

# スマートフォンを動かすドラッグ手法の提案と実装

前寺 百合子 (指導教員：椎尾 一郎)

## 1 はじめに

近年、スマートフォンやスマートウォッチなどのタッチサーフェスを実装した端末 (以下、タッチ端末もしくは端末) は広く普及した。通常、タッチ端末におけるタッチ入力では、ユーザは端末に向けて指を動かし、画面に振れることでタッチ入力を行う。一方で、端末を指の方向に動かしタッチ入力を行い、この動作を通常のタッチ入力と識別することでタッチ入力語彙を拡張する手法として Copernican-Touch[1] が提案されている。

そこで本論文では、Copernican-Touch の拡張手法として Copernican-Drag を提案する。通常のドラッグではユーザは端末を固定し指を動かすが、Copernican-Drag では指を固定し端末を動かす。両ドラッグを識別するために、センサデータからの特徴量を検討し、iOS の Core ML フレームワーク上に分類器を実装した。また、両ドラッグにより操作モードを切り替える iOS アプリケーションを作成した。

## 2 先行研究

Copernican-Touch [2] では、指または端末を動かすタッチの識別に、デバイス内蔵の加速度センサ及びジャイロセンサ (以下、センサ) を利用する。機械学習によりタッチ直前 200 ms のセンサデータを分類した結果、スマートフォンでは 92% の精度で両タッチを識別可能であることが示されている。

## 3 ドラッグの識別

指を動かすドラッグと端末を動かすドラッグ動作を識別するにあたり、ドラッグ発生直後 100 ms から 600 ms のセンサデータを取得し、これを学習データセットとして特徴量を抽出し、機械学習により分類器を作成する。この分類器を端末アプリケーションに組み込むことで、両ドラッグによる操作モード切り替えを実装できる。

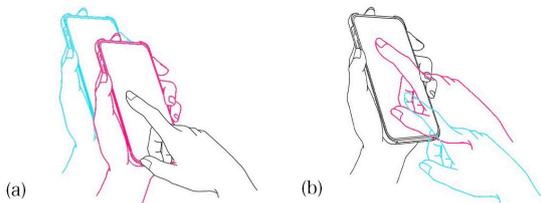


図 1: (a) タッチサーフェスを動かし指に当てるタッチ入力動作を用いた Copernican-Drag, (b) 通常の指を動かしてタッチサーフェスへ触れるドラッグ入力。青の描画線は動作の開始時点を、赤の描画線は動作の終了時点を示す。

## 3.1 取得データ

先行研究 [2] では、タッチ直前の 200 ms のセンサデータを学習データとして採用した。これは、端末を動かすタッチではタッチ直前の端末のセンサデータが重要であることから採用された。本研究では、通常のドラッグと端末を動かすドラッグを識別するにあたり、指が画面に触れた直後のデータの影響が大きいと判断した。また、ドラッグを開始してから安定して端末を動かすことができるようになるまでの端末の動きを排除した。そして、排除した時間直後の 100, 200, 300, 400, 500 ms のデータを用いて識別を試みた。その結果、500 ms において最も高い識別精度が得られた。500 ms はシングルタップとダブルタップ判定時間として一般である<sup>1</sup> ことから、ドラッグ発生直後 100 ms から 600 ms のセンサデータを学習データとして採用した。

また、センサデータのサンプリングレートに関しては先行研究と同様に、本論文の実験で使用する端末 (iPhone X) 内蔵の三軸加速度とジャイロセンサが更新可能な最大周波数の 100 Hz にて取得することにした。<sup>2</sup>

## 3.2 特徴量

センサデータの取得には OS の Core Motion フレームワーク<sup>3</sup> を使用した。これにより、iPhone X の内蔵センサから、ドラッグ直後 100 ms からドラッグ直後 600 ms の端末の加速度 (x, y, z) および角速度・姿勢 (pitch, roll, yaw) と指が画面に触れている座標 (x, y) を 100 Hz のサンプリングレートで取得した。得られた加速度、角速度、姿勢、座標のそれぞれの値、加速度、角速度のスカラー値に対して、それぞれ 500 ms 間の平均値・最大値・分散を算出した。さらに、これらのデータにつき 10ms 前 (1 個前のサンプル値) との差分を取り、それらの平均値・最大値・分散値を算出した。これらの特徴量を使用し機械学習を行った。

## 3.3 機械学習

学習に用いるセンサデータは、画面のランダムな場所にターゲットを表示し、これに指または端末を動かしてドラッグする指示をランダムに与える予備実験により収集した。被験者は 7 名 (男性 1 名、女性 6 名、平均年齢 31 歳、標準偏差 14.73) である。それぞれが片手 (左手 6 名、右手 1 名) にスマートフォンを把持し、把持しない側の手で画面ドラッグ操作を行なった。

<sup>1</sup><https://docs.microsoft.com/ja-jp/windows/win32/controls/ttm-setdelaytime>

<sup>2</sup>[https://developer.apple.com/documentation/coremotion/getting\\_raw\\_accelerometer\\_events](https://developer.apple.com/documentation/coremotion/getting_raw_accelerometer_events)

<sup>3</sup><https://developer.apple.com/documentation/coremotion/getting-processed-device-motion-data/understanding-reference-frames-and-device-attitude>

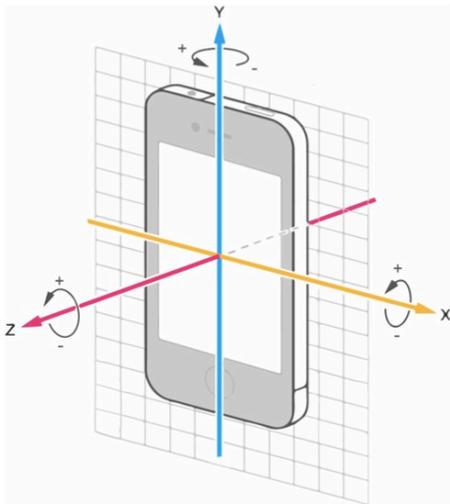


図 2: スマートフォンにおける加速度・ジャイロセンサの座標系の定義 (引用元 [1])

まず、被験者一人のデータ (ドラッグ 270 回)のうちドラッグ 180 回を学習用に、ドラッグ 90 回をテスト用に分けて学習を行った。その結果、トレーニングスコアは平均 99%, テストスコアは平均 87.17% だった。

### 3.3.1 クロスバリデーション

次に、全被験者データを学習用・パラメータ評価用・テスト用に分けて学習を行った。本研究ではデータセットの分割方法に leave-one-user-out-cross-validation を採用した。これは 1 被験者のデータ群を学習・評価・テストデータセットのいずれか 1 つに振り分けて学習を行い、分類器生成時に利用したデータセットに含まれない被験者データを用いてテストを行う。このテストスコアが汎用的な精度となる。今回、生成された分類器のテストスコアは平均 85.3% であり、被験者一人のデータで行ったものより精度が低くなっていた。考えられる原因として、被験者のよって利き手や端末を動かすドラッグの動きが違っていることが挙げられる。また、実験時において、通常のドラッグ時に端末を持つ手を補助的に動かしてしまう被験者もいたことが影響していると考えられる。被験者全員の共通モデルにおいて、通常ドラッグの際に意識的に端末の固定を促したり、データ量を増やしたりすることで更なる精度向上が見込まれる。

## 4 アプリケーション実装

前節で生成した機械学習によるモデルを iOS の Core ML フレームワークに移植し、指または端末を動かすドラッグにより、端末操作モードを切り替えるアプリケーションを iPhone X (iOS 13.51) 上に実装した。端末を動かすドラッグは、ショートカット機能の代替、ゲームの操作拡張、操作系の選択に利用できると思われる。ゲームの一例として、通常ドラッグでは弓の位置を、端末を動かすドラッグでは弓を弾く動作を付与した的当てゲームの作成の

可能性を考えた。また、描画ソフトでのペンツールとスクロールの切り替えの際に行われるボタン操作を、端末を動かすドラッグにより省略することができる。

### 4.1 図形アプリ

描画アプリの簡易版に本機能を組み込んだ。通常のドラッグでは絵を種類ごとに移動させることができ、端末を動かすドラッグでは絵をまとめて移動させることができる。これにより、ユーザが操作を切り替えるための命令を減らすことができる。

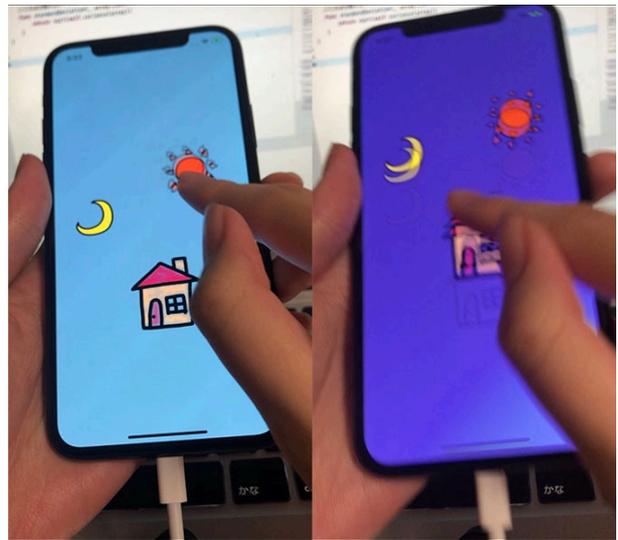


図 3: 左: 通常ドラッグにより絵をバラバラに動かすことができる。右: Copernican-Drag により絵をまとめて動かすことができる。

## 5 まとめ

端末を動かすドラッグの提案を行い、スマートフォン上でアプリケーションを実装した。今後は、アプリケーションの動作に安定性がないため安定して識別できるように改良し、実際にアプリケーションの評価をしていきたい。また、指を動かすタッチ、端末を動かすタッチ、指を動かすドラッグ、端末を動かすドラッグの四種類を識別を行い、本研究の操作手法を使ったゲームの実装を行いたい。

## 参考文献

- [1] 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.
- [2] 下田寛子, 椎尾一郎. スマートフォンを動かし指に接触させるタッチ手法の改良と実装.