

タッチ操作ログに基づいた操作形態推定手法

平部 裕子^{1,a)} 諏訪 博彦¹ 荒川 豊¹ 安本 慶一¹

概要：ユーザがどのようにスマートフォンを持ち、どの指で操作しているかという情報（操作形態）がわかれば、様々な新しい応用、例えば、操作形態に適した UI（ユーザインターフェース）への動的切替えや、ながらコンテキスト（スマートフォン操作時に行われている別行動、歩行、喫煙、飲酒など）の推定など、への道が拓ける可能性がある。本論文では、タッチ操作によって出力されるログに着目し、タッチやスワイプなどの操作領域や頻度からどのような操作形態かを推定する方法を提案する。そのために、Android 上でのあらゆるタッチ操作を取得可能な TouchAnalyzer を開発するとともに、そのログデータから代表的な 8 通りの操作形態を推定する機械学習に基づくアルゴリズムを提案する。16 名の実験参加者による評価実験を通じて、提案手法によって操作形態を概ね 90% の F 値で推定できることがわかった。

キーワード：タッチ操作ログ、スマートフォン、操作形態推定、機械学習

YUKO HIRABE^{1,a)} HIROHIKO SUWA¹ YUTAKA ARAKAWA¹ KEIICHI YASUMOTO¹

1. はじめに

近年、子供から老人まであらゆる世代がスマートフォンを利用するようになってきている。多様化するユーザに対して、適切なサービス、適切な広告、そして適切な UI (User Interface) を提供することが重要になっている。そのような機能を提供するためには、ユーザの状態（コンテキスト）やユーザの属性（プロフィール）を何らかの方法でスマートフォンが認識する必要がある。コンテキストに関しては、スマートフォンに内蔵されたモーションセンサを用いることで、動作を検出し歩行時や着座時に応じた機能提供^{*1*2} [1] といった試みがなされている。また、位置情報を用いて、位置に応じたサービスを提供するといったサービス^{*3*4*5} [2] も多数提案されている。一方、年齢層や性別、利き手、操作スキルなどのユーザの属性に関して、スマートフォン上で認識を試みた研究はほとんど先例がない。著者らは、スマートフォンの操作状況に基づくユーザ

プロファイリングを実現したいと考えており、本論文では特にタッチ操作ログに基づいた操作形態分析手法を提案する。操作形態とは、どのようにスマートフォンを持ち、どの指で操作しているかという情報である。

操作形態を認識する研究として、Goel らは GripSense [3] を提案している。この論文ではスマートフォンの加速度センサやジャイロセンサ、振動センサを使って、タッチパネルへの入力時に起こる端末の傾きや振動、親指のスワイプ、タッチポイントの接地面積の違いを各被験者ごとに計測し、端末を操作する指（親指か人差し指）と左右の手、端末が机の上に置かれているか、といった情報を含む操作形態認識をヒューリスティックに行っている。しかしながら、Gripsense は独自のアプリケーション上で動いているため、単一アプリ内でしかデータ取得ができない。そのためアプリケーションに依らず端末利用者の操作情報を把握し、統合的に操作形態を識別することは難しい。また QWERTY キーボードから得られる入力情報を利用しているため、日本語入力によく利用されるフリックキーボードや UI が異なるアプリケーションにおいては対応できない。他の手法としては、静電タッチセンサ [4, 14] や Grip センサを用いて認識する手法 [6] がある。これらの手法に、加速度センサを組み合わせて認識を試みる研究 [7] も存在する。また、端末カバーに光センサを多数組み込むことで操作形態を認識している研究 [8] も存在するが、いずれも追加のセンサ

¹ 奈良先端科学技術大学院大学
NAIST, Takayamamachi, Ikomashi Nara 630-0192, Japan
^{a)} hirabe.yuko.ho2@is.naist.jp
^{*1} https://www.nttdocomo.co.jp/service/anshin_mode/
^{*2} <http://www.au.kddi.com/mobile/service/smartphone/safety/aruki/-sumaho/>
^{*3} <http://www.sharp.co.jp/products/sh01g/service/emopa/>
^{*4} <https://www.cyberagent.co.jp/newsinfo/info/detail/id=11495>
^{*5} <https://www.fout.co.jp/pr20160107/>

が必要となり、スマートフォン単体で利用できない。

一方で、操作形態を認識する研究には文字入力を減らすことを目的としたものが多い。例えば、TouchML [9] は、長期間にわたってスマートフォンの持ち方を観測し、データ収集を行ったのち、タッチエラーの統計的パターン情報を用いて機械学習によって、対象のタッチが基準値からどれほどずれているかをタッチオフセットモデルとして構築することで入力ミスを低減している。また文字入力時に GripSense から推定した操作形態情報とキー入力を機械学習にかけることで、入力ミスを改善するシステム ContextType [10] も提案されている。これらの研究は、タップ操作による文字入力に注力しており、スワイプやマルチタッチといったスマートフォンならではの、かつ、アプリケーションによって変化する操作は考慮されていない。

本論文では、1) センサの追加や長期観測を必要としない、2) アプリケーション非依存である、3) ユーザの負担が少ない、の3つの要求を満たす操作形態認識手法を提案する。提案手法は、タッチ操作によって出力されるログに着目し、タッチやスワイプなどの操作領域や頻度からどのような操作形態かを推定する。そのために、Android 上でのあらゆるタッチ操作を取得可能な TouchAnalyzer を開発するとともに、そのログデータから代表的な8通りの持ち方を推定するアルゴリズムを提案する。アルゴリズムには、決定木、ランダムフォレストの2種類の機械学習アルゴリズムを使用しそれぞれの認識精度を評価する。一般的なユーザのスマートフォンの使い方を観察したところ、持ち方によって、タッチ開始/終了位置、タッチ頻度やスワイプ軌跡の曲がり具合が異なることがわかった。また、関連研究でも指摘されている通り [11]、手のサイズが操作に大きな影響を与えることがわかった。そこで、タッチ開始/終了位置の x 軸および y 軸、X 軸のタッチ頻度、Y 軸のタッチ頻度などのタッチ情報と手のサイズを特徴量として使用することとした。16 名の実験参加者から得られたログに基づいて、操作形態認識を行った結果、ランダムフォレストを用いた際に概ね 90% の F 値で各操作形態を識別可能であることがわかった。

2. タッチ操作ログ分析に基づく操作形態推定手法の提案

2.1 操作形態の定義

本研究ではスマートフォンの操作形態を、スマートフォンを支える手(支える手)、およびスマートフォンの画面を操作する手と指(操作する手、操作する指)の3要素の組み合わせ(以下、操作形態と記述)とし、3要素が取りうる状態を以下の様に定義する。

- 支える手 { 右, 左, 無し }
- 操作する手 { 右, 左 }
- 操作する指 { 親指, その他の指(主に人差し指) }

	支持		操作		
	手	手	指		
①	右	右	親	→	
2			その他		
3	左	右	親	→	
④			その他		
⑤	無し(机上)	右	親	→	
⑥			その他		
⑦	左	左	親	→	
8			その他		
9	右	左	親	→	
⑩			その他		
⑪	無し(机上)	左	親	→	
⑫			その他		

図 1: 操作形態の種類

ここで、支える手がない場合とは、スマートフォンが机の上などに置かれている状況のことを指す。操作する指に関して、今回はユーザの指の使用頻度を考慮して親指とその他の指(主に人差し指を想定しているが、人によって中指などを使用する人もいる)とした。

定義に基づいて組み合わせを考えると12の操作形態が考えられるが、右手で支え、右手の人差し指等で操作するという形態は一般的とは言い難い。また、左手で支え、右手で操作する場合、人差し指等で操作するのが一般的と考える。中には、親指等で操作するユーザが存在することは確認されているが、本研究ではその場合でも人差し指で操作したこととして分析する。結果として、端末操作者は図1に示すとおり、8種類の操作形態のうちいずれかでスマートフォンを操作することとなる。図1では例として、支持手がある場合に起こる4つの操作形態に関し、イラスト化している。残り4つはイラスト化した操作形態が机に置かれた状態で起こる形態を指す。本研究ではこの8形態を推定対象とする。

2.2 要件定義

本研究では、追加のセンサなしに、アプリケーションを横断して端末非依存にスマートフォンのタッチ操作ログを取得、利用して、短いデータ収集期間で8つの操作形態を認識することを目的とし、タッチ操作ログ分析に基づく操作形態認識手法を提案する。タッチ操作は単一のアプリケーション内であれば、特殊なスクリプトを埋め込む事で簡単に取得できる。しかし Android に代表される OS では、セキュリティの観点から、アプリケーションを横断して情報を取得することは一般的に難しい。我々はアプリケーションに依らず情報を取得して統合的に操作形態を認識することを目的としているため、この手法が利用できない。よって、本手法の要件として以下の二つを設定する。

要件1 アプリ横断的かつ端末非依存な操作形態の認識

アプリ横断的かつ端末非依存な操作形態の認識をするためには、タッチ操作情報をロギングする特殊なスクリプト

を全アプリに埋め込む方法が考えられる。しかしこの手法は現実的とはいえない。一方でタッチ操作情報をロギングする専用アプリケーションを作成し、絶えず他のアプリを監視するといった方法も考えられる。しかし Android に代表される OS (Operation System) は、サンドボックスと呼ばれるセキュリティの観点から各アプリが強制的に分離、独立しているため、それぞれのアプリケーションが所有するデータを相互に見ることは、root と呼ばれる管理者権限を取得しない限り、不可能である。

これらのことから、アプリケーションを横断してタッチ操作情報を取得するには、スマートフォンの仕組み上、OS が出力する情報を取得するしかない。しかし OS が出力するデバイスログは、低レベルの文字の羅列であるうえ、端末ごとにフォーマットも異なる。例えばスワイプ動作を点の塊として単にログを出力しているため、取得したデバイスログから、それらのログがシングル(またはマルチ)タッチなのかシングル(マルチ)スワイプか、といったタッチ操作の挙動内容や、挙動頻度や速度、操作範囲などの高レベルの情報に変換する必要がある。

要件 1 に対応するために、データ収集システム TouchAnalyzer を開発した(詳細は、3.3 節で述べる)。

要件 2 ユーザや端末の負荷が少ないシンプルな認識手法

操作形態を認識するにあたり、ユーザや端末に負荷をかけることは望ましくない。例えば、学習データの作成のために長期間のデータ収集を必要となったり、各ユーザが事前に正解データを登録する作業が必要となれば、ユーザに負荷がかかり満足度は低下する。

また、操作形態を認識するために、多くのセンサを利用しデータ収集する場合、データ処理量・消費電力の観点から端末の負荷が大きくなり、ユーザの利用可能時間が減少する。結果として、ユーザの満足度は低下する。同様に、複雑な計算をさせることはより多くの負荷を端末にかけることになるため、シンプルなルールで操作形態を認識することが求められる。

要件 2 に対応するために、推定モデル構築に用いる特徴量は、ユーザがスマートフォンを本来の目的のために操作した際に収集可能なデータに限定する。また、特徴量の数を極力減らし、推定モデルを構築するためのアルゴリズムもシンプルなものを検討する。推定プロセスについては、3.4 節で述べる。

2.3 TouchAnalyzer の開発

要件 1 に対応するため、OS が出力するタッチイベントのデバイスログを高レベルの情報(シングル・マルチタッチ、スワイプなど)に変換し、タッチ操作の挙動情報として識別するシステム TouchAnalyzer を新たに開発した。TouchAnalyzer は PC 上で動作するシステムであり、対象

端末とシステムを USB 経由で接続する構成となっている。開発システムは Android SDK platform-tools に含まれる、Android Debug Bridge (以下 adb と記載) と python を用いて実現されている。

TouchAnalyzer は、/dev/input などのデバイスログに出力されるタッチイベントログを、adb 経由で絶えず監視して常時取得すると同時に、foreground (ユーザが操作している) アプリケーションログを取得する。

本システムは、機種ごとに異なるタッチイベントの出力先や時刻フォーマットを自動的に認識し、タッチイベントログをリアルタイムに収集・解析し、有意な情報として抽出する。これにより、ユーザのタッチ操作(シングル・マルチタッチ、シングル・マルチスワイプ、ピンチ、ローテート)のログを収集することが可能となる。また TouchAnalyzer を改良し、PC と対象端末を USB ケーブルで接続すること無く、ログデータをサーバへ収集可能な TouchAnalyzer アプリケーションを開発している。本研究では、主にこれらのシステムから得られたタッチ操作ログを用いて操作形態推定手法を構築する。

2.4 推定プロセス

推定プロセスの概要を以下に示す(図 2)。今回、推定したい操作形態は 8 形態である。この 8 形態は、手で持つ場合と、卓上に置く場合とに大別できる。手で持つ場合は、多少なりともスマートフォンが動くことになる。一方、卓上に置く場合は、スマートフォンはほとんど動くことなく静止している。この二つを分類するには、スマートフォンに内蔵されている加速度を計測すれば容易に判定できる。そこで、まず加速度を用いて 8 形態を 4 形態ずつに分ける。なお、加速度で分類する部分については、極めて単純な仕組みのため、本論文では取り扱わない。次に、それぞれの 4 形態を TouchAnalyzer から得られるタッチ操作ログ等に基づいて推定する。推定モデルの構築には、統計手法および機械学習を用いる。なお、要件 2 に対応するためには、使用する特徴量の種類は少ないほうが望ましい。また、推定モデル構築のための数理モデルもシンプルなものが望ましい。具体的な特徴量および数理モデルについては、4 章で述べる。

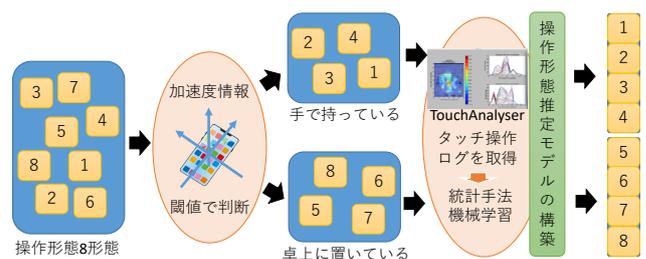


図 2: 推定プロセス

3. 操作形態推定手法の実現

本章では、タッチ操作ログ分析に基づく操作形態推定手法を実現するための手順として、実験環境、特徴量、および評価に用いる機械学習アルゴリズムのに関して詳述する。

3.1 実験の環境と内容

タッチ操作ログを収集するため、実験参加 16 名 (男性 9 名, 女性 7 名) に協力を依頼した。被験者の年齢は 23 歳から 28 歳である。

端末の大きさを考慮するため、実験端末には Nexus 5 および Nexus 6 を各 5 台ずつ利用した。それぞれの実験端末にはデータ収集アプリケーションが事前にインストールされており、アプリを起動した状態で操作する事で被験者のログデータを収集した。

実験参加者は図 1 に示した 8 つの持ち方に対し 3 分間ずつ、計 1 時間 30 分の操作を行った。実験中に操作するアプリケーションは、Yahoo!ニュース^{*6}を指定した。Yahoo!ニュースは利用者も多く、通常よく使用される一般的なサイトと考え採用している。各参加者は、3 分間アプリケーションを操作する際に操作形態を固定する以外、どのような指示も受けておらず、アプリケーションを自由に操作する。また実験前に、参加者の手の大きさを計測した。

収集ログデータのうち、評価データに、Nexus 5 に関して 10 名、Nexus 6 に対して 10 名のデータを利用した。その理由は、被験者の操作ミスによりデータ収集アプリケーションが起動しておらず、データ収集が出来ていなかったユーザがいたためであり、これらの被験者データは評価対象から除外している。

3.2 特徴量

本研究では操作形態推定モデルを構築するために、3 章の要件 2 で示した、ユーザや端末の不可が少ないシンプルな認識手法を実現可能な特徴量を利用する。推定モデル構築に利用する特徴量はユーザのスマートフォン操作から得られるものとし、スワイプ操作の情報を利用した。スワイプ操作を選択した理由は、タッチ操作と比較してより多くの特徴量が得られると想定したためである。例えばタッチ操作では、操作軌跡の移動量やスワイプが描く弧の方向など、操作形態に関連しうる特徴量を利用できない。評価で利用する具体的な特徴量を、以下に挙げる。

- 始点 { x 座標, y 座標 }
- 終点 { x 座標, y 座標 }
- 始点から終点への { x 座標の移動量, y 座標の移動量 }
- スワイプが描く弧の方向
- 出現最頻領域の {x 軸方向の分割番号, y 軸方向の分割

番号 }

ここで、出現最頻領域について述べる。操作形態推定に利用可能な特徴量を考慮した時、操作形態に依存して、スマートフォンのタッチパネル上で最もよく触られる領域 (出現最頻領域と定義) が出現すると考えられる。この領域を抽出するため、スマートフォンの画面領域を経験的に x 軸方向および y 軸方向へ、それぞれ 12 分割, 16 分割し、各スワイプ操作ごとに x 方向, y 方向への分割番号を導出した。

列挙した 9 つの特徴量をスワイプが行われる度に抜き出し、全ての参加者情報をマージして、データセット 1 (今後、データ 1 と記述) を作成した。

一方で我々の先行研究 [15] で、操作している手の指の長さが操作形態の判定に利用できる可能性が有ることが分かっていた。そのためデータ 1 に、操作している手の指の長さ (親指, 中指) を特徴量として追加し、データセット 2 (以降、データ 2 と記述) を作成した。

データ 1 およびデータ 2 のそれぞれに対して、機械学習を適応し、leave one person out cross-validation で提案手法を評価する。評価するにあたって、利用する機械学習アルゴリズムを検討、選択する。

3.3 アルゴリズムの検討

3 章の要件 2 に対応するため、端末の負荷が少ないシンプルな数理モデルを選択し、推定モデル構築を行なう必要がある。今回は比較的シンプルなモデルとして決定木 (decision tree) を選択した。また高精度で推定できるといわれている代表的なアルゴリズムとして、RandomForest も合わせて利用し、これら 2 つの数理モデルに基づいて推定モデルを構築し、比較・評価する。各モデルの特徴は次のとおりである。

(1) 決定木 (decision tree)

決定木は、属性値の条件によりノードを分岐させて木構造の分類器を生成する伝統的な機械学習手法の 1 つである。決定木学習の特徴として、if-then ルールに基づいて木を生成していくため、他の手法と比べて分類過程の可読性が高いという利点がある。本研究では、決定木学習の代表的なアルゴリズムである C4.5 決定木を用いた。

(2) RF (Random Forests)

RF はアンサンブル学習を用いた機械学習アルゴリズムの 1 つである。複数の決定木を弱分類器として使用し、その結果を統合して結果を得る。他の分類器に比べ性能が高く、判定速度も高いと言われているが、決定木やロジスティック回帰に比べると、計算量は一般的に大きい。

^{*6} <https://news.yahoo.co.jp/promo/app/yjnews/>

4. 評価結果

3章で述べたデータ1(支持手の指の長さ情報が、無いデータセット)とデータ2(有るデータセット)に対して、機械学習アルゴリズムの決定木(decision tree), RF(Random Forests)で推定モデルを構築し評価した結果を詳述する。

総合的な評価結果を表3に示す。Nexus5, および Nexus6, どちらも同様の傾向が見られたため 4.1 節では、特に Nexus5 についての評価結果を記載する。

また 8 種類の操作形態について、Nexus5 および Nexus6 それぞれに対し上記のアルゴリズムから構築した推定モデルによる推定結果は、4.2 節で述べる。

表 1: データ 1 に対する nexus5 での評価結果

(a) 決定木 (Decision Tree)

Class	Precision	Recall	F 値
1	0.602	0.606	0.581
2	0.594	0.598	0.584
5	0.608	0.575	0.545
6	0.432	0.434	0.429
Average	0.562	0.563	0.543

(b) RF (Random Forests)

Class	Precision	Recall	F 値
1	0.683	0.692	0.678
2	0.636	0.631	0.614
5	0.682	0.627	0.586
6	0.467	0.463	0.444
Average	0.602	0.596	0.576

表 2: データ 2 に対する nexus5 での評価結果

(a) 決定木 (Decision Tree)

Class	Precision	Recall	F 値
1	0.979	0.979	0.979
2	0.977	0.976	0.976
5	0.991	0.988	0.989
6	0.690	0.736	0.670
Average	0.898	0.923	0.899

(b) RF (Random Forests)

Class	Precision	Recall	F 値
1	0.989	0.989	0.989
2	0.971	0.972	0.971
5	0.995	0.997	0.996
6	0.928	0.887	0.889
Average	0.967	0.969	0.965

4.1 Nexus5 における操作形態推定結果

データ 1, およびデータ 2 を用いた時, 実験端末である Nexus5 に関して評価した結果を表 1, 2 に示す。各表のクラス番号 1, 2, 5, 6 は, 表 1 での 1, 4, 7, 10, にそれぞれ対応しており, 各々のクラスの Precision, Recall, F 値を記載している。

加えて評価に leave one person out cross-validation を採用している。すなわち各被験者データをテストした結果から得られた, Precision, Recall, F 値を単純平均したものを, 表中に Average で示している。

データ 1 に対する推定結果

表 1 の (a) 決定木 (Decision Tree), (b)RF(Random Forests), それぞれに関して平均 Precision, Recall, F 値をみると, 最も良かったのは RF アルゴリズムの推定モデルであり, Precision が 60.2%, Recall が 59.6%, F 値が 57.6%であった。

全体的な結果として, スマートフォン操作から得られる特徴量, 9 つのみでは精度として不十分な結果となった。RF アルゴリズムで構築した推定モデルで, 実験参加者 1 名のテストデータから得られた混同行列を確認すると, 右手か左手の識別は 100%推定できるものの, 親指もしくはその他の指での操作判定がほとんど出来ていないため, 精度が悪くなったと考えられる。

データ 2 に対する推定結果

データ 2 に関しても同様に, 表 2 の (a) 決定木 (decision tree), (b)RF(Random Forests) について順に平均 Precision, Recall, F 値を確認すると, 推定結果が最も良かったのは RF アルゴリズムの推定モデルで, 平均の Precision, Recall, F 値はそれぞれ, 96.7%, 96.9%, 96.5%であった。決定木の各平均精度の値は, 決定木での推定モデルでは, 89.8%, 92.3%, 89.9%であった。

全体的な結果として, データ 1 に対する推定結果と比べ飛躍的に推定精度が向上した。支持手の指の長さは, 親指もしくはその他の指での操作判定に有効であることが分かる。

総合推定結果

Nexus5, Nexus6 における操作形態推定結果に対する平均の F 値を表 3 に示す。表のクラス番号 1, 2, 5, 6 は上記結果と同様に対応している。一方で, 表クラス番号 3, 4, 7, 8 は図 1 での 5, 6, 11, 12, に合致している。

推定結果について Nexus5, 6 に対し, クラス 1, 2, 5, 6 の推定精度を確認する。まず Nexus5 に関して見てみると, データ 1 の時は RF アルゴリズムの場合が最も高く, 57.6%であった。データ 2 に対しては, RF が最も高く, 96.5%であった。一方でクラス 3, 4, 7, 8 について見ると, データ 1, データ 2 で高精度だったのは共に RF であり, 53.7%, 95.2%であった。

Nexus6 に関して表 3 を確認する。クラス 1, 2, 5, 6 の推定

精度をみると、データ 1, に対する最高精度を出したのは RF で構築した推定モデルで、各精度は 47.9%であった。データ 2 に対しては決定木で構築した推定モデルであった。各精度は 84.2%となった。クラス 3,4,7,8 に関しては RF がデータ 1, データ 2 に対して最も良い推定精度であった。その時の F 値は、それぞれ 52.9%, 85.0%であった。

表 3: 総合評価：すべての組み合わせの F 値一覧

機種		Nexus 5		Nexus 6	
		決定木	RF	決定木	RF
Class 1~4	データ 1	0.543	0.576	0.447	0.479
	データ 2	0.899	0.965	0.844	0.842
Class 5~8	データ 1	0.522	0.537	0.506	0.529
	データ 2	0.919	0.952	0.847	0.850

5. まとめ

本論文では、タッチ操作のログから、ユーザによるスマートフォンでの操作形態（支持している手、操作している手、操作している指の組）を推定する手法を提案した。スマートフォン OS が出力するタッチ操作のログが低レベルかつ機種依存であるという課題を、機種に依存しない高レベルのタッチ操作（シングル・マルチタッチ、スワイプ、ピンチ、ローテート）を出力可能な TouchAnalyzer を開発し解決した。その上で、高レベルのタッチ操作が発生したスクリーン上の場所および頻度を特徴とした操作形態を推定するモデルを様々な機械学習法により構築し評価した。同じ操作形態であっても個人差のため、精度が悪くなるという課題を、操作する指の長さを特徴量として追加することで、解決をはかった。最終的に、16 名の実験参加者による 8 種類の操作形態をランダムフォレストアルゴリズムを用いることにより、F 値約 90%で推定できた。今後は、操作形態の推定結果を用いたアプリケーションの提供を通じて、本手法の有効性を評価して行くことを検討している。謝辞 本研究は、JSPS 科研費 (26700007) の助成を受けたものである。

参考文献

- [1] Ho, J. and Intille, S. S.: Using context-aware computing to reduce the perceived burden of interruptions from mobile devices, Proceedings of the SIGCHI conference on Human factors in computing systems, ACM, pp. 909 918 (2005).
- [2] Partridge, K. and Begole, B.: Activity-based advertising techniques and challenges, Proc. Workshop on Pervasive Advertising (2009).
- [3] Goel, M., Wobbrock, J. and Patel, S.: GripSense: Using Built-in Sensors to Detect Hand Posture and Pressure on Commodity Mobile Phones, Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology, UIST ' 12, New York, NY, USA, ACM, pp. 545554 (online), DOI: 10.1145/2380116.2380184 (2012).
- [4] Harrison, B. L., Fishkin, K. P., Gujar, A., Mochon, C. and Want, R.: Squeeze Me, Hold Me, Tilt Me! An Exploration of Manipulative User Interfaces, Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI ' 98, New York, NY, USA, ACM Press/Addison-Wesley Publishing Co., pp. 1724 (on-line), DOI: 10.1145/274644.274647 (1998).
- [5] Kim, K.-E., Chang, W., Cho, S.-J., Shim, J., Lee, H., Park, J., Lee, Y. and Kim, S.: Hand Grip Pattern Recognition for Mobile User Interfaces, Proceedings of the 18th Conference on Innovative Applications of Artificial Intelligence - Volume 2, IAAI ' 06, AAAI Press, pp. 17891794 (online), available from <http://dl.acm.org/citation.cfm?id=1597122.1597138> (2006).
- [6] 塚本昌克, 樋口雄太, 岡田 隆: 把持特徴を利用したユーザインタフェースの開発: Grip UI, NTT DoCoMo テクニカル・ジャーナル, Vol. 21, No. 3, pp. 1723 (2013).
- [7] Taylor, B. T. and Bove, Jr., V. M.: Graspables: Grasp-recognition As a User Interface, Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI ' 09, New York, NY, USA, ACM, pp. 917 926 (online), DOI: 10.1145/1518701.1518842 (2009).
- [8] Cheng, L.-P., Lee, M. H., Wu, C.-Y., Hsiao, F.-I., Liu, Y.-T., Liang, H.-S., Chiu, Y.-C., Lee, M.-S. and Chen, M. Y.: IrotateGrasp: Automatic Screen Rotation Based on Grasp of Mobile Devices, Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI ' 13, New York, NY, USA, ACM, pp. 30513054 (on-line), DOI: 10.1145/2470654.2481424 (2013).
- [9] Buschek, D. and Alt, F.: TouchML: A Machine Learning Toolkit for Modeling Spatial Touch Targeting Behaviour, Proceedings of the 20th International Conference on Intelligent User Interfaces, IUI ' 15, New York, NY, USA, ACM, pp. 110114 (online), DOI: 10.1145/2678025.2701381 (2015).
- [10] Goel, M., Jansen, A., Mandel, T., Patel, S. N. and Wobbrock, J. O.: ContextType: Using Hand Posture Information to Improve Mobile Touch Screen Text Entry, Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI ' 13, New York, NY, USA, ACM, pp. 27952798 (online), DOI: 10.1145/2470654.2481386 (2013).
- [11] 河内まき子: AIST 日本人の手の寸法データ, 入手先 (<https://www.dh.aist.go.jp/database/hand/index.html>) (2012).
- [12] Lochtefeld, M., Schardt, P., Kru ger, A. and Boring, S.: Detecting users handedness for ergonomic adaptation of mobile user interfaces, Proceedings of the 14th International Conference on Mobile and Ubiquitous Multimedia, ACM, pp. 245249 (2015).
- [13] Buschek, D., Rogers, S. and Murray-Smith, R.: User-specific Touch Models in a Cross-device Context, Proceedings of the 15th International Conference on Human-computer Interaction with Mobile Devices and Services, MobileHCI ' 13, New York, NY, USA, ACM, pp. 382391 (online), DOI: 10.1145/2493190.2493206 (2013).
- [14] Hirabe, Y., Arakawa, Y. and Yasumoto, K.: Logging All the Touch Operations on Android, The 7th International Conference on Mobile Computing and Ubiquitous Networking (ICMU 2014), pp. 9394 (2014).
- [15] 平部 裕子, 津田 麻衣, 荒川 豊, 安本 慶一: タッチ操作の挙動に基づくユーザプロファイル推定手法の提案, 情報処理学会研究報告, Vol.2014-UBI-41, No.20, pp. 1-6 (2014-03)