

PredicTaps: シングルタップとダブルタップの識別予測に基づくシングルタップ操作のレイテンシ低減手法

池松 香^{1,a)} 山中 祥太^{1,b)} 坪内 孝太^{1,c)}

概要: タッチサーフェス入力操作において、シングルタップとダブルタップを同一のコンポーネントに対し実行可能な設計とする場合、シングルタップの判定所要時間が長くなるという課題がある。一般に、システムは検出されたタップがシングルタップまたはダブルタップの一回目のタップのどちらであるかを判定する際に、後続する可能性のあるタップ、すなわちダブルタップの二回目のタップを受け付ける時間（一般に、350–500ms 程度）を設ける。シングルタップにおいてはこの受付時間がレイテンシとなる。本研究では、一回目のタッチ入力終了時点で、該当のタップがシングルタップとダブルタップのどちらの入力であるかを機械学習により判定し、レイテンシを低減する手法として“PredicTaps”を提案する。PredicTapsは検出されたタップがシングルタップであるか否かの予測に対するスコアに基づきシングルタップイベントを迅速に実行するか、または後続する可能性のあるタップを待機するかを決定する。これにより、シングルタップとダブルタップの誤認識による操作性の低下を防ぎつつシングルタップ処理の高速化を目指す。本論文では、提案手法の実現可能性について調査し報告する。

1. はじめに

タッチパッドやスマートフォン、タブレット PC などのタッチサーフェスは広範に普及し、タッチ入力は今最も一般的な入力操作となった。タッチ入力においてシングルタップ及びダブルタップは基本的な操作の一つである。Graphical User Interface (GUI) の多くのアプリケーションにおいては、シングルタップとダブルタップの両方を実行可能なコンポーネントが存在する。これにはブックリーダー (電子書籍)、地図アプリケーション、イメージビューワあるいは web ブラウザなどのアプリケーションが該当する。例示すると、地図アプリケーションの Google Maps ^{*1}においては、シングルタップに地図上のコンポーネントの選択操作を、ダブルタップに地図全体の拡大操作を割り当てている。

一般に、タッチサーフェスの入力領域は限られているため、マルチタップによる入力語彙の拡張は効果的であると考えられる。一方で、従来のシングルタップとダブルタップの判定方法においては、シングルタップの判定所要時間が長くなるという課題がある。これは、実行されたタップ

がシングルタップまたはダブルタップの一回目のタップのどちらであるかを判定するために、一定時間（主要な OS に採用されている閾値は 350–500 ms [10], [19], [31], 以下、ダブルタップ閾値）待機する。この待機時間内に再度タップが実行された場合、システムはダブルタップイベントを処理し (図 1 (B)), タップが実行されなかった場合にはシングルタップイベントの処理が行われる (図 1 (A))。ダブルタップイベントは二回目のタップが検出された直後に実行可能であるが、シングルタップイベントはダブルタップ閾値が経過するまで待機する必要がある。一般に、健常なユーザはシングルタップ動作を 350 ms 以内に完了することが可能であるため [8], [29], シングルタップ動作に要する時間とダブルタップ閾値との差異がシングルタップ操作におけるレイテンシとなる (以下、シングルタップレイテンシ)。

タッチ入力においてシングルタップは最も使用頻度の高い操作のひとつである。そのため、シングルタップレイテンシの短縮は重要な課題であると考えられる。また、連続したシングルタップを意図して実行する際に、二回のタップ実行間隔が短いとダブルタップとして誤認識されるという課題がある。例示すると、ブックリーダーにおいて「二ページ前に戻したい」場合には意図的にゆっくりとした動作で二回のタップを行い、ダブルタップとして誤認識されないよう慎重に操作する必要がある。こうした課題に対す

¹ ヤフー株式会社

a) k-ikematsu@acm.org

b) syamanak@yahoo-corp.jp

c) ktsubouc@yahoo-corp.jp

*1 <https://www.google.com/maps>

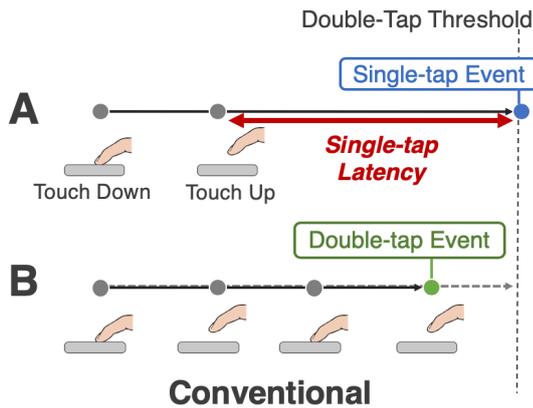
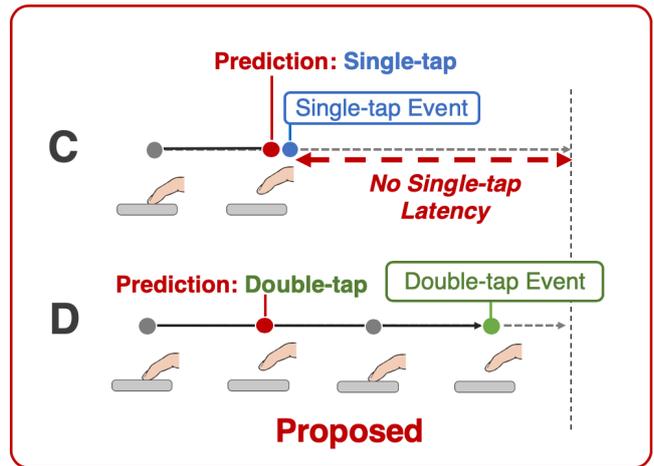


図 1 従来のタップ判定方法による処理: (A) シングルタップイベント及び (B) ダブルタップイベント. 提案手法による処理: (C) シングルタップであると予測された場合, 一回のタップ検出直後にシングルタップイベントを実行する, (D) ダブルタップであると予測された場合, 後続する可能性のあるタップを待機する.



るシンプルな対処法としてはダブルタップ閾値を短縮する方法が考えられる. この方法はシングルタップイベントを高速にするものの, より素早いダブルタップ動作を要求する処理となるため, ダブルタップ操作の成功率が低下する可能性がある [2]. また, 図 2 に示すように, タップが一回発生した時点で直ちにシングルタップイベントを実行することで, シングルタップレイテンシが発生しない実装を採用するアプリケーションもある. この実装は, ダブルタップイベント発生前にシングルタップイベントが発生してもダブルタップイベントに支障のない場合にのみ有効である. 例示すると, 多くのテキストエディタは図 2 の実装を採用し, シングルタップをキャレット移動に, ダブルタップをテキストの部分選択の操作に割り当てている. これは, ダブルタップイベントの直前にカーソル位置へキャレットが移動しても選択されるテキスト範囲への影響は無いためである. ブックリーダーなどのアプリケーションにおいてはシングルタップイベント実行有無によりダブルタップイベントの実行対象が変化するため, 上述の処理は適用が困難である.

上述の課題に対し, 我々は機械学習を利用したシングルタップレイテンシ低減手法である “PredicTaps” を提案する. PredicTaps は一回のタップが検出された際に計測されたタッチイベント関連のデータ (タッチ所要時間, 指とタッチサーフェス間の接触面積及びタッチ位置など) から, 検出されたタップがシングルタップであるか否かの予測に対するスコアを算出する. このスコアに応じ, シングルタップイベントを迅速に実行するか (図 1 (C)), または後続する可能性のあるタップを待機するか (図 1 (D)) を決定する. 前者の場合にはシングルタップレイテンシを削減し, 後者の場合は従来と同様のレイテンシを導入した処

理となる. 本論文では, 提案手法の実現可能性について調査し報告する. 本論文の貢献を以下に要約する.

- (1) シングルタップのレイテンシ低減を可能とする機械学習ベースの手法である PredicTaps を提案した.
- (2) タップのデータ収集を二種類の異なる実験環境下 (制御実験及び非制御実験) で行い, 識別のためのモデルを実装した.
- (3) 予測に対するスコアに基づき, PredicTaps によりどの程度のレイテンシ低減が可能であるかについて議論した.

2. 関連研究

2.1 タッチ入力操作におけるレイテンシ

タッチサーフェス操作固有の End-to-End のレイテンシ

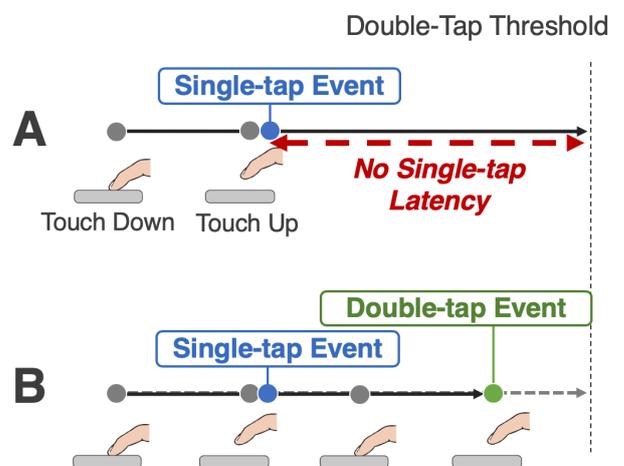


図 2 (A) テキストエディタなどにおけるシングルタップレイテンシの発生しない処理. (B) ダブルタップ実行時にも一回目のタップの時点でシングルタップイベントが発生する.

はユーザがタッチサーフェスに触れてから、操作に応じて画面出力が変化するまでの所要時間である。このレイテンシは操作性に影響することが知られている [7]。Ng ら [7] によると、レイテンシは主に次の三要素から発生する。(1) タッチ入力を検出する物理的なセンサ、(2) タッチイベントを処理し画面出力を生成するソフトウェア及び、(3) 画面出力のハードウェアである。本研究で扱うシングルタップレイテンシの発生源は (2) に該当する。市販のタッチサーフェスにおけるレイテンシはおよそ 50 ms から 200 ms の範囲である [7], [13]。また、ユーザが知覚可能な End-to-End のレイテンシは、フォームファクタやタスクの操作内容により異なることが知られている [7], [14], [15], [30]。Deber ら [15] はタッチパッドのカーソルを介した間接タッチ入力のフォームファクタにおいて、ドラッグ操作時は 55 ms から、タップ操作時は 96 ms からレイテンシが知覚可能であることを示した。また、スマートフォンやタブレット PC における直接タッチ入力については、ドラッグ操作時は 11 ms から、タップ操作時は 69 ms から知覚可能であることを示した。直接タッチの場合、間接タッチと比較しレイテンシはより知覚されやすくなる傾向にある。これは、ユーザの指と画面出力の物理的な位置関係の影響によるものである。すなわち、直接タッチの場合、操作指の直下に操作と対応する出力が表示されるため知覚されやすいことを示す [15]。以上のように、フォームファクタとタスクの組み合わせにより知覚可能な最小のレイテンシは異なるものの、市販のタッチサーフェスにおけるタッチ操作のレイテンシは知覚可能な範囲であり、かつ操作性に悪影響を及ぼす可能性がある [14]。よって、操作性向上のためにレイテンシ低減は望ましい目標であるといえる。

2.2 タッチ入力操作におけるレイテンシ低減手法

タッチ入力操作時のレイテンシ低減を目的とした手法が複数提案されている。センサやハードウェアの側面からレイテンシを低減するアプローチとしては、高速度カメラと高速プロジェクタを組み合わせて利用する手法 [1], [7] や、光学マーカーの利用 [26] により指の動きの検出または表示を高速化する手法が挙げられる。ソフトウェアベースのアプローチでは、直近のタッチ位置とその軌跡から、近い将来指が移動する可能性の高いタッチ位置を予測しインタラクションを行う Next-point Prediction の手法が挙げられる。Nancel らの分類 [20] によると、既存の Next-point Prediction は主にテイラー級数 [26]、カルマンフィルタ [32]、曲線近似 [6]、ニューラルネットワーク [16], [23], [27] 及びヒューリスティックな手法 [33] に基づくものである。これらの包括的な技術詳細については [20] を参照されたい。上述の手法は主にドラッグ操作時のレイテンシ低減を目的としている。本研究では、アプリケーションの仕様上発生するシングルタップレイテンシを対象とし、その低減

を目指す。また実用性を考慮し追加のハードウェアやセンサを用いない手法を検討した。

2.3 予測に基づくタッチ入力操作

直近のタッチイベント関連のデータからユーザの入力意図を予測し、操作を補助する手法が提案されている [3], [4], [12], [18], [24]。例示すると、Schwarz ら [3] は図形の配置や編集を行う際に、N-best の推定結果に基づいた操作候補を画面上へ継続的に表示し、操作をサポートするインタフェースを提案した。Buschek ら [24] は確率モデルに基づくタッチ GUI の framework として ProbUI を提案した。こうした研究群は、複雑なタッチジェスチャの混在により入力に不確実性があることを課題として扱っている。そして、過度に複雑な実装やヒューリスティックを予測に基づく処理に置き換え、効率的な操作を目指すものである。PredicTaps においても同様に、予測による処理を導入することでシンプルかつ実用的な処理を目指した。

3. PredicTaps

3.1 シングルタップとダブルタップの動作差異

一般に、人間が視覚などの刺激を受けてから、意図して手指の動作に反映させるまでには 260–290 ms 程度の時間を要する [25], [28]。ダブルタップ閾値が 350–500 ms 程度であることを考慮すると、シングルタップを意図して一回のタップを実行した後に「再度タップしてダブルタップ操作に変更する」という判断を行い、閾値時間内に二回目のタップを実行することは困難であると予想される。これは、ダブルタップ動作が完了するまでの所要時間が 200–300 ms 程度である [9] ことからも予想される。よって、ダブルタップは一回目のタップを行う前から、あらかじめ二回の素早いタップを実行すると決めている *pre-determined* な動作であると考えられる。そのため、ダブルタップの一回目のタップの際に、ユーザは指を画面へ接触させて離す動作をシングルタップ実行時と比較し素早く行うと考えられる。ダブルタップの一回目のタップとシングルタップの動作の差異は、タッチ所要時間や、指とタッチサーフェス間の接触面積といったタッチイベント関連のデータに現れると予想される。PredicTaps は、そうしたデータを用いてシングルタップとダブルタップの識別を行い、シングルタップレイテンシを低減する手法である。

3.2 処理概要

PredicTaps のシステムは、タッチサーフェスにおいてタッチダウンイベントが発生した際に、タッチイベント関連のデータを取得する。次に、タッチアップイベントが発生した時点で取得したデータを識別部に渡し、検出されたタップがシングルタップであるか、またはダブルタップの一回目のタップであるかを予測する。そして、予測結果が

表 1 PredicTaps における実際の入力と予測結果の組み合わせ毎の実行処理.

入力 \ 予測	Single-tap	Double-tap
Single-tap	(A) Single-Tap: レイテンシ削減	(B) Single-Tap: 従来処理と同様
Double-tap	(C) Single-Tap: 誤入力発生	(D) Double-Tap: 従来処理と同様

シングルタップであった場合には、シングルタップイベントを即座に実行し、レイテンシを削減する。また、ダブルタップであった場合には、従来の処理と同様に後続するタップが行われる可能性を考慮して一定時間待機する。

以上が PredicTaps の基本的な処理であるが、PredicTaps は機械学習による予測を基に処理を決定するため、誤った予測が発生することを考慮したシステム設計を行う必要がある。上述の予測と実際の入力の組み合わせ毎の実行処理について表 1 に示す。ダブルタップの偽陽性の場合（表 1 (B)）は、シングルタップレイテンシが発生した後にシングルタップイベントを実行する従来の処理方法と同様の処理になり、レイテンシの観点から性能劣化は生じない。一方で、シングルタップの偽陽性（表 1 (C)）は、意図しないシングルタップの実行であるため復帰操作が必要となり、また、ダブルタップ実行の難化につながる。よって、シングルタップの偽陽性のみが実際に対処すべき課題であるといえる。

3.3 予測に対するスコアに基づいた判定処理

前節の課題をふまえ、シングルタップの偽陽性発生率を低減させる処理を導入した。本処理では検出されたタップに対する予測のスコア（以下、スコア）に応じ、シングルタップレイテンシを削減するか否かを決定する。ロジスティック回帰におけるスコアとは、検出されたタップの特徴量から判断して、該当のタップがシングルタップである確からしさを推定する値である。この値は大きいほどシングルタップである可能性が高く、小さいほどダブルタップである可能性が高いことを表す。また、値が 0 付近であるほどシングルタップまたはダブルタップのどちらであるかの確度が低いことを表す。通常、スコアが 0 以上の場合にはシングルタップであると判定する（図 3 (B)）が、このスコアに対し任意の閾値を適用し（図 3 (A)）、確度の高いシングルタップのみをレイテンシ削減の対象とすれば、シングルタップの偽陽性を低減させることが可能である（図 3 (C)）。一方で、ダブルタップである可能性が高い場合、または、予測の信頼性が低い場合にはダブルタップ動作である可能性を考慮して待機する（図 3 (D)）。後者の処理はシングルタップレイテンシを導入する従来の判定手法と同様の処理であるため、レイテンシの観点から性能劣化は生じない。

スコアに対し閾値を適用することは、シングルタップか

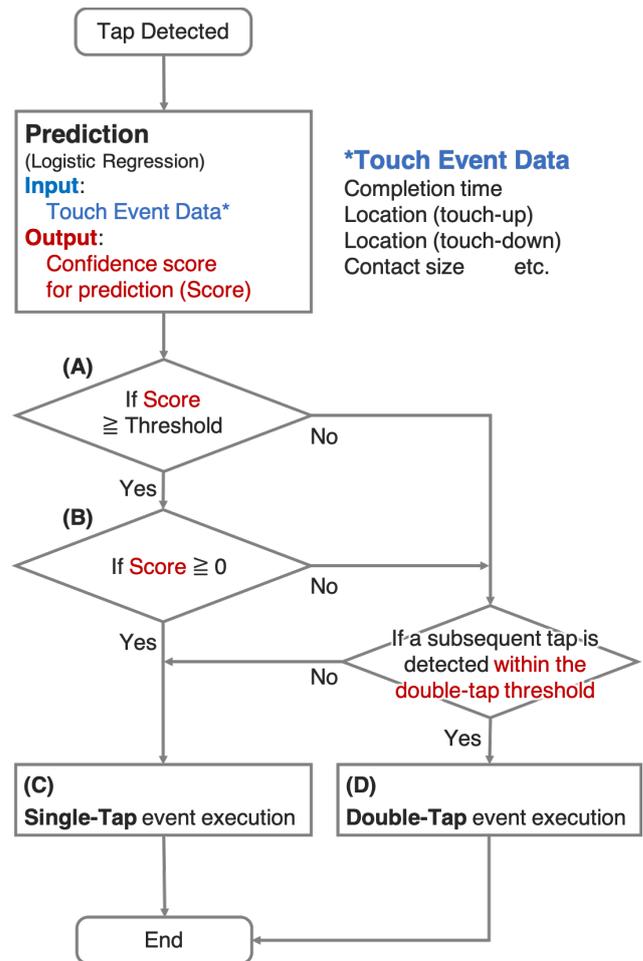


図 3 PredicTaps の処理概要。スコアに対する閾値を適用した処理を明確にするために条件分岐 (A) と (B) を区別して記載したが、実際には閾値は 0 以上を想定している。

ダブルタップかの識別に際し確度が低いデータを識別せず、従来処理と同様に待機時間を設けることを意味する。提案手法はスコアに対する閾値を設けることが技術的に可能であり、かつ全てのタップを判定処理する必要はない。よって高いスコアを得たデータのみを識別対象とし、スコアの低いデータを足切りすることが許容される設計である。これにより、高速化可能なシングルタップの頻度は低下するが、シングルタップとダブルタップの誤認識による操作性の低下を防ぎつつ、全体としてシングルタップ処理の高速化を目指す。

4. 制御実験におけるデータ収集

本節及び次節では、PredicTaps の実現可能性を調査するために実施した二種類の実験について述べる。第一の実験（本節）では、ブックリーダーを使用したページ操作時のデータを、第二の実験（次節）では、特定のタスクを定めず、タッチサーフェスの日常使用時のデータを対象として収集及び機械学習のモデルの開発を行った。いずれの実験においてもラップトップ PC 組み込みのタッチサーフェスを使

用した。

4.1 被験者及び使用機器

10名の被験者（女性8名，男性2名，右利き，年齢平均27.9歳，標準偏差10.6歳）が参加した。本実験では，ラップトップPC（MacBook，macOS 10.13）組み込みのタッチサーフェス（105 × 76 mm）を用いた。ラップトップPCは机上へ置き，ACアダプタに接続した状態にて使用した。本実験ではブックリーダー（eBookJapan^{*2}）を使用し，書籍閲覧中のページ操作時のデータを対象とし収集した。

4.2 実験手順

ブックリーダーはGoogle Chrome上でフルスクリーン表示した。被験者はブックリーダーで利用可能な二冊の漫画と二冊の写真集を選択し，合計四冊の本を閲覧するタスクを行った（ページ数平均173ページ/冊，標準偏差28ページ）。被験者らは選択した本のうち二冊（漫画及び写真集一冊ずつ）をシングルタップにより，残り二冊をダブルタップによりページ送り操作を行い閲覧した。シングルタップ操作とダブルタップ操作の順番は被験者毎にカウンターバランスを取った。タスク中，被験者らは指示されたページ送り操作及び通常のカーソル操作を自身の閲覧ペースに応じて自由に行った。被験者らは操作に利き手の人差し指のみを使用した。また，クリック及びダブルクリックを使用しないよう指示した。実験所要時間は参加者一人あたり60–90分程度であった。

ページ送りにシングルタップまたはダブルタップを割り当てるアプリケーションを実装し，常駐させた状態にて実験を実施した。書籍は画面上に見開き表示（二ページ分表示）されており，カーソルが画面の左半分（左側のページ）にある場合，ページ送り操作により次のページへ移動し，カーソルが画面の右半分（右側のページ）にある場合，ページ送り操作により前のページへ移動する設計とした。また，ダブルタップ閾値には500 msを設定した。被験者が指定された操作と異なるページ送り操作を実行した場合，ページ移動処理は実行しない設計とした。

4.3 収集データ

実験中，一回のページ送り操作毎に下記のデータを記録した。

- (1) タッチ所要時間（タッチダウンからタッチアップまでの経過時間）
- (2) タッチサーフェス上のタッチ位置（タッチダウン時及びタッチアップ時）
- (3) タッチダウン位置とタッチアップ位置の差異
- (4) タッチ座標の移動の速度平均（3を1で除算した値）

(5) タッチサーフェスと指の接触面積最大値

タッチアップ及びダウン位置については，タッチサーフェスの左上の角を原点とし，右方向がX軸正方向を，下（手前）方向がY軸正方向とした。

上述のように，ダブルタップは *pre-determined* な動作であると考えられるため，シングルタップを実行する場合よりも高速な動作であると予想される。よって，タッチ所要時間はシングルタップよりも短くなると考えられる。また，素早い指の動きのため，タッチ時の押圧は僅かに強くなり，タッチサーフェスと指との間の接触面積は大きくなると考えられる。さらに，タッチダウンからタッチアップまでの間に，指の位置（タッチ検出位置）が僅かに移動（ずれが発生）し，(2)(3)(4)の値に影響を与えると予想される。(1)–(5)のデータの取得にはMultitouchSupport.framework^{*3}を用いた。本実験では合計3,959のタッチイベント（シングルタップ: 1,967及びダブルタップ: 1,992）のデータを収集した。

4.4 データ処理及びモデル構築

シングルタップとダブルタップの一回目のタップを識別するために機械学習を利用した。分類問題を解決するにあたり，定性的に重みを確認することが可能であるロジスティック回帰（LIBLINEAR v1.94，L1正則化）を用いた。この際，表2に示す特徴量を用いた。実験にて取得した全被験者のデータのうち，90%を学習（判別モデル生成）に，残りを評価に用いた。学習データと評価データの分配はランダムに行った。不均衡なデータを用いているため，学習はランダムサンプリングを用いて，正例と負例のバランスが同一となるようにした。サンプリングにおけるランダム性を扱うため，10回の評価を行い得られたスコアを評価結果として用いた。また，ロジスティック回帰のコストパラメータをクロスバリデーションにより最適化した。具体的には得られたデータを学習用，パラメータ決定用，評価用の三つに分類し，特定のパラメータ条件下で学習用のデータでモデルの学習を行い，これをパラメータ決定用のデータを用いて評価した。複数のパラメータ条件で上述の操作を実行し，最も高いスコアとなったパラメータを採用した。そして，上述のパラメータ条件下で学習を行い，評価用のモデルで性能を調査し，それをスコアとした。この際，学習用，パラメータ決定用，評価用のデータ分割が恣意的にならないよう各データの割合のみを決定し，どのデータに属すかはランダムに決定した。また，上述の処理を複数回実行しスコアが安定していることを確認した。

4.5 結果と考察

図4に提案手法の識別結果を示す。図の横軸は，スコア

^{*2} <https://ebookjapan.yahoo.co.jp/>

^{*3} http://hci.rwth-aachen.de/guide_trackpad

表 2 L1 正則化により選択された特徴量及びその重み. 選択されなかった特徴量は表下部に記載した.

No.	Feature for common model	Weight
1	タッチ所要時間	12.13
2	タッチ移動速度 (X 軸)	-0.1848
3	タッチダウン位置 (X 軸)	-0.4405
4	タッチ移動速度 (Y 軸)	-0.6927
5	最大接触面積	-0.8047
6	タッチダウン位置 (Y 軸)	-
7	タッチアップ位置 (X 軸)	-
8	タッチアップ位置 (Y 軸)	-
9	タッチダウン-アップ位置の差異 (Y 軸)	-
10	タッチダウン-アップ位置の差異 (X 軸)	-

上位のデータの使用割合を、縦軸は使用割合に対応する識別精度を示す。例示すると、データ使用割合 10% の場合、スコアの上位 10% までの評価用データを対象とした際の識別精度を、データ使用割合 100% の場合、全てのデータを対象とした識別精度を意味する。一般に、スコアに対する識別精度はモデルごとに異なる。そのため、識別精度 (%) は比較・議論可能であるが、そこで用いられているスコア自体はモデル間で比較できない数値である。また、スコアの分布は一様ではないため、スコアに対して定数刻みの閾値を適用した場合、対象となるデータの数が閾値毎に大きく異なることが想定される。上述の理由から、本論文ではデータ使用割合を基準とした識別精度について述べる*4。

各被験者のデータ毎に識別モデルを構築し（以下、個人モデル）、個人利用環境で本手法を使用した際の識別結果を図 4 にドットで示す。また、全被験者のデータから識別モデルを構築し（以下、共通モデル）、複数人利用環境で本手法を使用した際の識別結果を図 4 に実線で示す。全ての評価データを用いた場合、個人モデルの平均識別精度は 83.3%（標準偏差 7.2%）であり、共通モデルの識別精度は 72.2% であった。ここから、使用するデータをスコアが高いものに限定していくと、徐々に識別精度は向上し、使用割合 10% の際にはそれぞれ 99% 及び 96% の識別精度に達した。

また、機械学習において使用した特徴量とその重みを表 2 に示す。正の重みを持つ特徴量はシングルタップの識別に、負の重みを持つ特徴量はダブルタップの識別に寄与したことを表す。最も大きい正の重みを持つ特徴量はタッチ所要時間であった。すなわち、タッチ所要時間が長いほどシングルタップとして識別されやすくなることを意味する。また、最も大きい負の重みを持つ特徴量は最大接触面積であった。すなわち、接触面積が大きいほどダブルタップと

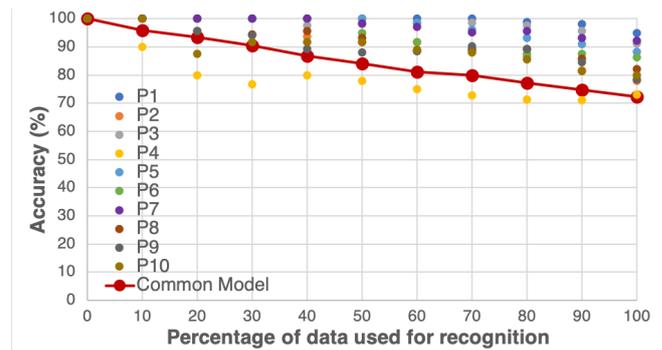


図 4 制御実験における識別精度. 横軸は PredicTaps の処理対象とするスコア上位のデータ割合を示す. 実線は全被験者の共通モデルによる精度を、ドットは各被験者の個人モデルの精度を表す.

して識別されやすくなることを意味する。これらについては、ダブルタップはシングルタップと比較し素早い動作であり、タッチ時の押圧は僅かに強くなるため、タッチサーフェスと指との間の接触面は大きくなるという予想、及びタッチ所要時間はシングルタップ時よりも短くなるという予想と合致するものである。また、タッチダウンからタッチアップまでの間の指の位置のずれ（位置座標の差異）自体は特徴量として選択されなかったものの、タッチの移動の速度が大きいほどダブルタップとして識別されやすくなる結果となった。これも同様に、素早いダブルタップ動作によりタッチ中の指の位置のずれ方は僅かに速かったものと考えられる。

5. 非制御実験におけるデータ収集

上述の実験では、特定のタスク（ブックリーダアプリにおけるページ操作）を対象とした制御実験においてデータを収集し、機械学習によるモデル実装を行った。本実験では、タスクに依存しないモデルを開発し、その識別精度を評価することを目的とした。そのため、非制御環境下におけるタッチサーフェスの日常使用時のデータを収集を行った。

5.1 被験者及び使用機器

6名の被験者（女性5名、男性1名、右利き、年齢平均25.3歳、標準偏差4.2歳）が参加した。本実験では、被験者ら自身が所有するラップトップPC（MacBook, macOS 10.13以降）組み込みのタッチサーフェスを用いた。

5.2 実験手順

タッチサーフェスの日常使用時のデータを収集するため、シングルタップ及びダブルタップを検出し記録する常駐アプリケーションを実装した。前述の制御実験と同様のデータを MultitouchSupport.framework により取得した。これに加え、タッチイベント発生時に、ラップトップPCがACアダプタに接続されていたかバッテリー駆動であったかを記

*4 なお、実際にモデルをアプリに組み込んで使う場合、下記のステップにより閾値を設定する。1. 目標とする識別精度を決定。2. 目標の精度が得られるデータ使用割合を確認。3. 該当モデルの評価に使ったデータのスコア群から、使用割合に対応する足切りのためのスコアを確認し、閾値として設定。

表 3 非制御実験のデータを利用した場合の特徴量及びその重み.

No.	Feature for common model	Weight
1	タッチ所要時間	7.340
2	タッチダウン-アップ位置の差異 (X 軸)	0.6507
3	タッチダウン位置 (Y 軸)	0.4372
4	AC/バッテリー条件	0.3078
5	タッチ移動速度 (X 軸)	0.1497
6	タッチ移動速度 (Y 軸)	0.09435
7	タッチダウン位置 (X 軸)	-0.07981
8	タッチアップ位置 (X 軸)	-0.1203
9	最大接触面積	-0.1748
10	タッチダウン-アップ位置の差異 (Y 軸)	-0.2086
11	タッチアップ位置 (Y 軸)	-0.2969

録した。タッチ入力の感度はタッチサーフェスの GND 条件により影響を受けることが知られており [11], [21], [22], これは、タッチ所要時間や接触面積の計測値に影響を与えうためである。日常使用時には AC アダプタ使用及びバッテリー駆動の両方が想定されるため記録した。被験者は各自が所有するラップトップ PC にこのアプリケーション常駐させ、自由に操作を行った。なお、取得する全てのデータについて被験者らに事前に説明し実験実施の合意を得た。本実験は 4 日間継続し、合計 22,057 のタッチイベント (シングルタップ: 18,861 及びダブルタップ: 3,196) のデータを収集した。

5.3 結果

制御実験と同様のデータ処理及び機械学習を表 3 に示す特徴量を用いて行い、得られた結果を図 5 に示す。全ての評価データを用いた場合、個人モデルの平均識別精度は 89.1% (標準偏差 2.1%) であり、共通モデルの識別精度は 86.2%であった。前実験同様に、使用するデータをスコアが高いものに限定していくと、徐々に識別精度は向上した。個人モデルの平均識別精度はデータ使用割合 40%で、共通モデルはデータ使用割合 20%で 99%の識別精度に達した。

機械学習において使用した特徴量とその重みを表 3 に示す。制御実験の結果と同様に、接触面積が大きいほどダブルタップとして識別されやすく、タッチ所要時間が長いほどシングルタップとして識別されやすくなる結果となった。一方で、タッチ移動の速度については、制御実験と重みの正負が逆転した。また、制御実験においては選択されなかったタッチダウン-アップの位置の差異及びタッチ位置 (ダウン Y 軸, アップ X・Y 軸) が本実験では選択された。上述の特徴量の差異については、制御実験と非制御実験でのタスクの差異により生じたものであると推察される。制御実験のタスクは、ブックリーダにおけるページめくり操作であるため、ユーザは連続してシングルタップまたはダブルタップ操作を行うことが多かった。一方で、非制御実験ではカーソル移動を行い、目的のアイコンやアプ

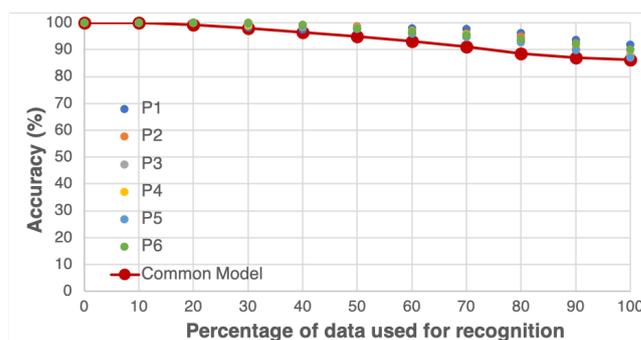


図 5 非制御実験における識別精度.

リケーション内のコンポーネントにカーソルを運んだ後にシングルタップまたはダブルタップを行うという操作が多く、連続したタップ操作は制御実験と比較し低頻度であったと考えられる。よって、連続したタップ操作であるか、またはポインティングとタップ操作を交互に行うかによってタッチ位置や移動速度の特徴量の傾向が異なると推察される。

6. 議論

6.1 レイテンシ削減頻度と識別精度

本論文の実験結果から、高いスコアを得たシングルタップのみをレイテンシ削減の対象とすることで、誤判定による性能劣化を高精度に回避可能であると考えられる。例として、二種類の実験の中で最も高精度なモデルである制御実験の P1 の個人モデル及び、制御実験の共通モデルについて検討する。P1 の個人モデルで 99%以上の精度を得るにはデータ使用割合を 80%に限定する必要がある。この場合、スコア上位の 80%のデータは 99%以上の精度で識別され、そのうちシングルタップと予測されたデータについては、レイテンシが削減される。また、ダブルタップと予測されたデータ及び、スコア下位の 20%のデータに対しては、シングルタップレイテンシを導入した従来と同様の処理が行われる。一方、共通モデルは P1 個人モデルより精度が低く、実用的な精度を得るためには、対象とするデータをより確度の高いデータに限定する必要がある。例えば、96%以上の識別精度を得るためにはデータ使用割合を上位 10%のみとする必要がある。この場合、スコア上位 10%のデータが 96%の精度で識別され、そのうちシングルタップと予測されたデータについては、レイテンシが削減される。また、ダブルタップと予測されたデータ及び、スコア下位の 90%のデータに対しては従来と同様の処理が行われる。以上のように、レイテンシ削減頻度と識別精度はトレードオフの関係にあり、他のモデルと比較し識別精度の低いモデルであっても削減頻度を低くすることで高精度な識別が可能となる。

6.2 個人モデルの適用

本論文で報告した二種類の実験において、一名の被験者を除き個人モデルの適用は識別精度向上に有効であることを確認した（全てのデータを使用した場合、制御実験にて平均 11.1%、非制御実験にて平均 2.9%向上）。これについて、識別の際に重要となった接触面積やタッチ所要時間といった特徴量は、指の形状・皮膚の乾燥状態、タッチする際の動作の癖などの個人差の影響を受けるデータ [22] であったためと考えられる。

6.3 低減可能なシングルタップレイテンシ

共通モデルの機械学習による識別に際し、サーバ (Intel (R) Xeon (R) CPU E5-2630L v3, 1.80GHz) 上での計算所要時間は平均 29×10^{-4} ms だった。本サーバより CPU 性能の低いラップトップ PC による計算所要時間はこれより長くなると予想されるものの、実験で使用した MacBook では 1 ms 未満の所要時間であり、タッチパッドのタップ操作における知覚可能な最短のレイテンシを考慮すると十分に短いといえる [15]。また、ラップトップ PC で識別を行う場合、基本的に機器上で学習を行うことは想定しておらず、学習されたモデルファイルをアプリに組み込み、そのモデル及びタッチセンシング時の計測値とモデルの重みの線形和を計算する。識別プログラムのメモリは 1 KB 程度であることが想定される。

一般に、ダブルタップ閾値は設定を変更可能であるが、本論文で用いた 500 ms の設定の場合について、どの程度シングルタップレイテンシが削減可能であるかについて検討する。本論文での二種類の実験において、シングルタップ動作の平均所要時間は 225 ms（標準偏差 116 ms）であった。この値に基づき単純計算すると、 500 （ダブルタップ閾値） $- 1$ （識別所要時間） $- 225$ （シングルタップ所要時間） $= 274$ ms が一回のシングルタップの低減可能なシングルタップレイテンシの平均値となる。

6.4 他フォームファクタへの適用可能性

本論文では、ラップトップ PC 組み込みのタッチパッドにおいて提案手法の実現可能性を実験的に示した。本実験を通じ開発したモデルは特徴量にタッチ位置を含むため、タッチ位置がコンテンツ依存となるスマートフォンなどに適用する際には特徴量を見直す必要があると考えられる。一方で、小型の携帯型デバイスの場合には加速度や角速度センサなどから取得可能なセンサ値を特徴量に追加することで、識別精度の向上が期待できる。

6.5 マルチタップによる入力手法への適用可能性

ヒューマン・コンピュータ・インタラクション (HCI) 研究領域においては、一般的な OS で広く導入されているダブルタップのみでなく、多様なマルチタップの手法が提案・

検証されてきた。具体的には、位置的に離れた二点に対して行うダブルタップ [9] や 3 点以上のマルチタップ [17]、あるいは、僅かにタップの瞬間をずらしたマルチタップ [5] などが挙げられる。こうした手法は、導入することで効果的に入力語彙の拡張を実現するものの、ダブルタップ導入時と同様、シングルタップとの識別のためにレイテンシが発生する。本論文ではシングルタップとダブルタップの識別のみを対象としたが、上述の手法も基本的にはダブルタップのように *pre-determined* な素早い動作であることを考慮すると、提案手法の導入を検討する余地があると考えられる。

6.6 ダブルタップイベント処理の高速化

本論文ではシングルタップレイテンシの削減を主な課題とし議論したが、ダブルタップイベント処理の高速化についても PredicTaps により可能であると考えられる。これは、一回目のタップが発生した時点で、該当の入力はダブルタップであるという予測の場合に、バックグラウンドでダブルタップ実行時の処理（コンテンツの読み込みなど）を事前実行しておき、二回目のタップが発生した直後に処理結果を表示するといった実装を検討できるためである。

7. 今後の課題

提案手法は実行頻度と識別精度がトレードオフとなる手法である。よって、誤判定を避けるために高いスコアのデータのみを使用する場合、数回に一回の頻度でシングルタップイベントの処理が高速化される。この場合、ユーザーにとって PredicTaps は不規則にレイテンシが短縮されるシステムであると認識され、「動作が不安定なシステムである」などのネガティブな印象を与えることが懸念される。こうした印象を与えるか否か、また操作性に影響を与えるか否かについては、今後 PredicTasp を組み込んだアプリを使用したユーザ実験により評価を行う。

8. 結論

本論文では、シングルタップとダブルタップを同一のコンポーネントに対し実行可能な設計とする場合、シングルタップ判定の際に生じるレイテンシを削減する手法として PredicTaps を提案した。PredicTaps は、一回目のタッチ入力終了時点で、検出されたタップがシングルタップとダブルタップの一回目のどちらの入力であるかを予測する。そして、予測に対するスコアに基づきシングルタップイベントを迅速に実行するか、または後続する可能性のあるタップを待機するかを決定する。これにより、シングルタップとダブルタップの誤認識による性能低下を防ぎつつシングルタップ処理の高速化を目指す。本論文では、制御されたタスク・環境及び、非制御環境（日常使用時）におけるタップのデータ収集を行い、機械学習によるモデル開発

を行った。機械学習の結果は、高いスコアを得たタップのみをシングルタップレイテンシ削減の対象とすることで、削減可能なレイテンシは減少するが、誤判定による性能劣化は高精度に回避可能であることを示唆するものである。今後は PredicTasp を組み込んだアプリを使用したユーザ実験により評価を行い、実環境での提案手法の有効性を検証する。

参考文献

- [1] Kusano, T. and Komuro, T.: 3D Tabletop User Interface with High Synchronization Accuracy Using a High-speed Stereo Camera, *Proceedings of the 2015 International Conference on Interactive Tabletops & Surfaces*, ITS '15, New York, NY, USA, ACM.
- [2] Smith, M. W., Sharit, J. and Czaja, S. J.: Aging, motor control, and the performance of computer mouse tasks, Vol. 41, No. 3, CA, Los Angeles, USA, SAGE Publications Sage.
- [3] Schwarz, J., Mankoff, J. and Hudson, S. E.: An Architecture for Generating Interactive Feedback in Probabilistic User Interfaces, *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, New York, NY, USA, ACM.
- [4] Bi, X. and Zhai, S.: Bayesian Touch: A Statistical Criterion of Target Selection with Finger Touch, *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*, New York, NY, USA, ACM.
- [5] Oakley, I., Lee, D., Islam, M. R. and Esteves, A.: Beats: Tapping Gestures for Smart Watches, *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, New York, NY, USA, ACM.
- [6] Man, Q. and Liu, X.: CN Patent 103902086A, Curve fitting based touch trajectory smoothing method and system, <https://www.google.ca/patents/CN103902086A?c1=en>.
- [7] Ng, A., Lepinski, J., Wigdor, D., Sanders, S. and Dietz, P.: Designing for Low-latency Direct-touch Input, *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology*, UIST '12, New York, NY, USA, ACM.
- [8] Kobayashi, M., Hiya, A., Miura, T., Asakawa, C., Hirose, M. and Ifukube, T.: Elderly User Evaluation of Mobile Touchscreen Interactions, *Human-Computer Interaction - INTERACT 2011*, Berlin, Heidelberg, Springer.
- [9] Heo, S., Gu, J. and Lee, G.: Expanding Touch Input Vocabulary by Using Consecutive Distant Taps, *Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems*, CHI '14, New York, NY, USA, ACM.
- [10] Surale, H. B., Matulic, F. and Vogel, D.: Experimental Analysis of Mode Switching Techniques in Touch-based User Interfaces, *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, New York, NY, USA, ACM.
- [11] Grosse-Puppenthal, T., Holz, C., Cohn, G., Wimmer, R., Bechtold, O., Hodges, S., Reynolds, M. S. and Smith, J. R.: Finding Common Ground: A Survey of Capacitive Sensing in Human-Computer Interaction, *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, New York, NY, USA, ACM.
- [12] Schwarz, J., Hudson, S., Mankoff, J. and Wilson, A. D.: A Framework for Robust and Flexible Handling of Inputs with Uncertainty, *Proceedings of the 23rd Annual ACM Symposium on User Interface Software and Technology*, New York, NY, USA, ACM.
- [13] Deber, J., Araujo, B., Jota, R., Forlines, C., Leigh, D., Sanders, S. and Wigdor, D.: Hammer Time!: A Low-Cost, High Precision, High Accuracy Tool to Measure the Latency of Touchscreen Devices, *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, New York, NY, USA, ACM.
- [14] Jota, R., Ng, A., Dietz, P. and Wigdor, D.: How Fast is Fast Enough?: A Study of the Effects of Latency in Direct-touch Pointing Tasks, *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, New York, NY, USA, ACM.
- [15] Deber, J., Jota, R., Forlines, C. and Wigdor, D.: How Much Faster is Fast Enough?: User Perception of Latency & Latency Improvements in Direct and Indirect Touch, *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, New York, NY, USA, ACM.
- [16] Henze, N., Mayer, S., Le, H. V. and Schwind, V.: Improving Software-reduced Touchscreen Latency, *Proceedings of the 19th International Conference on Human-Computer Interaction with Mobile Devices and Services*, MobileHCI '17, New York, NY, USA, ACM.
- [17] Heo, S., Jung, J. and Lee, G.: MelodicTap: Fingering Hotkey for Touch Tablets, *Proceedings of the 28th Australian Conference on Computer-Human Interaction*, OzCHI '16, New York, NY, USA, ACM.
- [18] Schwarz, J., Mankoff, J. and Hudson, S.: Monte Carlo Methods for Managing Interactive State, Action and Feedback under Uncertainty, *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, New York, NY, USA, ACM.
- [19] Hsieh, W.: More Responsive Tapping on iOS, <https://webkit.org/blog/5610/more-responsive-tapping-on-ios/>.
- [20] Nancel, M., Vogel, D., De Araujo, B., Jota, R. and Casiez, G.: Next-Point Prediction Metrics for Perceived Spatial Errors, *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, UIST '16, New York, NY, USA, ACM.
- [21] Ikematsu, K., Fukumoto, M. and Siio, I.: Ohmic-Sticker: Force-to-Motion Type Input Device That Extends Capacitive Touch Surface, *Proceedings of the 32Nd Annual ACM Symposium on User Interface Software and Technology*, UIST '19, New York, NY, USA, ACM.
- [22] Ikematsu, K. and Siio, I.: Ohmic-Touch: Extending Touch Interaction by Indirect Touch Through Resistive Objects, *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, New York, NY, USA, ACM.
- [23] Le, H. V., Schwind, V., Göttlich, P. and Henze, N.: PredicTouch: A System to Reduce Touchscreen Latency Using Neural Networks and Inertial Measurement Units, *Proceedings of the 2017 ACM International Conference on Interactive Surfaces and Spaces*, ISS '17, New York, NY, USA, ACM.
- [24] Buschek, D. and Alt, F.: ProbUI: Generalising Touch Target Representations to Enable Declarative Gesture Definition for Probabilistic GUIs, *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, New York, NY, USA, ACM.

- [25] Keele, S. W. and Posner, M. I.: Processing of visual feedback in rapid movements., Vol. 77, No. 1, Washington, USA, American Psychological Association.
- [26] Cattan, E., Rochet-Capellan, A., Perrier, P. and Bérard, F.: Reducing Latency with a Continuous Prediction: Effects on Users' Performance in Direct-Touch Target Acquisitions, *Proceedings of the 2015 International Conference on Interactive Tabletops & Surfaces*, ITS '15, New York, NY, USA, ACM.
- [27] Henze, N., Funk, M. and Shirazi, A. S.: Software-reduced Touchscreen Latency, *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services*, MobileHCI '16, New York, NY, USA, ACM.
- [28] Schmidt, R. A., Zelaznik, H. N. and Frank, J. S.: Sources of inaccuracy in rapid movement, *Information processing in motor control and learning*, Amsterdam, The Netherlands, Elsevier.
- [29] Vatavu, R.-D., Cramariuc, G. and Schipor, D. M.: Touch Interaction for Children Aged 3 to 6 Years, Vol. 74, No. C, Duluth, MN, USA, Academic Press, Inc.
- [30] Ritter, W., Kempster, G. and Werner, T.: User-Acceptance of Latency in Touch Interactions, *Universal Access in Human-Computer Interaction. Access to Interaction* (Antona, M. and Stephanidis, C., eds.), Berlin, Heidelberg, Springer.
- [31] Microsoft: Windows Dev Center, <https://docs.microsoft.com/ja-jp/windows/desktop/Controls/ttm-setdelaytime>.
- [32] Wang, W., Liu, X. and Guangdao, Z.: WO Patent 2013170521A1, Multi-touch tracking method, <https://www.google.com/patents/W02013170521A1?cl=en>.
- [33] Byoungeul, K. and Yeongkyu, L.: WO Patent 2014129753A1, Mobile terminal and touch coordinate predicting method thereof, <https://www.google.com/patents/W02014129753A1?cl=en>.