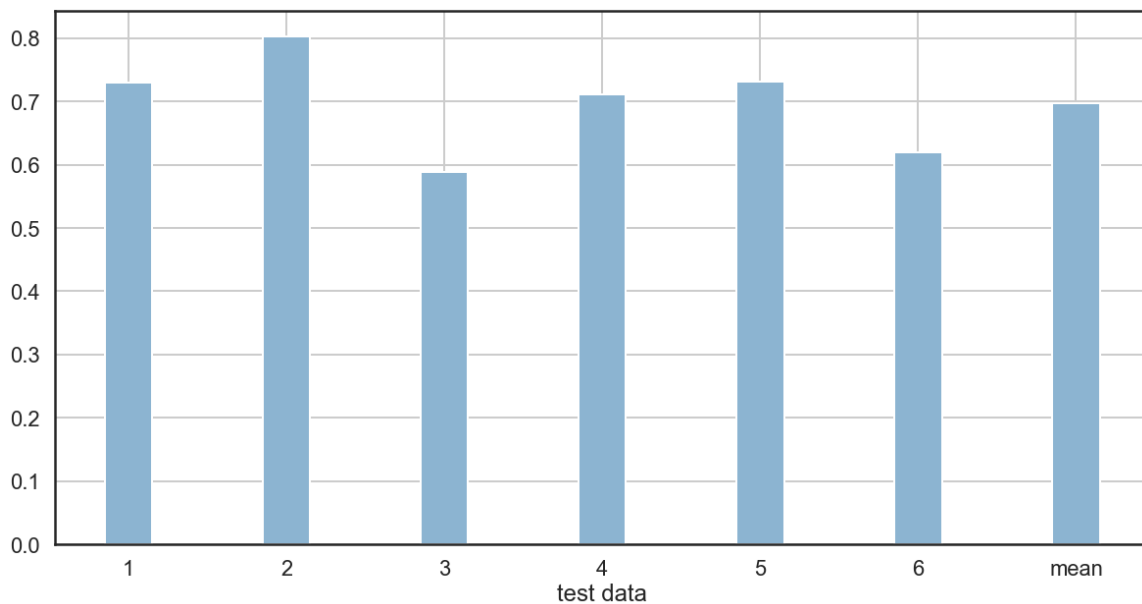
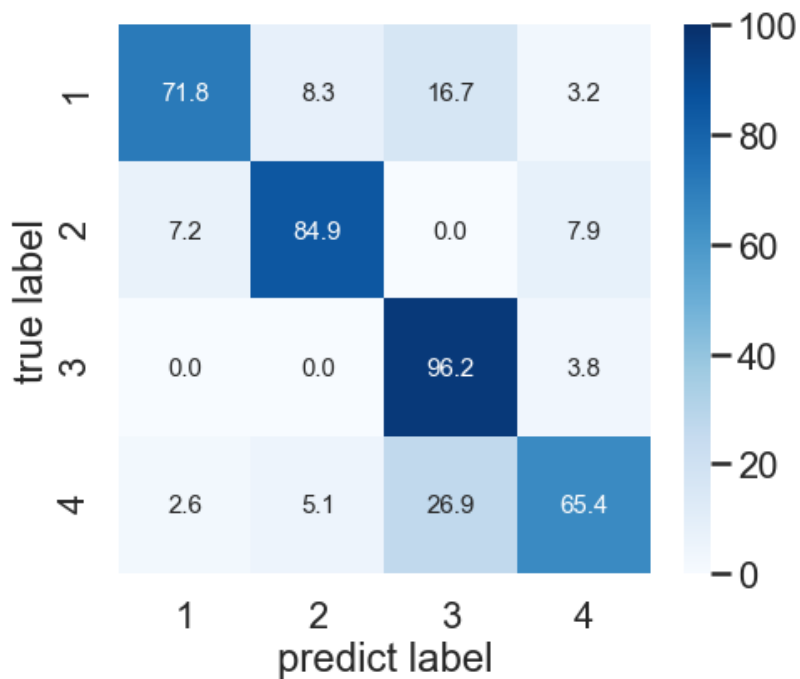


図 5.11: STEPSIZE を 1 から 100 とした場合の正解率（操作領域分類，SPLITSIZ E=4）

この STEPSIZE を用いて、記録した 6 回のラウンドのうち 5 回を訓練データとして選択し、残り 1 回のラウンドをテストデータとした場合の正解率を計算した。結果を図 5.12a に示す。平均値は 0.70、標準偏差は 0.07 となった。また正解ラベルに対する予測ラベルの正解率を計算した。図 5.12b に、最も正解率が高かったラウンド 2 の結果を示す（正解率：0.80）。STEPSIZE=1 の場合（平均値：0.69）と比べると、あまり変化がなかったため以降では、STEPSIZE=1 とする。



(a) 記録した6回のラウンドのうち5回を訓練データとして選択し、残り1回のラウンドをテストデータとした場合の正解率

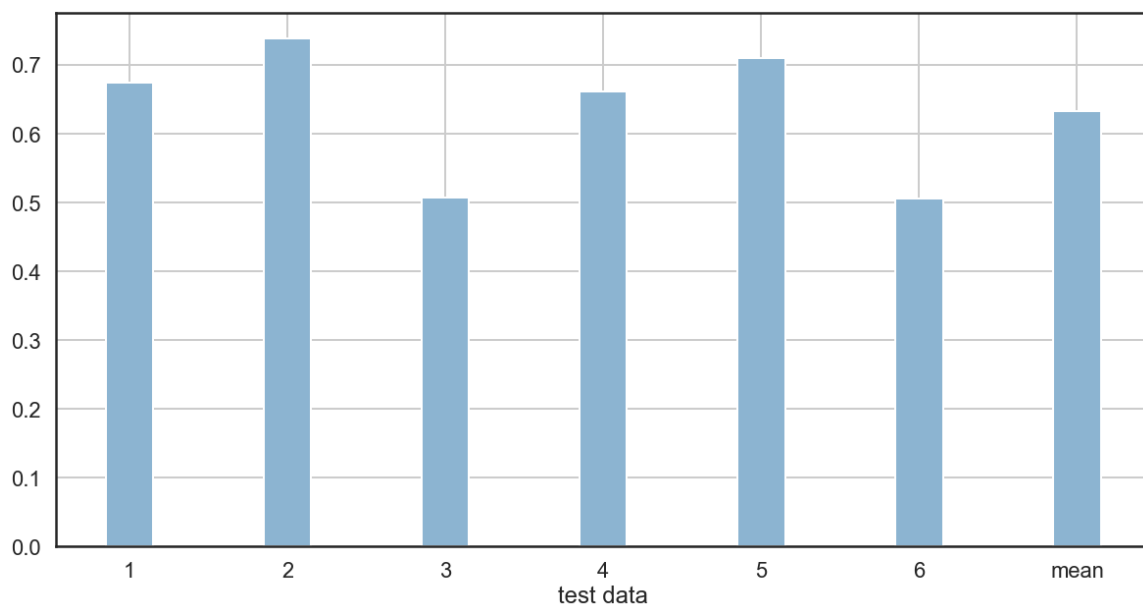


(b) 正解ラベルに対する予測ラベルの正解率 (ラウンド2)

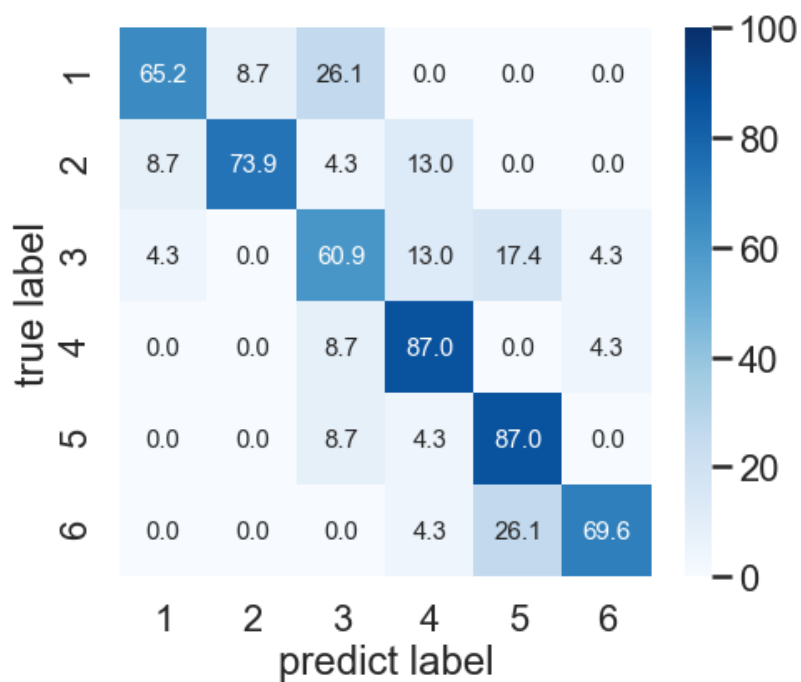
図 5.12: 操作領域分類の正解率 (SEGMENTSIZ=75, STEPSIZE=4, SPLITSIZE=4)

以上の結果は、SPLITSIZE=4 の場合の正解率である。そこで、さらに画面分割数を増やした場合の

正解率を計算した。SPLITSIZE は新たに 6, 9 を用いた。まず、記録した 6 回ラウンドのデータを、5 分割で交差検証を行い、その平均値を計算した。結果は SPLITSIZE=6 で 0.67, SPLITSIZE=9 で 0.49 となった。次に、記録した 6 回のラウンドのうち 5 回を訓練データとして選択し、残り 1 回のラウンドをテストデータとした場合の正解率を計算した。結果を図 5.13a, 図 5.14a に示す。SPLITSIZE=6 の平均は 0.63, SPLITSIZE=9 は 0.47 となり標準偏差はそれぞれ、0.09, 0.08 となった。また、正解ラベルに対する予測ラベルの正解率を図 5.13b と図 5.14b に示す。SPLITSIZE=6 では、ラウンド 2 の正解率が最も高く (0.74), 正解ラベル 1 が予測ラベル 3 と、正解ラベル 3 が予測ラベル 4, 5 と、正解ラベル 6 が予測ラベル 5 と間違いが多かった。SPLITSIZE=9 では、ラウンド 1 の正解率が最も高かった (0.59)。

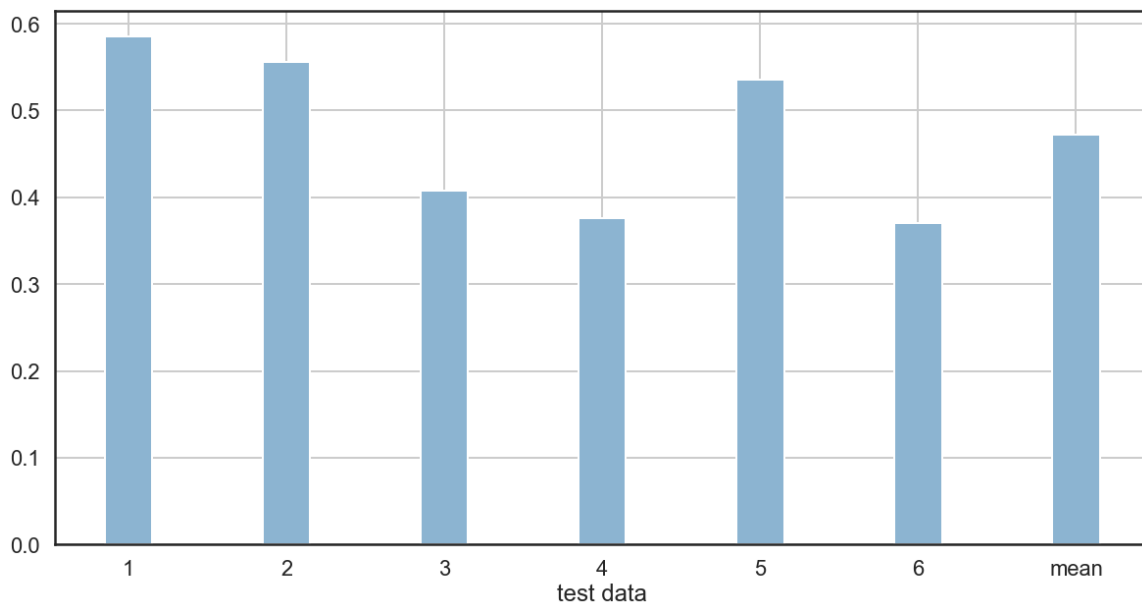


(a) 記録した6回のラウンドのうち5回を訓練データとして選択し、残り1回のラウンドをテストデータとした場合の正解率

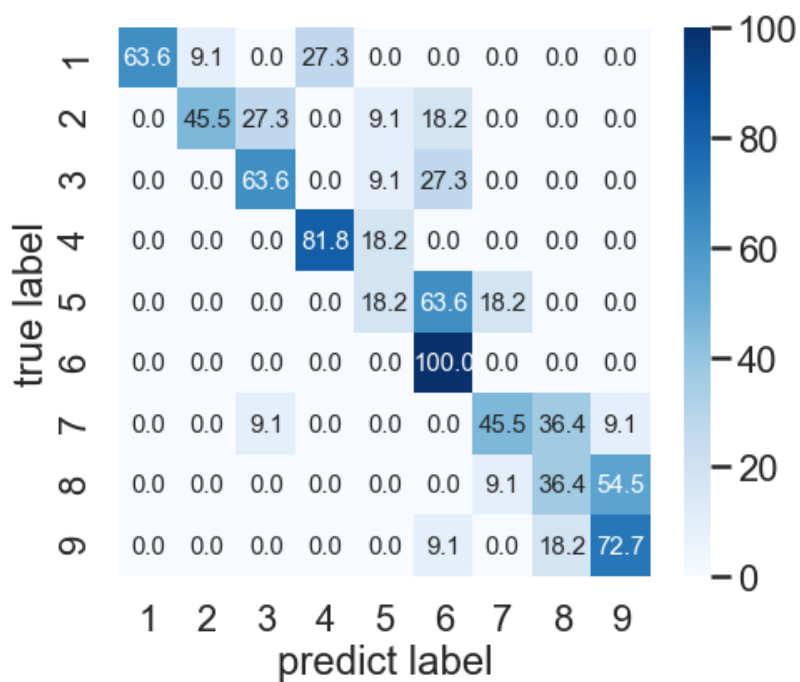


(b) 正解ラベルに対する予測ラベルの正解率 (ラウンド2)

図 5.13: SPLITSIZE=6 の場合の正解率 (操作領域分類, SEGMENTSIZ=75, STEPSIZE=1)



(a) 記録した6回のラウンドのうち5回を訓練データとして選択し、残り1回のラウンドをテストデータとした場合の正解率



(b) 正解ラベルに対する予測ラベルの正解率 (ラウンド 1)

図 5.14: SPLITSIZE=9 の場合の正解率 (操作領域分類, SEGMENTSIZ=75, STEPSIZE=1)

操作分類

操作分類は、様々なタッチジェスチャを行うマップアプリケーションのデータを利用して分類した。分類したジェスチャは、タップ、ドラッグ、ピンチである。SEGMENTSIZ E は、行う操作によって異なる可能性があるため、操作領域分類と同様に、SEGMENTSIZ E を 1 から 100 とした時の分類モデルの正解率を計算した。この結果を図 5.15 に示す。SEGMENTSIZ E=27 の時に正解率が最も高くなった（正解率：0.56）。また、50 以降では正解率は徐々に低くなっていくことがわかった。よって操作領域分類では、SEGMENTSIZ E=27 を選択する。

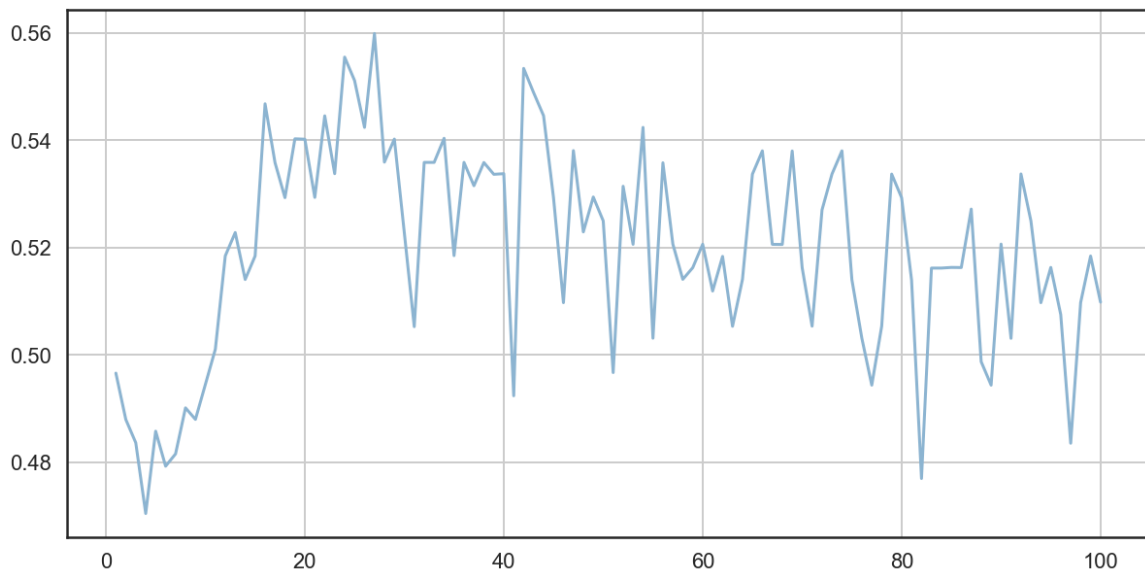
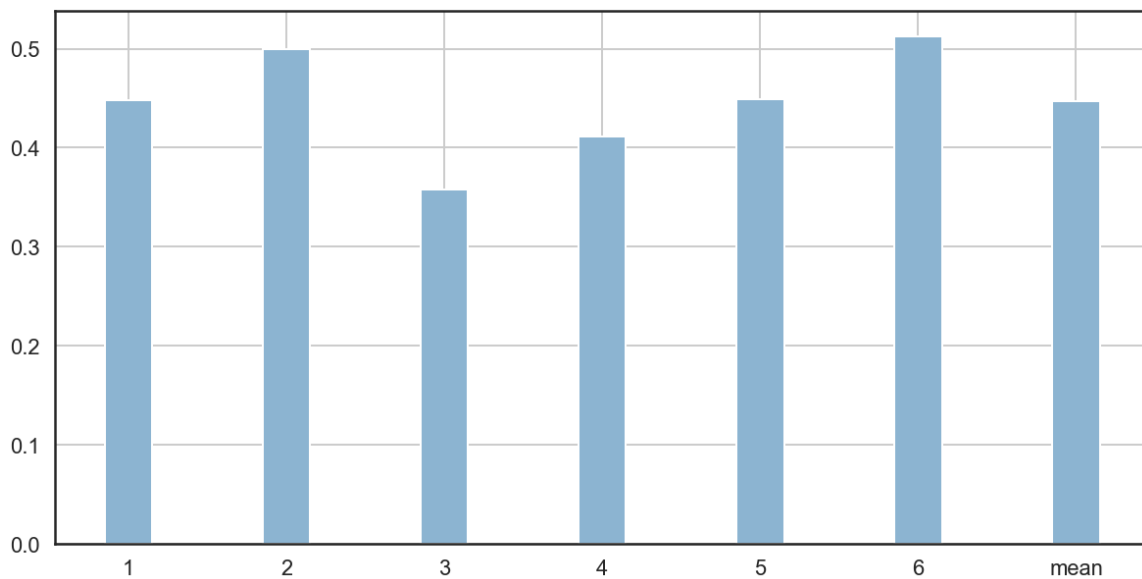
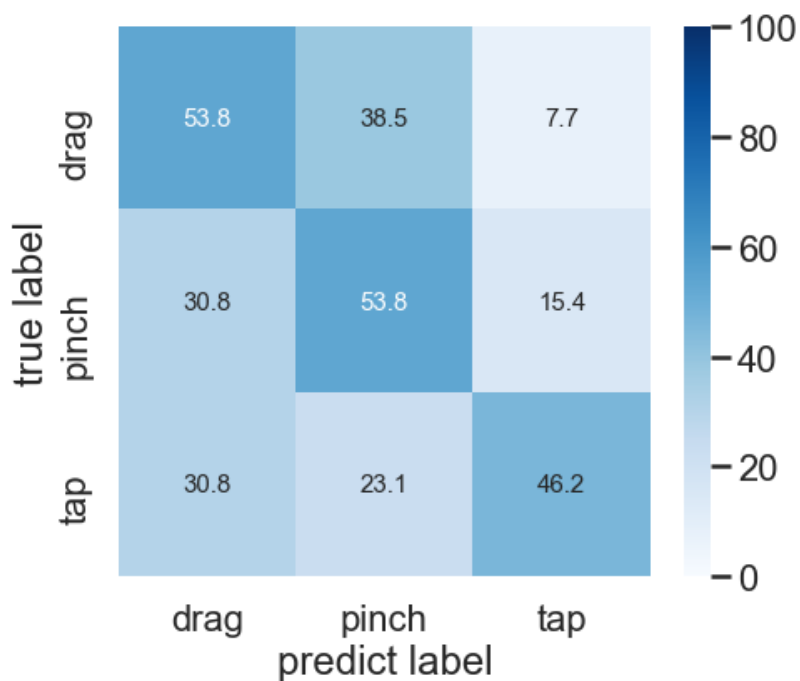


図 5.15: SEGMENTSIZ E を 1 から 100 とした場合の正解率（操作分類）

次に、記録した各データの偏りによる影響を確認するために記録した 6 回のラウンドのうち 5 回を訓練データとして選択し、残り 1 回のラウンドをテストデータとした場合の正解率を計算した。この結果を図 5.16a に示す。平均値は 0.45 となり、標準偏差は 0.07 であった。また、正解ラベルに対する予測ラベルの正解率も計算した。最も正解率が高かったラウンド 6 の場合を図 5.16b に示す（0.51）。正解ラベルに対して予測ラベルがドラッグとなる間違いが多かった。



(a) 記録した6回のラウンドのうち5回を訓練データとして選択し、残り1回のラウンドをテストデータとした場合の正解率



(b) 正解ラベルに対する予測ラベルの正解率 (ラウンド6)

図 5.16: 操作分類の正解率 (SEGMENTSIZ=27, STEPSIZE=1)

次に操作領域分類と同様に STEPSIZE を設定する。 STEPSIZE=1 の場合、 SEGMENTSIZ が 15 から 50 で一定の正解率で推移した。そこで STEPSIZE を 1 から 23 (=50-27) とした時の分類モデルの正解

率を計算した。SEGMENTSIZ E は先ほど設定した 27 とする。この結果を図 5.17 に示す。STEP SIZE=1 の時に正解率は最も高くなった（正解率：0.55）。よって STEP SIZE=1 を採用する。

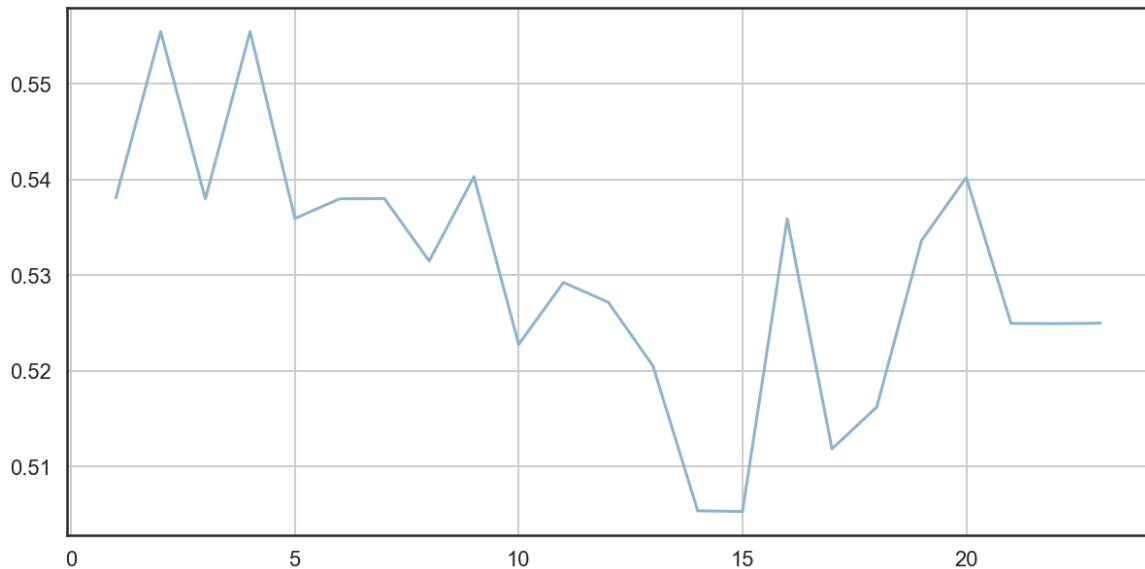
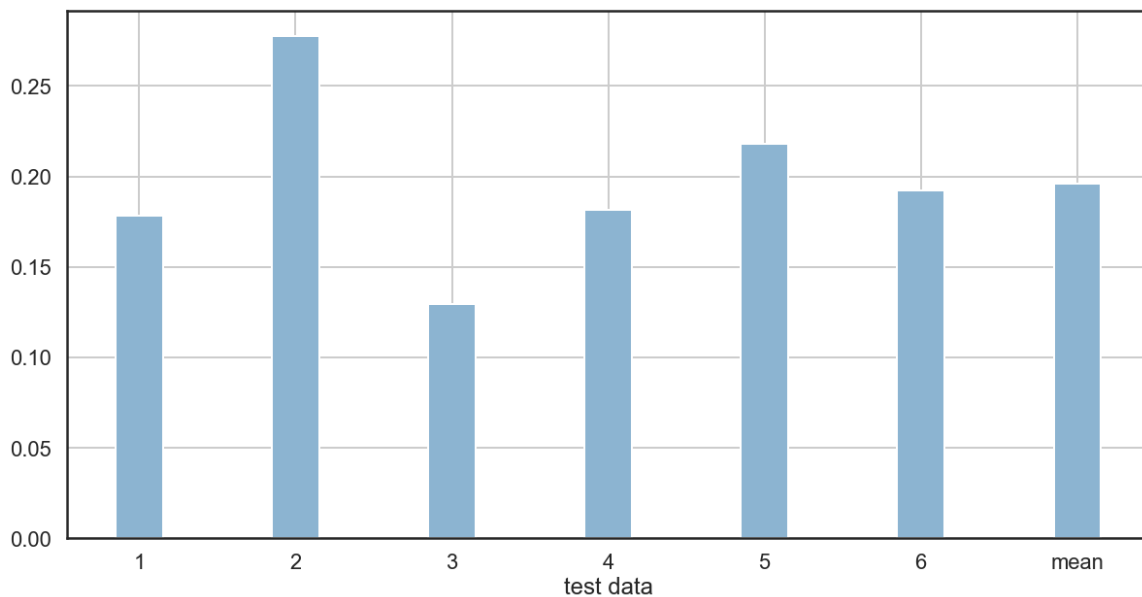
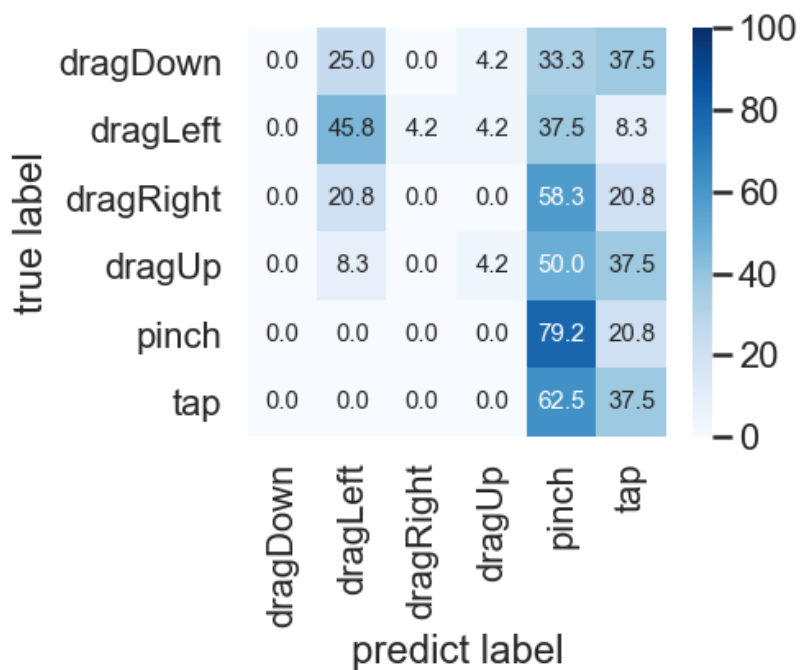


図 5.17: STEP SIZE を 1 から 100 とした場合の正解率（操作分類，SEGMENTSIZ E=27）

以上の場合では drag を方向によらず 1 つのラベルとして分類したが，drag は操作座標を用いて移動方向ごとに分けることが可能である。方向を分けて分類できればより多様な支援が可能になると考え，新たにラベル drag を上下左右の 4 つのラベルに分けた（dragUp, dragDown, dragLeft, dragRight）。この場合の，記録した 6 回ラウンドのデータを，5 分割で交差検証を行い，その平均値を計算した。結果は 0.27 となった。次に，記録した 6 回のラウンドのうち 5 回を訓練データとして選択し，残り 1 回のラウンドをテストデータとした場合の正解率を計算した。この結果を図 5.18a に示す。平均値は 0.20 となり，標準偏差は 0.04 であった。drag を 1 つのラベルとした場合と比べると，正解率は向上しなかった。また，正解ラベルに対する予測ラベルの正解率も計算した。最も正解率が高かったラウンド 2 の場合を図 5.18b に示す（正解率：0.28）。



(a) 記録した6回のラウンドのうち5回を訓練データとして選択し、残り1回のラウンドをテストデータとした場合の正解率



(b) 正解ラベルに対する予測ラベルの正解率 (ラウンド2)

図 5.18: 操作分類の正解率 (ラベル数:6, SEGMENTSIZ=27, STEPSIZE=1)

第6章 考察

6.1 用いた視線の振る舞い

本研究では、操作を分類するために用いる視線の振る舞いの長さ、操作と関係のあるセグメントと操作までのフレーム数の許容範囲を示す2つの定数を設定した。この定数に対する考察を述べる。SEGMENTSIZEはホーム画面アプリケーションとマップアプリケーションでは、異なる値の成功率が高かった。これは、ホーム画面アプリケーションとマップアプリケーションでは行う操作が異なっていたためだと考えられる。操作の種類ごとに視線の動き出すタイミングが異なる可能性がある。本研究では、操作位置に対する視線の特徴を、操作分類でも用いた。しかし、操作の種類に対する視線の振る舞いは調査しなかった。そのため、様々な操作に対して、特定の操作のみを行う場合の視線の振る舞いを調査する必要がある。STEPSIZEを1より大きくすることで、操作ごとの視線の振る舞いが発生する時間のズレに対応でき、予測精度が向上するのではないかと仮定し、正解率を計算したが、正解率が向上することはなかった。本研究ではデータを1人のみで記録していたことから、同一のユーザでは操作に対する視線の振る舞いの時間にズレが少ない可能性が考えられる。複数のユーザの場合に変化が現れるのかを調査する必要がある。STEPSIZEを増加させることによって、訓練データを増やすことが可能であり、このことはより少ない訓練データからでも分類を可能になると考えられる。

6.2 操作領域分類

ホーム画面における操作領域分類では、SPLITSIZE=4, 6の場合には、正解率は0.6を超えたが、9の場合には、正解率は0.5に届かなかった。しかし、3章で述べたようにユーザが指の届かない範囲を操作しようとしていることを予測するためには、画面の上方や持ち手から遠い範囲を操作しようとしていることがわかれば良いため、SPLITSIZE=4, 6の場合の予測ができれば十分であると考えられる。これによって、指の届かない範囲の操作予測の可能性が示された。また正解ラベルと予測ラベルの正解率を見ると、分類の間違いが発生しているのは上下または左右に隣あったラベルであった。これは4章の予備実験で示した視線推定のズレのためであると考えられ、細かな座標の分類が難しいことがわかった。すなわち、視線推定精度が向上することによって、より正確な操作領域分類が可能になると考えられる。

6.3 操作分類

マップアプリケーションにおける操作分類では、dragを1つの操作として扱うよりも、方向を分けて分類できればより多様な支援が可能になると考え、dragを方向ごとに分けた。しかし正解率は最高

で0.3に届かず、dragの方向を分類することは難しいことがわかった。さらにdragを1つの操作として扱った場合でも、正解率は0.6に届かなかったことから、視線の振る舞いから操作の種類を分類するのは難しいことがわかった。操作分類の精度を向上させるためには、視線の振る舞いだけでなく、ユーザの過去の操作傾向なども学習し、それと組み合わせて用いる方法が考えられる。

6.4 ラウンド間の正解率

データ間の正解率にはばらつきがあり、操作領域分類では、ラウンド3とラウンド6が、操作分類では、ラウンド3が低かった。また、4章の予備実験の結果からも、記録したラウンドごとに推定した視線位置と実際に見ている位置とのズレが異なっていた。これらは、データ記録をするごとに、スマートフォンの把持姿勢が変わってしまうためだと考えられる。視線推定には、デバイスやユーザの目の位置・向きが用いられるため、それらの環境が視線推定の精度に影響すると考えられる。しかし、本研究ではスマートフォン操作時の自然な視線の振る舞いを用いるために、スマートフォンやユーザの顔を固定することはしなかった。そのため、分類精度をより高くするためには、詳細なキャリブレーションが考えられる。

6.5 今後の展望

本研究では、操作と視線の関係について明らかにし、視線の振る舞いによって操作分類が可能であるかどうかを検証した。今回の分類精度は高々0.8であり、3章にて提案した自動操作を実現する場合には語分類による意図しない操作が発生してしまう可能性が高い。一方で、3章にて提案したもう一つの手法であるユーザ支援、すなわちモード切替えやレコメンド表示などを行える可能性がある。今後は、その手法を実際のアプリケーションに実装する方法や、その評価などを行う必要がある。また、本研究では操作が行われるとわかっているデータのみしか用いなかったが、実際にユーザ支援に用いるには、操作直前かどうかを判断する必要がある。そのためにスマートフォンに搭載された別のセンサ（加速度センサやジャイロセンサ）データと組み合わせる方法も考えられる。

本研究では、データの記録を著者1名のみで行ったため、ユーザ間の影響については調査できなかった。複数人のデータを記録することで、利き手や年齢、スマートフォンの習熟度などによる影響を調査することが可能になると考えられる。

本研究では、実際にユーザが見ている点を正確に推定することはできなかった。これは、スマートフォンの把持姿勢や、周囲の環境などが影響していると考えられる。これらの影響をなくすためには、細かなキャリブレーションやスマートフォンの固定が必要になるが、その場合の視線の振る舞いは通常の操作時の視線の振る舞いとは異なる可能性があるため。そのため、メガネ型視線追跡装置などの、スマートフォン操作に影響を与えない視線追跡装置による調査が必要であると考えられる。

スマートフォンと似た、視線の振る舞いが得られると考えられるタブレットでの調査も必要であると考えられる。今回スマートフォンでは、画面の分割数を多くすると正解率が低下したが、より画面の大きなタブレットにおいては、視線の動きも大きくなると考えられる。さらに画面が大きくなることによって、分類精度も向上するのではないかと考えられる。さらに、タブレットはスマートフォン

よりも画面が大きいことから、指の届かない範囲も大きく、その操作を予測することによるユーザ支援がより必要とされることが考えられる。

第7章 結論

本研究では、スマートフォンの操作と視線の関係を明らかにするために、スマートフォンの正面カメラを使用して視線推定を行い、スマートフォンの操作に関するデータとスマートフォンに向けられる視線データ記録・分析した。

記録したデータを観察したところ、推定した視線位置は、実際に見ている位置からズレているものの、移動距離や移動方向が操作座標と関係していることがわかった。スマートフォン操作時の自然な視線の振る舞いから、操作予測が可能かどうかを検証するために、分類モデルを作成した結果、操作領域分類では、SPLITSIZE=4 の場合 0.75, 6 の場合で 0.67, 9 の場合で 0.49 となった。このことから、ユーザが指の届かない範囲の操作を実行しようとする进行分类することは可能であることが示された。また操作分類では、drag を 1 つのラベルとした場合の正解率のが 0.56, drag を上下左右のラベルに分けた場合が 0.27 となった。このことから、視線の振る舞いによって操作自体进行分类することは難しいことがわかった。

スマートフォン操作時の自然な視線の振る舞いから、操作を予測できれば、操作モードの自動的な切り替えや操作のレコメンドが可能になる。例えば、ユーザが指の届かない範囲を操作しようすることを検出し、自動でカーソルを表示することができる。これによって、ユーザはトリガなしで新たな操作方法を実行することができ、トリガによる数の制約がなくなるためより多くの種類の操作を行うことも可能になる。また、ユーザが次に行おうとしている操作の種類を特定し、その操作のレコメンドを表示することができる。そして、レコメンドの表示によって、スマートフォン初心者のお操作を助けることが可能になる。

今後は、より多くのユーザのデータを集めることによる、ユーザの利き手や年齢、スマートフォンの習熟度などの影響の調査や、より細かなキャリブレーションや他の視線追跡装置を用いることによる視線推定精度の向上、より大きなデバイスであるタブレットにおける視線の振る舞いの調査が必要であると考えられる。また、視線の振る舞いだけでなく、スマートフォンの他のセンサデータも組み合わせた場合の分類精度や、実際に分類器をアプリケーションに実装し、どのようなユーザ支援を提供できるかの調査、そしてその評価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高山雄太さんと、スマートフォンにおける視線という分野において多大なるご助力を頂いた鈴木雄太郎さん、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

- [1] 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.
- [2] 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.
- [3] Joanna Bergstrom-Lehtovirta and Antti Oulasvirta. Modeling the functional area of the thumb on mobile touchscreen surfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '14*, pp. 1991–2000, New York, NY, USA, 2014. Association for Computing Machinery.
- [4] Huy Viet Le, Sven Mayer, Patrick Bader, and Niels Henze. Fingers' range and comfortable area for one-handed smartphone interaction beyond the touchscreen. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, CHI '18*, pp. 1–12, New York, NY, USA, 2018. Association for Computing Machinery.
- [5] Sunjun Kim, Jihyun Yu, and Geehyuk Lee. Interaction techniques for unreachable objects on the touchscreen. In *Proceedings of the 24th Australian Computer-Human Interaction Conference, OzCHI '12*, pp. 295–298, New York, NY, USA, 2012. Association for Computing Machinery.
- [6] Youli Chang, Sehi L'Yi, Kyle Koh, and Jinwook Seo. *Understanding Users' Touch Behavior on Large Mobile Touch-Screens and Assisted Targeting by Tilting Gesture*, pp. 1499–1508. Association for Computing Machinery, New York, NY, USA, 2015.
- [7] Christian Corsten, Marcel Lahaye, Jan Borchers, and Simon Voelker. Forceray: Extending thumb reach via force input stabilizes device grip for mobile touch input. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, CHI '19*, pp. 1–12, New York, NY, USA, 2019. Association for Computing Machinery.
- [8] Mohamed Khamis, Mariam Hassib, Emanuel von Zezschwitz, Andreas Bulling, and Florian Alt. Gaze-touchpin: 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.

- [9] Morten Lund Dybdal, Javier San Agustin, and John Paulin Hansen. Gaze input for mobile devices by dwell and gestures. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, ETRA '12, pp. 225–228, New York, NY, USA, 2012. Association for Computing Machinery.
- [10] Sophie Stellmach and Raimund Dachsel. Investigating gaze-supported multimodal pan and zoom. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, ETRA '12, pp. 357–360, New York, NY, USA, 2012. Association for Computing Machinery.
- [11] Emiliano Miluzzo, Tianyu Wang, and Andrew T. Campbell. Eyephone: Activating mobile phones with your eyes. In *Proceedings of the Second ACM SIGCOMM Workshop on Networking, Systems, and Applications on Mobile Handhelds*, MobiHeld '10, pp. 15–20, New York, NY, USA, 2010. Association for Computing Machinery.
- [12] 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.
- [13] 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.
- [14] Megan D Wiedbusch and Roger Azevedo. Modeling metacomprehension monitoring accuracy with eye gaze on informational content in a multimedia learning environment. In *ACM Symposium on Eye Tracking Research and Applications*, ETRA '20 Full Papers, New York, NY, USA, 2020. Association for Computing Machinery.
- [15] Tanja Blascheck and Bonita Sharif. Visually analyzing eye movements on natural language texts and source code snippets. In *Proceedings of the 11th ACM Symposium on Eye Tracking Research & Applications*, ETRA '19, New York, NY, USA, 2019. Association for Computing Machinery.
- [16] Alexandra Papoutsaki, Aaron Gokaslan, James Tompkin, Yuze He, and Jeff Huang. The eye of the typer: A benchmark and analysis of gaze behavior during typing. In *Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications*, ETRA '18, New York, NY, USA, 2018. Association for Computing Machinery.
- [17] Michael Xuelin Huang, Jiajia Li, Grace Ngai, and Hong Va Leong. Stressclick: Sensing stress from gaze-click patterns. In *Proceedings of the 24th ACM International Conference on Multimedia*, MM '16, pp. 1395–1404, New York, NY, USA, 2016. Association for Computing Machinery.
- [18] Kamalpreet Singh Saluja, JeevithaShree Dv, Somnath Arjun, Pradipta Biswas, and Teena Paul. Analyzing eye gaze of users with learning disability. In *Proceedings of the 2019 3rd International Conference on Graphics and Signal Processing*, ICGSP '19, pp. 95–99, New York, NY, USA, 2019. Association for Computing Machinery.

- [19] 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.
- [20] Thomas J. Grindinger, Andrew T. Duchowski, and Pilar Orero. Differentiating aggregate gaze distributions. In *Proceedings of the ACM SIGGRAPH Symposium on Applied Perception in Graphics and Visualization*, APGV '11, p. 110, New York, NY, USA, 2011. Association for Computing Machinery.
- [21] Nora Castner, Thomas C Kuebler, Katharina Scheiter, Juliane Richter, Therese Eder, Fabian Huettig, Constanze Keutel, and Enkelejda Kasneci. Deep semantic gaze embedding and scanpath comparison for expertise classification during opt viewing. In *ACM Symposium on Eye Tracking Research and Applications*, ETRA '20 Full Papers, New York, NY, USA, 2020. Association for Computing Machinery.
- [22] Stefanie Mueller. Inferring target locations from gaze data: A smartphone study. In *Proceedings of the 11th ACM Symposium on Eye Tracking Research & Applications*, ETRA '19, New York, NY, USA, 2019. Association for Computing Machinery.
- [23] Pragma Kar, Samiran Chattopadhyay, and Sandip Chakraborty. Gestatten: Estimation of user's attention in mobile moocs from eye gaze and gaze gesture tracking. *Proc. ACM Hum.-Comput. Interact.*, Vol. 4, No. EICS, June 2020.
- [24] Dmitry Lagun, Chih-Hung Hsieh, Dale Webster, and Vidhya Navalpakkam. Towards better measurement of attention and satisfaction in mobile search. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '14, pp. 113–122, New York, NY, USA, 2014. Association for Computing Machinery.
- [25] Shahram Jalaliniya and Diako Mardanbegi. *EyeGrip: Detecting Targets in a Series of Uni-Directional Moving Objects Using Optokinetic Nystagmus Eye Movements*, pp. 5801–5811. Association for Computing Machinery, New York, NY, USA, 2016.
- [26] Ioannis Giannopoulos, Peter Kiefer, and Martin Raubal. Geogazemarks: Providing gaze history for the orientation on small display maps. In *Proceedings of the 14th ACM International Conference on Multimodal Interaction*, ICMI '12, pp. 165–172, New York, NY, USA, 2012. Association for Computing Machinery.