

# 非線形関数を用いた仮想タッチパッドによる 大画面スマートフォンの片手操作手法 A One-Handed Operation Method for Large-Screen Smartphones Using a Virtual Touchpad with Nonlinear Functions

中村 竜蔵

Ryuzo Nakamura

法政大学情報科学部デジタルメディア学科

E-mail: ryuzo.nakamura.2u@stu.hosei.ac.jp

## Abstract

*In recent years, smartphones with 6-inch or larger screens have become popular. However, such large-screen smartphones are difficult to operate with one hand. Some functions for supporting one-handed operation reduce the size of the screen to be displayed, which loses the benefits of large-screen smartphones. By contrast, the virtual touchpad supports one-handed operation without losing the benefits. This paper proposes a virtual touchpad that can manipulate a distant area and accurately manipulate a close area by using a nonlinear function. The user can move the cursor by sliding the thumb within the virtual touchpad displayed at the bottom of the screen. Within the close area, the cursor performs the same action as the thumb, but outside the close area, it begins to move largely in relation to the move of the thumb. A preliminary experiment was conducted to compare eight virtual touchpads with different functions and conditions, such as where the cursor begins to move largely and whether the cursor coordinates are reset. Also, further experiments were conducted to compare the three best virtual touchpads with a conventional direct touch operation, as well as to evaluate the proposed method in terms of website operations. The results showed that the proposed method could reduce the selection time and significantly reduce the workload of the lower thumb.*

## 1. はじめに

近年、大画面のスマートフォンが普及している。iPhone シリーズや Xperia シリーズでも、画面サイズが 6 インチを超えるモデルが主流である。しかし、大画面スマートフォンは、1 画面に表示できる情報量が多いといったメリットを持つ代わりに、片手で操作しにくいといった明確なデメリットをもつ。つまり、片手で荷物を持っているときや電車のつり革に掴まっているときのような片手操作を強いられる状況では、スマートフォンを快適に操作することが困難である。片手で操作しにくいというデメリットに対する解決策として、iPhone では簡易アクセス、Android スマートフォンでは片手モードが存在

するが、どちらも表示させる画面のサイズを縮小するのであり、大画面スマートフォンのメリットを失わせている。

本論文では、大画面スマートフォンのメリットを失わせることなく、片手操作を補助する新たな操作手法を提案する。画面下部の利き手側に半透明なタッチパッドを表示させる。ユーザーは、タッチパッド上で指をスライドさせることによって、画面上に表示されているカーソルを動かすことができ、指を離すことでカーソルの位置が選択される。これにより、ユーザーは親指のみで全画面の操作が可能となる。従来の手法と異なり、指の移動距離に対するカーソルの移動距離は、線形関数ではなく非線形関数にしたがっている。

## 2. 関連研究

Lai ら [1]は、大画面スマートフォンの片手操作手法として挙げられる 3 つの仮想ポインティングモデルを比較した。具体的には、親指操作によるカーソル制御のための既存の仮想ポインティング技術である Virtual Touchpad, Virtual Joystick, Virtual Direction Key の比較実験を行った。結果として、Virtual Touchpad は平均選択時間が 3 つの中で最も短く、ユーザー満足度は 4 項目中 3 項目で最も高い評価を得た。しかし、平均エラー率に関しては、Virtual Direction Key より 0.6% 近く高かった。それにより、ユーザー満足度の正確さの項目では、Virtual Direction Key より低い評価となった。原因として、カーソルを画面全体に動かせるようにするために、指をスライドさせた距離より大きくカーソルが動くように設計されているからだと考えられる。

実際の距離よりも遠くの距離のものを操作できる技術として、Poupyrev ら [2]が提案した Go-Go Interaction Technique が挙げられる。この技術は、VR 空間における仮想の腕の長さを自由に変えられるというメタファーに基づいており、腕を通常届かない距離まで伸ばすことができる。ユーザーがある一定の距離より短い範囲で操作している場合は、仮想の手は実際の手と同様の動きをする。しかし、ある一定の距離を超えると、非線形関数にしたがって仮想の腕は大きく伸びる。これにより、近くにある物体も遠くにある物体もシームレスに直接操作することが可能になる。

### 3. 提案手法

本論文では、Go-Go Interaction Technique [2]の考え方を利用し、スマートフォンの画面上に表示されたカーソルを非線形関数にしたがって操作する仮想タッチパッドを提案する。これにより、Lai ら [1]の実験で使用された、線形関数を用いた従来の仮想タッチパッドよりもエラー率を減少させることを目的とする。

指の移動距離に対するカーソルの移動距離を決める非線形関数は、以下の2つである。1つ目の非線形関数①は、指の移動距離 $d$ 、カーソルの移動距離 $f(d)$ 、カーソルが指の移動距離より大きく動き始める距離 $D$ を用いて以下の式で表される。

$$f(d) = \begin{cases} d & , d \leq D \\ d + k(d - D)^2 & , d > D \end{cases}$$

ここで、 $k$ は係数で、 $0 < k < 1$ である。この関数を図1(a)に示す。この式は、Go-Go Interaction Techniqueで使用された式を仮想タッチパッドによるカーソル移動に適用したものである。 $D$ に達するまでは、カーソルの移動距離は指の移動距離と等しいが、 $D$ を超えるとカーソルの移動距離は指の移動距離より大きくなり、その比率は指の移動距離が大きくなるほど大きい。

2つ目の非線形関数②は、カーソルが指の移動距離より大きく動き始める距離 $D_1$ 、カーソルが指の移動距離より大きく動き終わる距離 $D_2$ を用いて以下の式で表される。

$$f(d) = \begin{cases} d & , d \leq D_1 \\ d + k(d - D_1)^2 & , d > D_1 \wedge d \leq \frac{D_1 + D_2}{2} \\ d - k(d - D_2)^2 + a & , d > \frac{D_1 + D_2}{2} \wedge d \leq D_2 \\ d + a & , d > D_2 \end{cases}$$

ここで、 $a$ は、 $D_1$ と $D_1 \sim D_2$ の midpoint の2次関数と、 $D_2$ と $D_1 \sim D_2$ の midpoint の2次関数が、 $D_1 \sim D_2$ の midpoint で接するときの値、つまり、以下の式を満たす値である。

$$a = k \left( 2 \left( \frac{D_1 + D_2}{2} \right)^2 - 2D_1 \left( \frac{D_1 + D_2}{2} \right) - 2D_2 \left( \frac{D_1 + D_2}{2} \right) + D_1^2 + D_2^2 \right)$$

この関数を図1(b)に示す。 $D_1$ を超えるとカーソルの移動距離が指の移動距離より大きくなる場所は1つ目の関数と同様である。しかし、 $D_1 \sim D_2$ の midpoint を超えるとカーソルの移動距離は指の移動距離に近づいていき、 $D_2$ を超えると再び指の移動距離と等しくなる。それに伴い、1つ目の関数よりも係数 $k$ が大きくなり、指の移動距離に対するカーソルの移動距離の増加率が大きくなる。

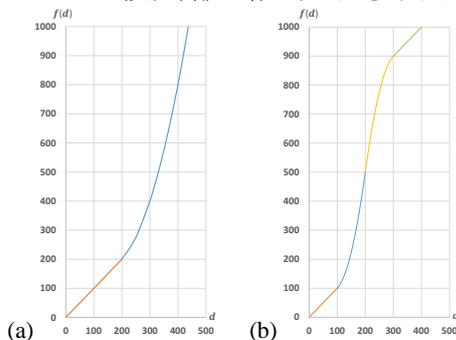


図1 (a)非線形関数①, (b)非線形関数②

提案手法の画面を図2(a)に示す。(1)が仮想タッチパッドであり、この範囲内を指でスライドすることで(2)のカーソルを移動できる。指を離れたとき、カーソルの位置にタッチイベントが発生する。しかし、タッチパッドの範囲外で指を離れた場合は、タッチイベントは発生しない。タッチパッドの範囲内の部分を選択したい場合には、通常通りタップすることでタップした位置にタッチイベントが発生する。(3)をタップすると1つ前の画面に戻る。(4)を長押しし、指をスライドさせることで、タッチパッドの大きさを自由に変えることができる。

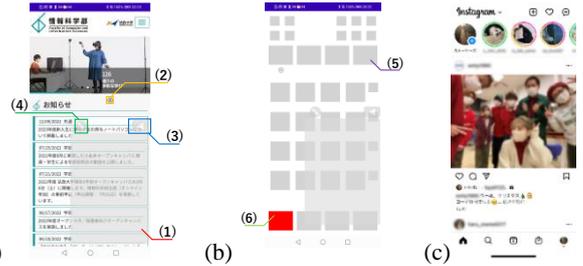


図2 (a) Web サイトの操作に提案手法を使用した画面, (b)比較実験で使用したアプリケーションの画面, (c) Instagram の画面

### 4. 実装

仮想タッチパッドの実装および、実験に使用するアプリケーションの実装には、Android Studioを使用した。使用言語はJavaである。実装した仮想タッチパッドは、作成したアプリケーション内のみで動作する。Web サイトの操作においては、AndroidにおいてWebコンテンツを表示させる機能をもつWebView [3]を使用することで、開発したアプリケーション内で操作を可能にした。また、実験で使用したそれぞれの仮想タッチパッドに対応したアプリケーションを作成した。

### 5. 予備実験

2つの非線形関数における、カーソルが指の移動距離より大きく動き始める、または、大きく動き終わる適切な距離の値と指を離れた時にカーソルの位置をリセットする必要性を調べるために予備実験を行った。この実験では、2つの非線形関数において、2パターンの値を設定する。さらに、それぞれのパターンについてカーソルの座標リセットを行うものと行わないもの、計8種類の仮想タッチパッド(表1)を使用した。

表1 予備実験で使用した8種類の仮想タッチパッド

	非線形関数	$D_1(D)$	$D_2$	カーソルリセット
1	①	200	×	なし
2	①	200	×	あり
3	①	300	×	なし
4	①	300	×	あり
5	②	100	300	なし
6	②	100	300	あり
7	②	200	400	なし
8	②	200	400	あり

### 5.1. 実験手順

実験には Huawei P30 lite (画面サイズ 6.2 インチ) を使用し、8 名 (男性 6 名, 女性 2 名) の被験者に協力してもらった。被験者は全員右利きで、平均年齢は 21.8 歳である。被験者には条件の異なる 8 種類の仮想タッチパッドを使用して、画面上に均等に表示されている 24 個のターゲットを各 2 回ずつ計 48 回、指定された順番で選択してもらった。順番自体はランダムだが、仮想タッチパッドごとに異なっており、選択する順番による選択時間に影響がでないようにした。結果として、平均選択時間と平均エラー率を算出した。また、実験後に正確さ、単純さ、親指の負担の少なさ、全体的な満足度の 4 項目において 5 段階評価を得るためのアンケートを実施し、ユーザー評価を取得した。

### 5.2. 実験結果

平均選択時間は、カーソルの座標リセットありのものがすべて 1210ms 以下、なしのものがすべて 1300ms 以上だった。平均エラー率は、非線形関数②を使用し、カーソルの座標リセットなしのものが 9% を超えていた。ユーザー評価は、ほとんどの項目においてカーソルの座標リセットありのものがなしのものより高い評価を得ていた。また、カーソルの動きが変化し始める値において、非線形関数①では 200 のもの、非線形関数②では 100 のものが全体的に良い結果を得ていた。

## 6. 比較実験

直接タッチすることによって選択する通常操作方法と提案手法による操作方法を比較し、提案手法の有用性を調べるためにこの実験を行った。この実験で使用する提案手法は、予備実験で結果が芳しくなかった非線形関数②を使用し、カーソルの座標リセットなしのものを除く、3 種類の仮想タッチパッド (表 2) である。

表 2 評価実験で使用した 3 種類の仮想タッチパッド

	非線形関数	$D_1(D)$	$D_2$	カーソルリセット
1	①	200	×	なし
2	①	200	×	あり
3	②	100	300	あり

### 6.1. 実験手順

図 2(b) のようなアプリケーションを用いて実験を行い、提案手法の有用性を調べた。予備実験では、24 個のターゲットを均等に表示させていたが、一般的なアプリケーションでは選択すべきターゲットの位置が均等とは限らない。現在のタッチ式 UI では、画面の角や端に近い部分を使用しているものが多い [4]。画面上部の角には複数の小さい UI が配置されている、画面上部もしくは画面下部に表示させる画面を変更するための UI が横一列に配置されている、という共通点があるが、LINE や Twitter, Instagram (図 2(c)) のような頻繁に利用するアプリケーションに見られる。また、Instagram や YouTube には画面中央の右端にも小さい UI が存在する。

図 2(b) で (5) がターゲットであり、被験者には (6) のような赤いターゲットを選択してもらった。選択する順番に関

しては、予備実験と同様である。結果として、平均選択時間と平均エラー率を算出した、また、タッチパッドごとに得手不得手を調べるため、ターゲットごとに平均選択時間や平均エラー率も算出する。

実験には Huawei P30 lite (画面サイズ 6.2 インチ) を使用し、右利きの 8 名 (男性 6 名, 女性 2 名)、平均年齢は 22 歳の被験者に協力してもらった。

### 6.2. 実験結果

実験結果を図 3 に示す。全体の平均選択時間に関しては、カーソル座標リセットありの仮想タッチパッドの約 1090ms、平均エラー率は直接タッチの通常操作の約 6% という結果が最も良かった。ターゲットの各位置における平均選択時間に関しては、画面左上の角、画面中央の右端、画面下部において、すべての仮想タッチパッドによる操作が通常操作より短い選択時間であった。ターゲットの各位置における平均エラー率に関しては、画面左上と右上の角では非線形関数②を使用しているカーソル座標リセットありの仮想タッチパッド、画面中央の上部では非線形関数①を使用しているカーソルリセット座標なしの仮想タッチパッド、画面中央の右端ではすべての仮想タッチパッドが、通常操作よりも低いエラー率であった。

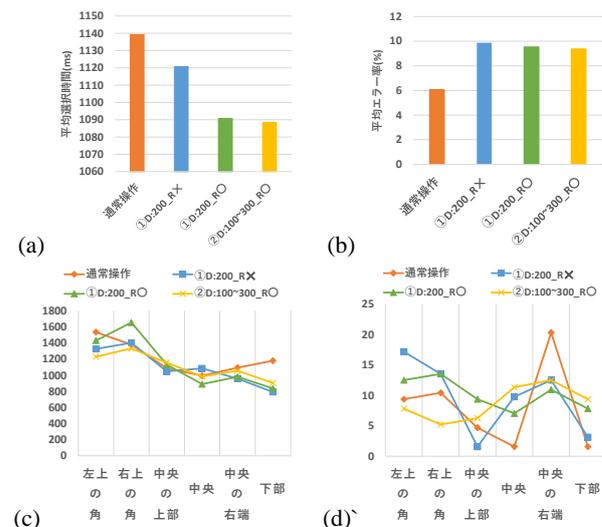


図 3 (a)全体の平均選択時間, (b)全体の平均エラー率, (c)ターゲットの各位置による平均選択時間, (d)ターゲットの各位置による平均エラー率

## 7. 評価実験

日常で実際に行う操作の 1 つである Web サイトの操作を、提案手法を用いて行ってもらい、その評価を行った。使用した提案手法は、比較実験で使用した 3 種類の仮想タッチパッドである。

### 7.1. 実験手順

WebView [3] を使用し、アプリケーション内に Web ページを表示させた。実験機器や被験者は比較実験と同様

である。実際の画面は図 2(a)である。被験者には、比較実験で使用した 3 種類の仮想タッチパッドを用いて自由にネットサーフィンをしてもらった。5 分間操作した後に、正確さ、単純さ、親指の負担の少なさ、全体的な満足度の 4 項目において 5 段階で評価してもらった。

## 7.2. 実験結果

実験後、得られたユーザー評価を図 4 に示す。正確さは、手慣れている通常操作が 4.5 以上と最も高い評価を得ていた。しかし、親指の負担の少なさについては、すべての仮想タッチパッドが 4 以上と通常操作よりも高い評価を得た。全体的な満足度は非線形関数②を使用しているカーソル座標リセットありの仮想タッチパッドが 4.25 と最も良い結果を得た。

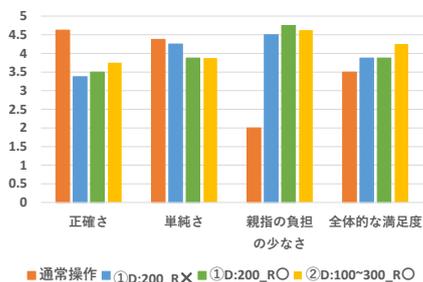


図 4 ユーザー評価

## 8. 議論

### 8.1. 比較実験について

比較実験の結果、全体の平均選択時間においては、すべての仮想タッチパッドが通常操作よりも短かった。中でもカーソル座標リセットのものは通常操作よりも 50ms 程短い結果となった。仮想タッチパッドを使用することで指先が届かない範囲を選択する際に持ち替えを必要としないため、選択時間が短くなったと考えられる。また、カーソル座標リセットがあることで、被験者は指の移動距離に対するカーソルの移動距離を把握しやすくなり、さらに選択時間が短くなったと考えられる。ターゲットの各位置における平均選択時間に注目すると、画面左上の角、画面中央の右端、画面下部において、すべての仮想タッチパッドによる操作が通常操作より短い選択時間であった。通常操作は、指先の届かない範囲を選択する際に持ち替えが必要となる。そのため、画面端の選択時間は仮想タッチパッドよりも通常操作の方が長くなったと考えられる。

全体の平均エラー率に関しては、通常操作が約 6%、すべての仮想タッチパッドが 9 から 10% であった。直接画面をタッチする通常操作は選択する場所を誤ることは少ないからである。しかし、非線形関数②を使用しているカーソル座標リセットありの仮想タッチパッドは画面の角付近ではカーソルは指と同じ動作をするため、持ち替えが必要となりやすい通常操作よりもエラー率が低い。また、非線形関数①を使用しているカーソル座標リセットなしの仮想タッチパッドはカーソルが画面上部に位置することが多いため、画面中央の上部におけるエラー率

が低い。画面中央の右端において、通常操作のエラー率が最も高い要因は、持ち替えたことにより、ターゲットの位置が手で隠れてしまうからだと考えられる。

### 8.2. 評価実験について

評価実験の結果、正確さにおいては通常操作が優れていると考える被験者が多かった。しかし、親指の負担という面では、持ち替えが必要となる通常操作は 2 と評価が低く、持ち替えが必要でない仮想タッチパッドはすべて 4 以上と高い評価を得た。全体的な満足度では、非線形関数②を使用しているカーソル座標リセットありの仮想タッチパッドが 4 を超える最も高い評価を得た。小さいため選択しにくいターゲットが配置されている画面の角付近にカーソルを移動させる際に、カーソルが指と同じ動作をするため使いやすさという被験者が多かった。そのため、非線形関数①を使用している 2 つの仮想タッチパッドよりも評価が高かったと考えられる。

## 9. おわりに

本論文では、大画面スマートフォンの片手操作を補助する機能として、非線形関数を利用した仮想タッチパッドによる操作方法を提案した。直接タッチによる通常操作との比較実験の結果、平均選択時間を短縮することができた。また、ユーザー評価により、親指の負担を大幅に軽減できたことを確認した。しかし、平均エラー率に関しては、直接タッチによる通常操作がすべての仮想タッチパッドより良い結果を得ていた。比較実験及び評価実験で使用した 3 種類の仮想タッチパッドの内、非線形関数②を使用しているカーソル座標リセットありの仮想タッチパッドが最も良い結果を示した。

今後の課題として、提案手法のエラー率を下げることで挙げられる。解決策として、ユーザーが選択することのできるターゲットの位置を仮想タッチパッドのシステム側で読み込むことで、選択範囲の小さいターゲット付近では自動的にカーソルがそのターゲットに引き寄せられるような仮想タッチパッドを設計することが考えられる。

## 文 献

- [1] Y. R. Lai and T. P. Hwang, "Virtual Touchpad for Cursor Control of Touchscreen Thumb Operation in the Mobile Context," *Proc. HCI Intl., LNCS*, vol. 9187, pp. 563-574, 2015.
- [2] I. Poupyrev, M. Billingham, S. Weghorst and T. Ichikawa, "The Go-Go Interaction Technique: Non-Linear Mapping for Direct Manipulation in VR," *Proc. ACM UIST*, pp. 79-80, 1996.
- [3] "Build Web Apps in WebView," [Online]. Available: <https://developer.android.com/develop/ui/views/layout/webapps/webview>. [Accessed 14 12 2022].
- [4] N.-H. Yu, D.-Y. Huang, J.-J. Hsu and Y.-P. Hung, "Rapid Selection of Hard-to-Access Targets by Thumb on Mobile Touch-Screens," *Proc. ACM MobileHCI*, pp. 400-403, 2013.