

## 非視認状態を想定した新たなタップ操作インタフェースの多様化の提案

加藤馨 (学生会員)      牧隼永 (非会員)      舟橋健司 (非会員)  
名古屋工業大学

### Diversification of a novel tap operation interface for non-visual state

Kyo Kato      Hayato Maki      Kenji Funahashi

Nagoya Institute of Technology, Japan

k.kato.789@stn.nitech.ac.jp

kenji@nitech.ac.jp

#### 概要

現在、タブレットやスマートフォンなどのタッチパネル式デバイスが普及している。我々はこれまでに、これらのデバイスが有する静電容量式のタッチパネルを対象に、新しく「逆タップ」という操作方法を提案している。逆タップは通常のタップとは異なり、画面から指が離れて再び触れた場合にタップ操作と判定する。この新しいタップ方法と画面に突起物の付いたタッチパネル式デバイスにより、視覚障害者でもボタンの配置や機能を覚えておくことで操作が可能である。本論文では、逆タップを元にして2本指逆タップという新しい操作方法を提案する。2本指逆タップは逆タップを2本の指で行うものであり、2本指逆タップを提案することで、1つの操作箇所に対する操作の選択肢の多様化を試みる。

#### 1 はじめに

現在、タブレットやスマートフォンなどのタッチパネル式デバイスが普及している。総務省による令和元年版情報通信白書によると、2018年における世帯のスマートフォンの保有率は79.2%であり、パソコンの保有率の74.0%を上回っている [1]。スマートフォンやゲーム機などの多くのタッチパネル式デバイスでは、静電容量式のタッチ操作検出方式が多く利用されている。これらのデバイスにはパソコンや従来型の携帯電話に搭載されているキーボードや通話ボタンなどの物理的なボタンが搭載されていない場合が多い。したがって、点字のような触覚の手掛かりがないため画面を視認していない状態で思い通りに操作することはできず、また画面を一瞥した

後に画面を見ずに操作することも難しい。そこで我々は静電容量式のタッチパネルを対象に、「逆タップ」という新しいタップ方法を提案している [2]。指が画面から離れている状態だけでなく、触れている状態も基本的に無操作状態であると考え、操作箇所では指を離れたのちにすぐ同じ場所に触れた場合にタップ操作が行われたと判定する。なお、タッチパネル画面上に物理的に点字のような突起物をつけておき、また通常のタップ、スワイプ操作の判定はしない。これにより、画面を見ずに突起物の触覚的特徴で操作箇所を見つけ、タップ相当の操作を行うことができる。

ところで、パソコンではマウスの左クリックと右クリック、スマートフォンではタップとプレス&ホールドなど、一つの操作箇所に対して複数の操作が可能の場合

が多く、それはユーザにとって便利である。そこで、本論文では逆タップに加えて 2 本指逆タップという新しい操作方法を提案する。2 本指逆タップは逆タップを 2 本指で同時に行うことである。2 本指逆タップを提案することで、1 つの操作箇所に対する操作の選択肢の多様化を試みる。

## 2 逆タップ

本節では、文献 [2] の逆タップの概要を説明する。ユーザが以下の動作を行った時に逆タップと判定する。

1. 指で画面に触れる
2. 画面に触れたまま指を動かし、操作箇所を見つける
3. タップしたい箇所で指を画面から離す
4. 一定時間内に画面の同じ箇所に触れる

また、合わせてダブル逆タップ操作についても考慮している。逆タップを短時間に連続で行うことでダブル逆タップの操作と判定する。図 1 に逆タップ操作のアルゴリズムフローを示す。実装には AndroidStudio の `onTouchEvent` を用いている。イベントは以下のとおりである。

- ACTION\_DOWN: 1 箇所もタッチしていない状態でタッチした時
- ACTION\_UP: 1 本の指でタッチしている状態でタッチしなくなった時
- ACTION\_POINTER\_DOWN: すでに 1 箇所以上でタッチしている状態で追加タッチした時
- ACTION\_POINTER\_UP: 複数の指でタッチしている状態で、そのうち 1 つの指を離れた時
- ACTION\_MOVE: タッチ中に指を動かした時

ACTION\_UP, ACTION\_POINTER\_UP は画面から指が離れた時のアクションであり、大域変数  $u$  に指が離れた座標と時刻を保存する。ACTION\_DOWN, ACTION\_POINTER\_DOWN は指が画面に触れた時のアクションであり、大域変数  $d$  に指が触れた座標と時刻を保存する。ACTION\_MOVE は指が画面上を動いた時のアクションであるため、なにも行わない。画面に指が触れた時の座標と時刻が、前回離れた時の座標と時刻と近ければ、逆タップ操作と判定する。

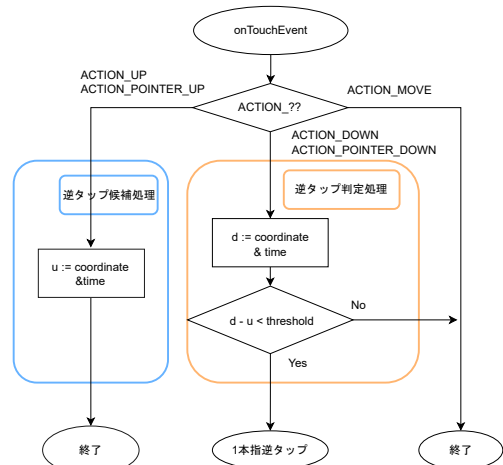


図 1: 逆タップのフローチャート ([2] より引用)

## 3 2 本指逆タップ

本節では新たに提案する 2 本指逆タップについて説明する。ユーザが以下の動作を行った時に 2 本指逆タップ操作が行われたと判定する。なお、この判定においても、通常の逆タップ、およびダブル逆タップ操作への対応も考慮しておく必要がある。

1. 2 本以上の指で画面に触れる
2. 画面に触れたまま指を動かし、操作箇所を見つける
3. タップしたい箇所で、少なくとも 2 本の指を、互いに近い位置で画面から離す
4. 一定時間以内に画面の同じ箇所に、2 本指で触れる

しかし実際には、2 本の指を厳密に同時に離したり触れたりすることは困難であり、またタブレット端末自体も同時に判定することはできない。そのため、以下のような動作と考える。

1. 2 本以上の指で画面に触れる
2. 画面に触れたまま指を動かし、操作箇所を見つける
3. タップしたい箇所で 1 本目の指を画面から離す
4. 一定の距離以内のもう 1 本の指を画面から離す
5. 一定時間以内にある指でほぼ同じ箇所に触れる
6. 一定時間以内に他の指でほぼ同じ箇所に触れる

上記 3 と 4 および 5 と 6 はほぼ同時に行われる。また、離すとき、および触れるときの 2 本の指（位置）の順序は問わない。この一連の動作は、「互いに近い 2 本の指

が同時に逆タップを行う」と言い換えることができる。

図2に逆タップ、すなわち1本指逆タップ操作も考慮されている2本指逆タップのアルゴリズムフローを示す。実装には1本指逆タップ操作の時と同様に AndroidStudio の onTouchEvent を用いる。また、スレッド処理を行う場合に利用されることの多い Java の Handler クラスの postDelayed と removeCallbacks も用いる。postDelayed は第一引数で指定した Runnable オブジェクトの処理を、第二引数で指定した時間 [ミリ秒] の後に実行する。removeCallbacks は postDelayed で実行しようとした処理を取り消すことができる。フローチャート内で座標と時刻を格納しておく  $u, u2, d, d2, d3$  は大域変数とし、初期値はともに  $-\infty$  ( $x, y$  座標, および時刻をプログラムにおける最小値) としておく。なお、これらの大域変数に「[t]」とついている場合はその変数の時刻要素を表す。定数である **threshold, threshold2** は閾値であり、事前に値を決めておく。まず、タッチパネル上で指が触れる (ACTION\_DOWN または ACTION\_POINTER\_DOWN), 指が離れる (ACTION\_UP または ACTION\_POINTER\_UP), または指が触れながら動く (ACTION\_MOVE) アクションのいずれかが起こると onTouchEvent が開始され、アクションの種類の判定をする。2本指逆タップの判定は、逆タップ候補としての処理パートと、逆タップ判定パートに分けて行う。タッチパネルから指が離れた場合に逆タップ候補処理へ移行する。この時、短時間に近い位置で離す動作を行っている、かつ、連続で離す動作を行っているという条件が満たされる場合は、短時間に連続で近い位置で離す動作が行われており、2本指逆タップ候補とする。次に、タッチパネルに指が触れた場合に逆タップ判定処理へ以降する。この時、短時間で近い位置に触れる動作を2回行っている、かつ、前回の指を触れる動作と今回の指を触れる動作の間に指を離す動作を行っていない、かつ2回連続で指を触れる動作をしているが3回連続ではないという条件が満たされる場合には、2本指逆タップ判定処理へ移行する。2本指逆タップ判定処理で、2回ずつの離す動作と触れる動作を同じ場所で短時間で行っているという条件が満たされる場合に、2本指逆タップと判定する。

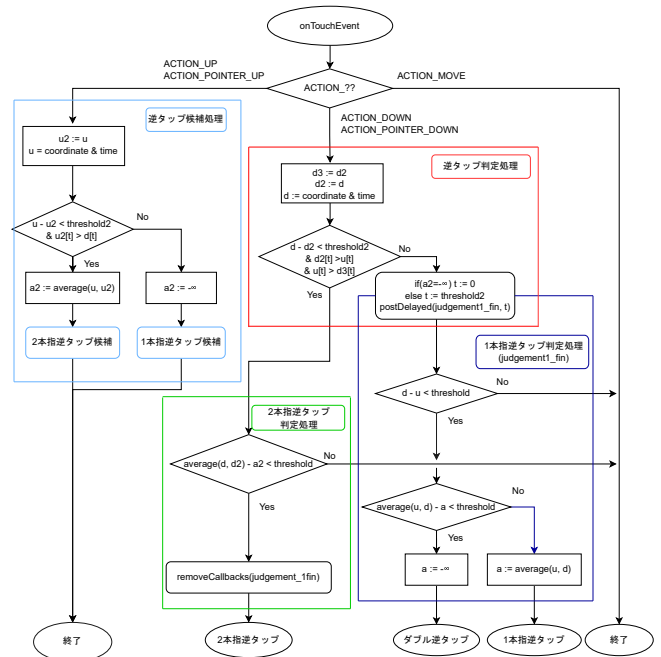


図2: 2本指逆タップのフローチャート

#### 4 視覚障害者を想定した実験

提案した2本指逆タップが、目が見えない状態で実際に操作可能であるのかどうかを確認するために、全盲状態での空調操作パネルの操作を想定した通常のタップと2本指逆タップの比較実験を行った。まず、電話やテンキーを参考に、図3(a)のように画面を12のボタン領域に分けた。次に、画面を見ずに触覚的手掛かりのみで操作できるように各ボタンに点字状の突起物を張り付けた。これにより被験者は目が見えない状態でも手探りで自分が画面上のどこを触っているか判別できる。また、何らかの操作が行われた時にはバイブレーションにより合図する。これによりユーザは画面を見ずとも操作できたことを確認できる。実験を始める前に被験者に1本指の逆タップと2本指逆タップの操作方法を説明し、タブレットの操作練習時間を設けた。これは逆タップ操作に慣れてもらい、またボタンの配置と突起物の配置を記憶してもらうことを目的としている。被験者は大学生および大学院生7名である。この実験は全盲者を想定しているため、被験者にアイマスクを装着してもらい、全盲状態で実験を行った。実験では通常タップと2本指逆タップそれぞれにおいて、1分間、タブレット操作タスク(例: 温度を上げる)を聞き、タスクを実行したら合図をしても



(a) 操作画面



(b) 実験の様子

図 3: 使用したタブレット及び実験の様子

らった. 図 3(b) に実験の様子を示す.

まず, 全盲状態での指定通りのボタン押し操作の成功回数と失敗回数を表 1, 表 2 に示す. なお, 表 2 における成功回数には, 指定通りのボタンに対する 2 本指逆タップだけでなく, 1 本指の逆タップ実行数も含まれる. 通常のタップ操作では, 誤ったボタンの箇所で操作が多かった. 一方で, 逆タップ操作では誤ったボタンの箇所で操作はなかった. 全盲状態において, 通常のタップよりも逆タップ操作の方が正しく操作できることが確認できる. 次に, 同実験において, 各被験者の 2 本指逆タップ操作の成功回数のうち, 実際には 1 本指での逆タップ操作であった回数を調べた (表 3). 実験では全ての操作を 2 本指逆タップで行うように指示をしてあり, また被験者も 2 本指逆タップの操作を行ったと申告している. しかし実際には, 1.5 割程度の操作は 1 本指の逆タップとして判定されていた. 被験者が 2 本指逆タップ操作に慣れていなかったかもしれないことに加えて, タッチパネルに貼り付けた突起物の真上は指の接触判定精度が低くなってしまっていたこと, 被験者がタッチパネルの分解能よりも 2 本の指の間隔を狭くしてしまっていたこと, などの原因が考えられる. 操作の 8 割以上は正しく 2 本指逆タップを行っていたので, 単に慣れにより, 改善が期待できそうである.

## 5 むすび

本研究では, これまでに提案している, 非視認状態を想定したタップ操作方法である逆タップに加えて, 2 本指

表 1: 通常タップ実験

被験者	失敗	成功
A	0	8
B	1	10
C	3	9
D	4	10
E	4	10
F	5	6
G	5	7
平均	3.12	8.57

表 2: 2 本指逆タップ実験

被験者	失敗	成功
A	0	9
B	0	11
C	0	10
D	0	13
E	0	10
F	0	10
G	0	11
平均	0.0	10.57

表 3: 2 本指逆タップ操作時の実際の内訳

被験者	操作回数	2 本指逆タップ	1 本指逆タップ
A	9	6	3
B	11	9	2
C	10	8	2
D	13	12	1
E	10	10	0
F	10	9	1
G	11	8	3
平均	10.57	8.86	1.71

逆タップという新たな操作方法を提案した. 1 つの操作箇所に対して様々な操作方法を用意することで, より複雑なアプリケーションの実現が可能となる. 今後は, 操作方法の種類が過剰にならないように熟慮しつつ, 見えない状態での操作をより容易にするような新たな, 例えばスワイプ操作に相当するような操作方法についても検討していきたい.

## 謝辞

本研究の一部は JSPS 科研費 JP20K11918 の助成を受けたものです.

## 参考文献

- [1] 総務省 令和元年版情報通信白書 情報通信機器の保有状況, <https://www.soumu.go.jp/johotsusintokei/whitepaper/ja/r01/html/nd232110.html>. 参照: 2021-7-20.
- [2] Kenji Funahashi, Hayato Maki, Yuji Iwahori . Novel tap operation on capacitive touch screen for people with visual impairment. ICAT-EGVE 2020, pp.3-4 (Posters and Demos), 2020.