

# LoopTouch: 画面ループを用いたモバイル端末片手操作手法

土佐 伸一郎<sup>1,a)</sup> 田中 二郎<sup>2,b)</sup>

**概要:** スマートフォンに代表されるタッチパネルを搭載したモバイル端末の画面操作領域が拡大している。このようなモバイル端末は画面情報量が多いが、片手操作が困難になるという問題がある。この原因は、現在のモバイル端末のタッチインタフェースが画面上のオブジェクトに直接触れて操作するという指直接操作体系を取っている一方、片手操作時には親指可動領域外に GUI が配置されてしまうことにあると我々は考える。そこで我々は画面のループを用いる事によって、指直接操作の操作体系を残したまま親指可動領域外の GUI を操作可能にするインタラクション LoopTouch を提案、実装した。また、LoopTouch を適用した場合の GUI 操作についての評価実験を行った。

## LoopTouch: One-handed operation technique using the screen looping on mobile devices

TOSA SHINICHIRO<sup>1,a)</sup> TANAKA JIRO<sup>2,b)</sup>

**Abstract:** The operation area of smart phones equipped with touch panels is expanding. Such mobile devices' displays have a large amount of screen information, but they are difficult to operate with one hand. The reason for this is that while current touch interfaces operate by touching the object directly on the screen, when we perform operations with one hand, the GUI is placed outside the movable area of the thumb. We proposed an interaction technique called LoopTouch. By using a loop of the screen, this interaction enables users to operate the GUI placed outside the movable area of the thumb, keeping the operation system of directly touching the object on the screen. Furthermore, we carried out evaluation experiments for GUI operation when applying LoopTouch.

### 1. はじめに

スマートフォンに代表されるタッチパネルを搭載したモバイル端末の操作方法は、大きく分けて片手操作と両手操作に分類可能である。ここで述べる片手操作と両手操作の定義は以下である。

**片手操作** 片方の手でモバイル端末を把持し、その手の操作指(親指)で操作

**両手操作** 片方の手でモバイル端末を把持しもう片方の手の指で操作、もしくは、両手でモバイル端末を把持し

### 両方の手の指で端末を操作

両手操作ではモバイル端末画面の好きな位置に自由に触れることや、複数の指を利用したマルチタッチ操作を行うことが可能であり、一般的には片手操作よりも快適なモバイル端末操作が可能である。しかし、実際モバイル端末を利用すると両手では操作が出来ない状況が多々存在する。例えば、勤務先に立ったまま電車で通勤するという状況を考える。そのような時に片手操作が可能ならば、もう一方の手に鞆を持ちながらもモバイル端末でメールやドキュメントを確認することが出来る。このように片手操作というのは日常生活では自然に利用される操作であり、モバイル端末を利用する上では欠かすことの出来ない操作体系と言える。さらに Karlson らの調査 [1] では、大多数のユーザが「インタフェースが対応していれば、日常的に片手操

<sup>1</sup> 筑波大学大学院システム情報工学研究科コンピュータサイエンス専攻

<sup>2</sup> 筑波大学大学院システム情報系

<sup>a)</sup> tosa@iplab.cs.tsukuba.ac.jp

<sup>b)</sup> jiro@cs.tsukuba.ac.jp

作を行いたい」と回答したとされている。このような需要のために、マルチタッチ操作で頻りに利用される操作方法は、シングルタッチ操作で代替出来るように設計されていることが多い。例えば、2本指にてピンチ操作を行うと画面内容が拡大縮小するという操作がある。これはシングルタッチのダブルタップという操作に割り当てる事や画面下端に拡大縮小のソフトウェアボタンを表示させる事で代替手段とする事が一般的となっている。そのためゲーム等の一部のアプリケーションを除いて、マルチタッチ操作なしに、片手でのシングルタッチ操作にて十分に利用可能であるアプリケーションは非常に多い。

一方、近年画面操作領域の大きなモバイル端末の需要が大きくなっており、今後さらに増えるとされている [2]。しかし、このような画面操作領域の大きなモバイル端末での片手操作を行う際には、ある問題が発生する。それは、親指で触れにくい領域が存在するために、その領域の GUI に触れることが非常に困難になるということである。このような時は親指の長さが物理的に足りないために、両手操作を強いられたり無理な持ち替えを行うことになる。

このような問題の原因は、現在のモバイル端末のタッチインタフェースが画面上のオブジェクトに直接触れて操作 (以降、指直接操作) するという操作体系を取っている一方、片手操作時には親指可動領域外に GUI が配置されてしまうためであると我々は考える。そこで我々は両面からのタッチ入力が可能なモバイル端末を用いて、指直接操作の操作体系を残したまま親指可動領域外の GUI を操作可能にするインタラクション LoopTouch を提案、実装した (図 1)。本手法では端末の画面全体をループさせることにより、親指可動領域外の GUI を可動領域内に移動させる事が可能となる。現行のタッチインタフェースで採用されている指直接操作の操作体系をそのまま利用できることで、親指可動領域外に配置された様々なタイプの GUI が操作可能となる。

## 2. 関連研究

モバイル端末の片手操作のための操作手法は古くから研究されている。Rekimoto らはモバイル端末に搭載された加速度センサーを用いて、モバイル端末を操作する手法を示した [3]。また、モバイル端末にアナログ入力機構を取り付けて、ポインティング可能にする研究も存在する [4]。トラックパッドが背面に取り付けられたモバイル端末は近年商品化もされている [5]。また、ユーザの手から端末に加わる力をセンシングし、操作に利用する研究も成されている [6][7][8]。

### 2.1 特殊 GUI により片手操作を支援する研究

片手操作研究の中でも、画面レイアウトの工夫や特殊 GUI を用いる事により、片手操作時における親指操作を支

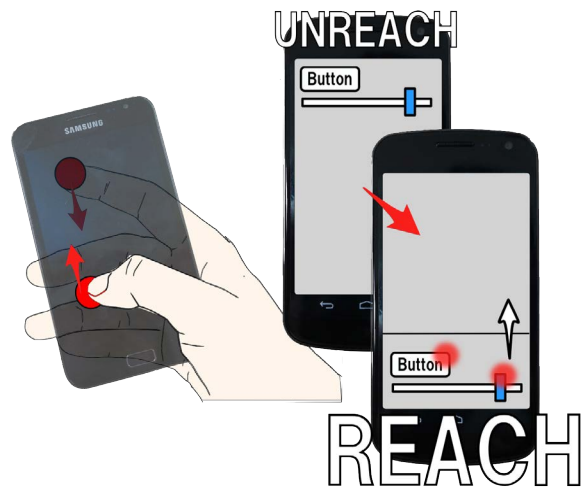


図 1 LoopTouch  
Fig. 1 LoopTouch

援する研究がいくつか成されている。

ThumbSpace[9]では、画面のスクリーンキャプチャを縮小したものを親指可動領域内に表示し、その縮小スクリーンキャプチャ上で目的の GUI が表示されている部分にタップし、大まかに選択エリアを指定する。その後は、親指の上下左右のスイープ操作により選択する GUI のフォーカスをずらして目的の GUI を選択できる。MagStick[10]では、指のスイープと逆方向に移動する特殊なカーソルを用いる事で、親指を容易に動かせる位置に置きながら目的の GUI を選択する事が可能となる。これらの研究はモバイル端末片手操作における親指操作の支援に着目している点、特殊な GUI を用いて親指での操作が困難な GUI への操作を容易にしている点で関連している。しかしこれらの研究で達成しているのは、カーソルによるポインティングや、GUI へのフォーカスの変更などを用いた、間接的な操作による GUI の選択にとどまっている。現行の Android や iOS における GUI の操作は、操作指そのものをポインティングデバイスとし、画面の GUI に直接触れて操作を行う指直接操作という体系をとっている。そのために、画面に別のカーソルを表示し GUI を選択するといったような、指位置と GUI の位置が離れた間接的な操作手法を用いるのは非常に不自然であると考えられる。本手法では画面上のあらゆる GUI に対して、親指による指直接操作を行える。また GUI の選択だけでなく、タップやスイープ等の既存のシングルタッチによる操作手法を行う事が可能なので、親指可動領域外の GUI に対してよりリッチな操作を行う事が可能となる。

## 3. LoopTouch

LoopTouch はロール操作と画面ループを用いた、モバイル端末の操作手法である。ユーザは両面からのタッチ入力が可能なモバイル端末を利用し、操作を行う。



図 2 ロール操作

Fig. 2 Roll operation.



図 3 画面ループ

Fig. 3 Screen looping.

### 3.1 ロール操作

端末の表面と裏面に対して、人指し指と親指の相対位置が近づく方向にスワイプする動作をロール操作と呼ぶ(図 2)。例えば図 2 の左のように、親指と人指し指の相対位置が端末に対して親指：下、人指し指：上の場合、親指は上方向に、人指し指は下方向に動かす操作である。操作可能方向は上下左右の 4 方向である。また、これに近い操作手法は [11] らの研究でも提案されているが、本研究では片手にてこの操作を行っている点で異なる。

### 3.2 画面ループ

画面ループはロール操作をトリガーとして起こる画面移動の事である。図 3 のように画面全体が指定方向に移動し、画面からはみ出た部分はその反対側から移動してくる。Karlson らの調査 [1] によると、PDA 片手操作時には親指の可動領域外である画面端への操作が困難であり、画面上端、左端の操作が特に困難であるとされている。本手法を用いる事で、例えば上方向に画面ループを行った場合、最上部に配置されていた GUI が画面下部から移動してくる事によって、親指を用いて容易に指直接操作する事が可能になる。画面ループする方向は上下左右の 4 方向であり、ロール操作時の親指の移動方向と一致する。上に画面移動してから、右に画面移動させるといった利用も可能である。目的の GUI を選択するか、あるいは画面をダブルタップすることで、画面ループが終了し、ループが発生していな

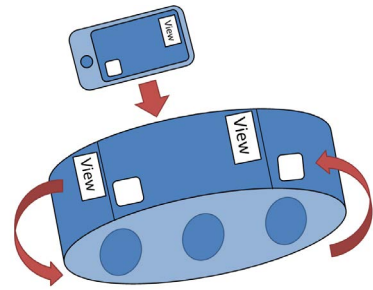


図 4 ベルトメタファの概念図

Fig. 4 Conceptual diagram of belt-metapha.

い初期の画面に戻る。

### 3.3 ベルトメタファを用いた端末操作と画面効果の自然な対応

ベルトメタファのイメージを図 4 に示す。本研究では両面からのタッチ入力可能なモバイル端末でインタラクションを行う。従来の片面のみのタッチインタフェースでは、ユーザは二次元平面に対してタッチ操作行っていた。しかし、表と裏の両面に対してタッチ操作を行おうとする場合、ユーザは端末という三次元物体を意識しつつ操作を行う事になる。つまり、単純に表と裏の 2 つの二次元平面に対する操作というよりは、三次元物体に対する操作という感覚でユーザは操作を行うと考える。本研究ではメタファとして三次元物体である「ベルトコンベア」を取り入れた。デバイス自身を「ベルトコンベア」と見立てて、その両面からベルトをスライドさせ回転させるイメージで操作する事が可能である。また、ベルトは平面がつながって形成されており、ループし続けるものである。そのため、このベルトメタファにより「ロール操作を行うと、画面が回転してループする」という端末操作と画面効果の対応付けは自然であると考えられる。

### 3.4 画面ループによる指直接操作

本手法では親指可動領域外の GUI に対して、画面ループによる指直接操作が可能である。指直接操作が可能という事はつまり、従来のタッチインタフェースで使用されているシングルタッチによるあらゆる操作(タップ、ダブルタップ、ロングタップ、スワイプ…etc)が可能になるということである。そのために、ただ GUI をタップして選択するだけでなく、スライダーをスワイプでスライドさせたり、オブジェクトにロングタップして、ドラッグアンドドロップすることも可能である(図 5)。これにより、例えば画面下部に配置されたアイコンを、画面をまたいで画面上部にドラッグアンドドロップすることが可能となる。画面サイズが大きなモバイル端末を片手操作する際に、画面下部から画面上部にオブジェクトをドラッグアンドドロップすることは本来ならば非常に困難な動作であるが、本手法

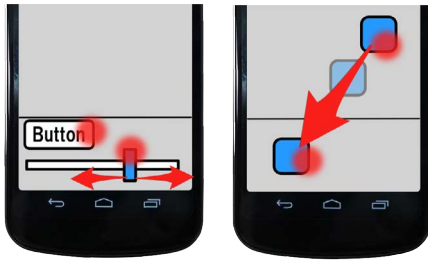


図 5 指直接操作

Fig. 5 Finger-direct operation.

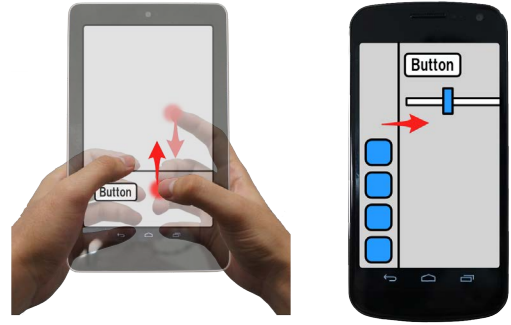


図 6 左:タブレット端末への応用 右:ロール操作による画面拡張  
Fig. 6 Left : apply to tablet-type device. Right : screen expansion by rolling operation.

を用いる事によって親指が容易に動作可能な位置で行う事が可能となる。

### 3.5 本手法の適用範囲

本手法はタッチインタフェース環境において、親指可動領域外にシングルタッチ操作が可能な GUI が配置されている全ての場合において適用可能であると考えられる。具体的には Android や iOS といったようなモバイル端末向けの OS で見られるネイティブ GUI や、web ページにおける GUI に対して適応する事が可能である。例えば Android の UI ガイドライン<sup>\*1</sup>によると、画面最上部にメニューを設置する事は一般的である。また、モバイル端末でブラウザアプリケーションにて web ページを閲覧することは多く、その web ページの画面上部にリンクやボタンなどの GUI が配置されていることがある。以上のように親指の可動領域外とされる画面上部に GUI が配置されている例は数多く存在する。このような場合に本手法を適用する事で、親指で容易に指直接操作することが可能となる。

#### 3.5.1 応用的利用

本手法はスマートフォンに代表されるモバイル端末の片手操作時の操作を円滑にするものであり、両手操作については考慮していない。しかし応用的な利用として、両手によるタブレット端末操作時にも適用可能であると考えられる。図 6 左のように端末を両手で把持した場合、端末上部の GUI に触れるためにはタブレットの持ち替えが発生する。タブレット端末は小型端末と比べ重量があり画面サイズが非常に大きいので、画面上部へ手を移動した際の腕や手への負担は小さいものではない。しかし、本手法をタブレット端末に応用すると、端末下部を常に把持したまま操作を行う事が可能となると考える。また、ロール操作は画面を「ずらす」操作である。そのため、ずれた領域分画面を拡張する利用方法も可能だと考える。例えば、図 6 右のようにアプリケーション固有のメニューを表示させるという例が挙げられる。

### 3.6 現行の操作手法との共存

現行のスマートフォンに代表されるモバイル端末の GUI 操作には、タップ、ダブルタップ、スワイプ等いくつか存在する。これらの操作の内、操作と結果の対応付けが一般的に浸透しているものがある。例えば、画面をダブルタップすることでブラウザの画面が拡大する、スワイプを行うと画面がスクロールしたり、異なる画面に移移する等である。これらの操作は一般的に浸透しているために、これら操作を行ったときに期待しているものと異なる結果が生じるのはユーザにとって不親切である。本手法で提案するロール操作は両面入力を用いることで、このような既存の操作手法と共存しつつ、競合せずに組み合わせて利用する事が可能であると考えられる。例えば、写真を閲覧するアプリケーションにて画面上部にメニューが配置されているという場合を考える。ある画像の閲覧中に画像を拡大するためダブルタップすると友人が映っているのに気付いた。そこで画面全体をロール操作によってずらし、画面上部のメニューに配置されたボタンをタッチして友人にその画像を送信する、といったような既存の操作手法と自然に組み合わせた利用が可能となる。

## 4. プロトタイプデバイス

提案するインタラクションが可能な簡易的なプロトタイプデバイスを実装した(図 7)。Android 端末 2 台 (Samsung Galaxy Note 147(H) x 83(W) x 9.7(D) mm 5.3 インチ, Google Nexus S 123.9(H) x 63(W) x 10.9(D) mm 4 インチ) から構成される。プロトタイプのサイズは 147(H) x 83(W) x 21.1(D) mm, 重さ 329.8g であった。本研究では画面サイズの大きなモバイル端末を対象とするため、一般的に大きなスマートフォンに分類される 4.5 インチから 5 インチ程度のモバイル端末がプロトタイプ作成に適切だと考えた。さらに、端末両面に対するタッチ入力が可能である必要がある。

両面からのタッチ入力が可能なモバイル端末を用いた研究は近年注目されており、モバイルゲーム機として商品化

\*1 <http://developer.android.com/index.html>





図 7 プロトタイプデバイス  
Fig. 7 Prototype device

されているものもある\*1。背面からの入力を可能にすることで、操作指によるオクルージョンを防ぐ [12][13][14] ものや、背面指を操作する事によって文字入力を行うもの [15] が存在する。今後両面タッチ入力可能なモバイル端末は増加するであろうと考えられるため、本研究ではそのようなモバイル端末を操作デバイスとする。

Android 端末を二台背中合わせに重ね合わせる事によって、簡易的にプロトタイプを作成した。このプロトタイプは右手操作用に作成した。利用した端末は Galaxy Note(表面) と Galaxy S(裏面) である。異なる端末を利用したのは軽量化を図るためである。これらを背中合わせに重ね合わせ、隙間は軽量紙粘土を用いて埋めている。また、重ね合わせる際には図 7 のように表側の端末に対して裏側の端末を、左端と下端を合わせるように配置した。下端を合わせた理由としては、現行の多くのモバイル端末は端末操作時に頻繁に押下する事となるハードウェアキーやソフトウェアキーが端末下部に配置されているため、ユーザは端末の下部を把持して持つ事が多いからである。左端を合わせた理由は、端末を右手で把持する際には端末側親指以外の背面指は端末背面左部に位置することが多いためである。端末間のデータ送受信には Bluetooth を用いている。

## 5. インタラクション設計と実装

### 5.1 ロール操作

ロール操作時には表面を親指、裏面を人指し指にて操作する。背面の操作指として人指し指を選んだ理由としては、Wobbrock らの研究 [16] によると人指し指は背面とほぼ同様の入力パフォーマンスを持つとされ、背面指の中でも最も動かしやすい指だと考えたためである。背面の接触指の情報を、表面の端末に Bluetooth で送信し続ける。背面には人指し、中指、薬指、小指と最大で 4 本の指が接触する。この背面接触指の中で、人指し指の動作を検知する必要がある。今回の実装では背面接触指の中でも、最上部の接触点を人指し指の接触点と見なす事とした。端末を把持する際に、手の構造上人指し指よりも上部にそれ以外の指が接触する事は通常あり得ないためこのような実装とした。

ロール操作を発生させるためには以下の全ての条件を満たす必要がある。以下で述べる ACTION\_MOVE イベントとは Android 端末のタッチイベントにおいて、親指の移動が発生した際に生じるイベントのことである。

- (1) Xmsec 内 (イベント監視区間) に背面最上部指の ACTION\_MOVE イベントが N 回連続発生
- (2) (1) が満たされてから Ymsec 以内に親指が画面最上部指と反対方向に ACTION\_MOVE する
- (3) 親指の接触位置と背面最上部指の接触位置が適切な位置に存在 (例: 上方向への画面ループならば、親指接触位置が背面最上部指の接触位置よりも下部に存在)

この X, Y, N の値は背面最上部指の動かしやすさ、つまり端末の大きさや厚さによって変更されるべきであると考えた。今回作成したプロトタイプでは設定した値は X:225, Y:50, N:2 とした。背面には基本的に常に指が触れた状態である。そのため、端末の持ち替え等によって背面最上部指の ACTION\_MOVE イベントが発生し、ロール操作が誤動作してしまう可能性が考えられる。上記の (2), (3) の条件により誤動作を軽減できるよう期待し、このような設計とした。

### 5.2 画面ループ

#### 5.2.1 画面ループの挙動

ロール操作をキーイベントとしてループ受付状態に遷移する。画面ループの方向は上下左右の 4 方向であり、親指の移動方向と一致する。ループ受付状態に遷移すると、親指の移動のみで画面ループの移動量を変化させることが可能である。ループ受付状態時に親指を画面から離すと受付状態が終了する。これは、ユーザの親指を離すという動作は画面ループ動作の完了であると判断できるためである。というのも、一度ループ受付状態に遷移すれば、親指のSwipe動作を用いて画面ループの移動量を自由に調整可能である。そして、触れたい GUI が期待の場所に移動したら親指を画面から離し、GUI を操作することになる。そのため親指を離すことを画面ループ動作の完了と見なすのは自然であると考ええる。

目的の GUI への操作が完了したら、自動的に画面が初期位置に戻ってくれるのが望ましいと考えた。今回の実装では目的の GUI への操作が完了する、もしくは画面をダブルタップすると初期位置に元に戻るようにした。目的の GUI への操作の完了の判別方法は、GUI の種類によって異なる。例えば、ボタンやリンクをタップすることはそのまま操作の完了と見なすこととした。またスライダーはSwipe操作が完了し、親指を画面から離れた時点で完了と見なすこととした。しかしトグルボタンやチェックボックス等の GUI に対する操作の場合、ユーザの操作がどこで完了したのかをシステム側で自動判別するのが困難である。そのため、このような GUI に対する操作の場合は画面を

\*1 <http://www.jp.playstation.com/psvita/>

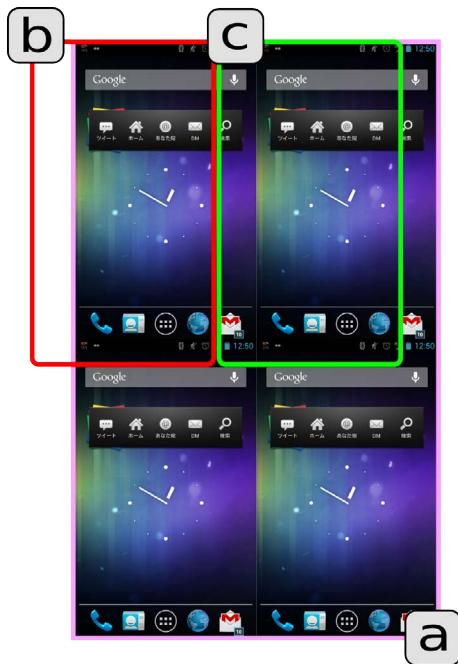


図 8 ループビュー  
Fig. 8 Loop view

自動で初期位置に戻さず、親指のダブルタップという明示的な操作により画面を戻せるようにした。

### 5.2.2 ループビュー

図 8 の a のようなループビューを生成している。ループビューは 4 つの領域から成っており、左上の領域をメイン領域と呼び、それ以外を複製領域と呼ぶ。複製領域にはメイン領域と同様の内容を表示している。画面ループが発生していない状態では、端末の画面にはメイン領域を表示している。画面ループ発生時にはこの表示領域をずらすことによりあたかも画面がループしているようにユーザに見せかけている。例えば、b 領域を画面に表示しようとした場合、c 領域を代わりに表示する事によって、ユーザにはループしているように見せかける事が可能である。上下も同様の方法で画面ループを実現する。複製画面で発生した各種イベントはメイン画面へと伝えられる。

## 6. ケーススタディ

### 6.1 実験目的

画面サイズの大きなモバイル端末片手操作時において、画面ループを用いる事によって、親指可動領域外の GUI を円滑に操作可能になるかどうかを検証する。そこで簡易プロトタイプ上で動作する、本手法 LoopTouch を適用した実験用のアプリケーションを作成し被験者に利用してもらった。実験用のアプリケーションは Android OS 固有の GUI を用いたアプリケーションである。利用後は、快適性と負担度に関するアンケートを行った。

### 6.2 実験内容

本手法の非適用時と適用時の二つを比較することにより検証を行う。非適用時は本手法を適用せずにタスクを行ってもらい、適用時は本手法を利用してタスクを行ってもらい。操作方法とプロトタイプデバイスに慣れるために、非適用時と適用時のタスクそれぞれを行う前に練習を行わせ、練習時間は自由とした。タスクの内容は簡易プロトタイプ上で動作するアプリケーションの GUI 操作である。本手法の適用対象として、画面上部に GUI が配置されているという状況を想定しているため、そのような状況を満たすアプリケーションを作成した (図 9)。特殊な GUI を除き、一般的な GUI はタッチ操作、あるいはスワイプ操作により操作を行う事が可能である。そのため今回の実験では配置する GUI をボタン、チェックボックス、シークバーの 3 タイプとした。ボタンとチェックボックスはタッチ操作の操作性を、シークバーはスワイプ操作の操作性を検証するために採用した。また、今回の実験ではロール操作は左右方向は行わず、上下方向のみを行うようにした。

図 9 のように画面上部に同タイプの GUI を 4 つ配置し、それらの内一つまたは複数個の GUI を操作するように指示した。タスクの種類は GUI のタイプと同様に 3 種類ある。ボタントラックについては画面上部に表示される 4 つのボタンの内、指定されたボタンをタップすることで完了する。チェックボックスタスクは画面上部に表示される 4 つのボタンの内、指定された 2 つをタップし選択状態にすることにより完了する。シークバーは画面上部に表示される 4 つのシークバーの内、指定された 1 つをスワイプ操作により指定の範囲まで動かしてから親指を画面から離す事で完了する。完了すると、画面下部に次のタスクへ進むボタン (以降、next ボタン) が表示され、それを押下する事で次のタスクへ進む事が出来る。画面下部に next ボタンを配置した理由は、画面下側をホームポジションとして保持する事を被験者に促すためである。というのも、現行のモバイル端末の多くは端末操作に頻繁に利用する事になるハードウェアキーやソフトウェアキーが画面下部に配置されているため、一般的に画面下側を保持して操作を行う事は一般的であるからである。タスクはタイプごとに 8 タスクずつ行い、合計で 24 タスクである。操作対象の GUI は均等かつ無作為に指定した。タスクの実行順はタイプごとに 8 タスクずつ行い (例: ボタントラック x8 → チェックボックスタスク x8 → シークバータスク x8)、被験者間での釣合いを取った。

### 6.3 結果

被験者は 10 人で内一人は女性であり、全員右利きであった。実験中はタスク実行の様子をビデオ撮影し、観察を行った。また実験後にアンケートを行った。非適用時と適用時それぞれについて、5 段階のリッカート尺度を用いて

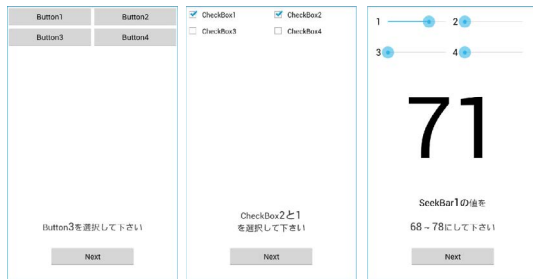


図 9 実験用アプリケーションの画面構成. 左:ボタンスク 中央:チェックボックススク 右:シークバースク

Fig. 9 Screen structure of the laboratory application. Left : task of the button. Center : task of the check box. Right : task of the seek bar.



図 10 実験時の様子

Fig. 10 The situation of the experimentation

GUI 操作の快適性 (1:快適ではない, 5:快適), 手への負担 (1:負担, 5:負担ではない) について回答させた. またそれぞれを選んだ理由についても記述させた. さらに適用時に関する自由記述欄も設けた.

5段階リッカート評価の結果を図 11 に示す. (a) が非適用時, (b) が適用時の結果である. (a), (b) 左図の横軸は評価点, 縦軸は被験者数である. (a), (b) 右図の横軸は快適性と負担度, 縦軸は全被験者の評価点の平均値である. アンケートでは快適性, 負担度ともに適用時の方が良い評価が得られた.

## 7. 考察

適用時に多く得られた快適性に関する意見として, 「非適用時は端末を落としそうになったが, 適用時は安定して把持したまま操作が可能だった」(5人) 「非適用時は操作の度に持ち替えを行い把持位置を変える必要があったが, 適用時は把持の仕方を変えずに操作を行えるので操作が容易であった」(5人) が得られた. これらの肯定的な評価は図 11 の 5段階評価の結果にも表れていることが分かる.

しかし一方で否定的なコメントとして「画面の左半分の GUI 操作に関しては, 親指位置まで GUI を動かしても操作は困難であった」(2人) という意見があった. 今回使用したプロトタイプは 5.3 インチの端末を利用しており, ス

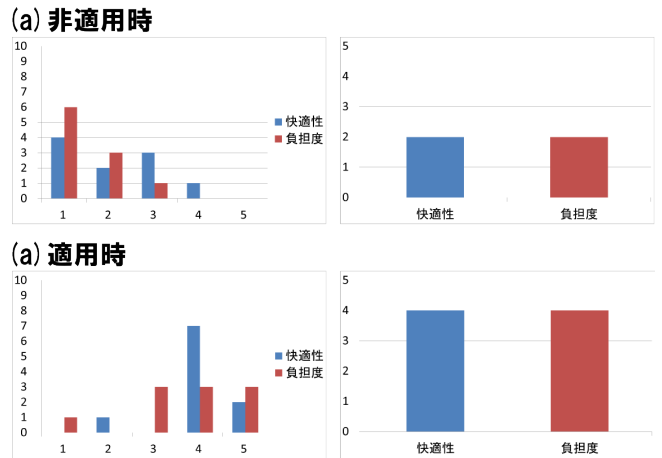


図 11 実験結果

Fig. 11 Experimental results

マートフォンの中でも非常に大きなものに分類される. そのため, 画面上部だけでなく画面の左端にも親指が届かない領域が存在してしまう. そのため同 GUI タイプのタスクでも, 画面右側の GUI を選択させる場合は容易に操作が出来ても, 画面左側の GUI を選択させる場合は操作が困難になる被験者が見受けられた. 今回の実験では上下方向の画面ループのみを実行可能としているが, 左右方向の画面ループを実行可能にする事によりこの問題は解決可能であると考えられる.

また, 「ボタン, シークバーは操作完了後に, 自動で画面が初期位置に戻るのはいけなく」(2人) という意見が得られた. 例えばボタンスクでは, 指示されたボタンを正しく選択すると, 画面下部に next ボタンが表示され, それと同時に画面がアニメーションし初期位置に戻る. この際に被験者は画面をループさせてタスクを行っているため, 画面上の表示では 4 つ配置された GUI のすぐ上部に next ボタンが表示されることになる. この next ボタンは親指でも容易に届く位置に配置される事になるので, 被験者はすぐにそのボタンをタップし次のタスクに進もうとする. しかし, そのボタンに触れようとする画面が自動的に初期位置に戻り選択操作を失敗してしまうために, 不快感を感じてしまうと考えられる. チェックボックススクについては操作が完了しても, 自動で初期位置には戻らないのでそのような意見は得られなかった. これらから, ボタンならば画面を初期位置に戻しチェックボックスならば戻さないといったように GUI ごとに動作を決めるのではなく, GUI 操作後のコンテキストに応じて初期位置に戻すかどうかを決めるべきであると考えられる. 例を挙げると, 今回のタスクのように GUI を選択後にさらに連続して任意 GUI 操作を実行したい場合は画面を初期位置に戻す事は不快感に繋がる可能性がある. しかし, ブラウザ上のリンクのように GUI 選択後に画面レイアウトが大きく変化するものや, 次の画面が表示されるまでに時間がかかるものに対し

ては、自動的に画面を初期位置に戻す事はユーザに不快感をもたらすものではないと考える。これについては今後の実験でさらに調査を行う必要がある。

負担度に関しては、「非適用時は持ち替えが発生する事により手に負担が生じたが、適用時は把持位置を変える必要がなく指の移動量が少なかったため負担が少なかった」(6人)という意見が多く得られた。一方、「背面指の操作に負担を感じた」(2人)「端末の厚さ、重さのために適用時、非適用時どちらも負担を感じた」(3人)との意見があった。そのため今後はより薄く、軽量のプロトタイプを作成する必要があると考えられる。それにより、端末の把持が行いやすくなり背面指の操作の負担が軽減されると考える。

## 8. まとめと今後の課題

評価実験後のアンケートの結果、快適性、負担度について適用時の方が良い傾向にあることが分かった。そして適用時の方が、親指可動領域外のGUIを円滑に指直接操作できることが確認された。しかし一方で、画面の幅が大きな場合、画面左端に寄ったGUIに関しては上下方向のループのみでは完全に対応する事が出来ない事が分かった。画面ループを行い、GUIを選択したあとの挙動に関しても今後考察を踏まえて設計を改める必要があると考える。そして、今回利用したプロトタイプが快適性と負担度に少なからず悪い影響を与えている事が、アンケートにて見受けられた。そのため今後はプロトタイプデバイスの薄型化、軽量化を図る。

また、本研究で提案するLoopTouchは画面ループとロール操作が組み合わせり実現されるものである。そのため、今後は上下左右方向のロール操作の精度に関する実験を行う必要があると考えられる。その際には背面に指が触れることによる誤動作や、現行の操作手法との競合という点に着目した実験設計を行う必要があると考える。

### 参考文献

- [1] Amy K. Karlson and Benjamin B. Bederson. Understanding single-handed mobile device interaction. Technical report, 2006.
- [2] Paul Brown. Smartphone owners want thin devices with larger displays. Technical report, Wireless Device Lab, 2012.
- [3] Jun Rekimoto. Tilting operations for small screen interfaces. In *Proceedings of the 9th annual ACM symposium on User interface software and technology*, UIST '96, pp. 167–168, New York, NY, USA, 1996. ACM.
- [4] 清久仁河内谷, 浩石川. 超小型機器の「指一本操作」のための入力機構. 全国大会講演論文集, Vol. 54, No. 4, pp. 111–112, 1997.
- [5] MOTOROLA. : Motorola charm. 入手先 <[https://motorola-global-portal.custhelp.com/app/product\\_page/faqs/p/30,6720,7489](https://motorola-global-portal.custhelp.com/app/product_page/faqs/p/30,6720,7489)>(参照 2012-10-25).
- [6] Mayank Goel, Jacob Wobbrock, and Shwetak Patel. Grippsense: using built-in sensors to detect hand posture and pressure on commodity mobile phones. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*, UIST '12, pp. 545–554, New York, NY, USA, 2012. ACM.
- [7] Craig Stewart, Eve Hoggan, Laura Haverinen, Hugues Salamin, and Giulio Jacucci. An exploration of inadvertent variations in mobile pressure input. In *Proceedings of the 14th international conference on Human-computer interaction with mobile devices and services*, MobileHCI '12, pp. 35–38, New York, NY, USA, 2012. ACM.
- [8] NTT DOCOMO. : Grip ui. 入手先 <<http://docomo-exhibition.jp/cj2012/pc/index.html>>(参照 2012-10-25).
- [9] Amy K. Karlson and Benjamin B. Bederson. Thumb-space: generalized one-handed input for touchscreen-based mobile devices. In *Proc. INTERACT 2007*, pp. 324–338. Springer, 2007.
- [10] Anne Roudaut, Stéphane Huot, and Eric Lecolinet. Tap-tap and magstick: improving one-handed target acquisition on small touch-screens. In *Proceedings of the working conference on Advanced visual interfaces*, AVI '08, pp. 146–153, New York, NY, USA, 2008. ACM.
- [11] Erh-li Early Shen, Sung-sheng Daniel Tsai, Hao-hua Chu, Yung-jen Jane Hsu, and Chi-wen Euro Chen. Double-side multi-touch input for mobile devices. In *CHI '09 extended abstracts on Human factors in computing systems*, CHI EA '09, pp. 4339–4344, New York, NY, USA, 2009. ACM.
- [12] Daniel Wigdor, Clifton Forlines, Patrick Baudisch, John Barnwell, and Chia Shen. Lucid touch: a see-through mobile device. In *Proceedings of the 20th annual ACM symposium on User interface software and technology*, UIST '07, pp. 269–278, New York, NY, USA, 2007. ACM.
- [13] Patrick Baudisch and Gerry Chu. Back-of-device interaction allows creating very small touch devices. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '09, pp. 1923–1932, New York, NY, USA, 2009. ACM.
- [14] Tomoko Ohtani, Tomoko Hashida, Yasuaki Kakehi, and Takeshi Naemura. Comparison of front touch and back touch while using transparent double-sided touch display. In *ACM SIGGRAPH 2011 Posters*, SIGGRAPH '11, pp. 42:1–42:1, New York, NY, USA, 2011. ACM.
- [15] Kevin A. Li, Patrick Baudisch, and Ken Hinckley. Blind-sight: eyes-free access to mobile phones. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '08, pp. 1389–1398, New York, NY, USA, 2008. ACM.
- [16] Jacob O. Wobbrock, Brad A. Myers, and Htet Htet Aung. The performance of hand postures in front- and back-of-device interaction for mobile computing. *Int. J. Hum.-Comput. Stud.*, Vol. 66, No. 12, pp. 857–875, December 2008.