

大画面を備える携帯情報端末を安定して 操作するための片手親指操作手法

大西 主紗 志築 文太郎 田中 二郎

大型のタッチ画面を備える携帯情報端末(以降, 大端末)を片手にて操作する場合, その大きさゆえに指が届かず, 操作が困難な画面領域が存在する. この時ユーザは片手のみを用いて端末の把持姿勢を変えるという煩雑な動作を行う必要があり, 不安定な把持及び端末の落下の原因となる. そこで我々は, 大端末向けの安定した片手親指操作手法である TouchOver を示す. TouchOver は画面下部にて発生したタッチイベントを画面上部に転送する. 画面上部に対する TouchOver と画面下部に対する直接操作とを使い分けることにより, 大端末に対する安定な操作, すなわち片手親指のみを用いた持ち替えを必要としない, 画面の全領域に対する操作を実現する. 本稿においては, 操作の切り替えにホームボタンのダブルタップを用いることにした. この動作の検出のため, 我々は Android 端末におけるホームボタンのダブルタップを検出するウィジェットを作成した. 提案手法の評価を行うため, 我々は TouchOver のプロトタイプを Android アプリケーションとして実装した. 本稿ではプロトタイプを用いた使用例及び, 性能評価のために行った比較実験の結果を示す.

When users use large mobile devices, which are equipped with a large touchscreen, with their single hand, the region distant from their thumb is too distant for users to control. Then, users have to change their hand posture to control the top half, which is a complex action causing an unstable grip. To solve this problem, we present TouchOver, a single-handed stable control technique using a thumb for large mobile devices. This technique solves the problem by transferring touch events on the bottom half of a touchscreen to the upper half. Using TouchOver for the top half and direct touch for the bottom half allows users to control a large mobile device stably, that is a control for all region of a touchscreen only using their thumb without changing the hand posture. We made a widget to make the Android smartphone to detect double tapping its home button, to provide a trigger to activate/deactivate TouchOver. We present the use cases and user studies for measuring the performance of TouchOver.

1 はじめに

Parhi らによれば, 多くのユーザは携帯情報端末の片手操作, すなわち端末を把持した手の親指のみを用いた操作を欲している[17]. なぜならば, 携帯情報端末のブラウザを用いて「ちょっと調べ物をする」際や, 携帯情報端末を音楽プレーヤとして使用する場合などは, 片手のみを用いた操作が非常に便利だからである. この欲求は, 傘やつり革などで片手がふさがって

いる場面(図 1a, b) や, 片手をけがして両手による操作が困難な場面に誰にでも生じるものである.

一方で, 大型のタッチ画面をもつ携帯情報端末(以降, 大端末)が市場に登場し, ユーザに使用されるようになった. 大端末にはその画面の大きさから一度に表示可能な情報量が増加するというメリットが存在す



図 1 a: 傘で片手がふさがっている場面. b: つり革につかまっている場面. c: 片手による不安定な端末の把持.

A Stable Single-handed Control Technique using a Thumb for Mobile Devices with a Large Touchscreen.

Kazusa Onishi, Buntarou Shizuki, and Jiro Tanaka, 筑波大学, University of Tsukuba.

コンピュータソフトウェア, Vol.33, No.1 (2016), pp.78-90.

[研究論文] 2015 年 3 月 3 日受付.

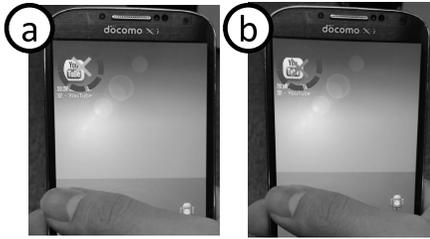


図 2 TouchOver 使用時に表示されるフィードバック。

る一方、片手によって大端末を把持している際にユーザの指が届かない画面領域に対して親指を用いて直接操作を行なうのが困難であるという問題が存在する [24]。この場合ユーザは片手のみを用いて大端末の把持姿勢を変える必要がある。しかし、その動作は煩雑であり、かつ不安定である (図 1c) ため端末の落下の原因となる (持ち替え問題)。特につり革を持っている場面においては、ユーザは揺れる車内にて大端末の把持姿勢を変える必要があり、この問題はより顕著となる。

本稿において、我々は大端末向けの片手親指操作手法である TouchOver を示す。この手法は大端末に対する安定した片手操作、すなわち片手親指のみを用いた、持ち替えを必要としない、画面の全領域に対する操作を可能とする手法である。本手法は図 2 に示すように、画面下部にて発生したタッチイベントを画面上部に転送する。画面上部に対する TouchOver と画面下部に対する従来通りの直接操作を使い分けることにより、ユーザの親指が届く大端末の画面下部のみを用いて画面の全領域に対する操作を可能とする。本手法を用いた場合、ユーザは大端末の把持姿勢を変えることなくポインティングを含む既存のタッチジェスチャ全般を画面の全領域に対して行うことが可能であるため、常に端末を安定して把持し続けることが可能である。

なお本手法はタッチイベントが転送される位置 (転送予想点) を、ホバーイベントを活用することにより分かりやすく表示する。具体的には、ホバーイベントが発生している際には転送予想点を緑色のカーソルとして表示し (図 2a)、画面がタッチされている際にはカーソルの色を赤色に変化させる (図 2b)。これにより

ユーザはタッチ前に「このままタッチすると画面上部のどこをタッチすることになるのか」、またタッチ後に「現在は画面上部のどこを操作していることになっているのか」を把握することが可能となるため、本手法は従来のカーソルを用いた手法と同様に操作を行うことができる。現在ホバー検出が可能なスマートフォンには ELUGA P P-03E, Galaxy S4 SC-04E, AQUOS Phone ZETA SH-06E, ARROWS NX F-06E などが存在するが、タッチイベントの転送自体にはホバーイベントを必要としない。したがって、フィードバックを用いることなく転送予想点を把握できる場合、例えば画面上にアイコンが規則正しく並んでいるホーム画面などにおいては、ホバー検出が不可能なスマートフォンにおいても本手法は使用可能である。

本手法においては、ユーザが画面の全領域に対する操作を画面下部にて行うため、画面上部への間接操作と画面下部への直接操作をユーザが使い分ける必要がある。そこで本稿においては、これらの操作の切り替え、すなわち TouchOver の起動/終了にホームボタンのダブルタップを用いることにした。

2 関連研究

大端末及び大画面デスクトップ環境などの大領域を対象とする操作手法が多く存在する。本節ではそれらの間接/直接操作手法と比較しながら、間接操作手法である TouchOver の立ち位置を述べる。また、本手法は TouchOver 及び直接操作を使い分けることによって全画面領域の操作を行うため、大端末及びタブレットなどのタッチパネル端末向けの操作切り替え手法についても述べる。

2.1 大領域向けの間接/直接操作手法

間接操作手法を用いて親指の届かない領域に対する操作を実現した手法が多く存在する。MagStic [20] は親指の動きと逆方向に移動し、ターゲットに吸い付くカーソルである。CornerSpace [24] は、BezelSwipe [19] を用いて出現させた UI によって出現位置を画面 4 隅の中から任意に選択できるカーソルである。ただし、これらの手法では、ユーザはタッチアップが行われた際にカーソルのある位置に対してポインティン

グを行うのみであり、ドラッグやロングタップなどのタッチジェスチャを行うことはできない。本手法はポインティングのみならず、ダブルタップやロングタップなどのタッチジェスチャ全般を大端末に対して使用可能とする。ExtendedThumb[16]も上記の手法と同様にドラッグによってカーソルを親指の届かない領域に対して移動させるが、カーソルを移動させタッチアップを行った後に続くタッチ操作がすべてカーソルの位置に対して転送される。これにより親指の届かない領域に対する操作を実現しているが、1つの操作に対してカーソルの移動及び領域に対する操作という2段階のタッチ操作を毎回行わなければならない。TouchOverも起動/終了、及び大端末の操作という2段階の操作を必要とするが、1度起動したTouchOverは再び終了させるまでタッチイベントを画面上部に対して転送し続ける。したがって操作対象が画面上部に集まっている場合はExtendedThumbよりも少ない操作量にて操作可能である。

一方、直接操作を補助する手法を用いて大端末の片手操作を実現した手法も存在する。ThumbSpace[12]は、大端末画面全体の縮小画像が表示されたタッチパッドを生成する。ユーザはこのタッチパッドに表示されたターゲットをタッチすることにより、指が直接届かない画面端のターゲットを選択することができる。ThumbSpaceは画面領域全体のタッチパッドを生成するのに対し、TouchOverは画面下部をタッチパッドとして用いることにより画面上部の操作を容易にしている。またDrag-and-PopやDrag-and-Pick[2]、Vacuum[3]は、画面上に存在するアイコンやウィンドウをユーザの近くに移動させる大画面デスクトップ環境向きの操作手法である。ただし、Drag-and-Pop及びDrag-and-Pickではアプリケーションアイコンに対するファイルアイコンのドロップやフォルダの展開などアイコンに対する操作しか行えず、Vacuumでは手元に移動したウィンドウの中からユーザが選択したものを手元に残すことしかできないため、ユーザは操作領域から遠い位置に表示されたUIの操作を行うことができない。LoopTouch[26]はロールジェスチャと呼ばれる、大端末の表面に触れている親指と背面に触れている人差し指の相対位置が近づくよう

なジェスチャを行うことにより、画面自体をループさせ操作対象を親指の近くに移動させる。Reachability[1]はiPhone 6や6 plusに実装されている機能であり、ホームボタンのダブルタップによって画面を下方方向に移動させ、指の届かない領域をユーザの指が届く位置まで移動させる。これらの手法は画面のレイアウトを変化させるものであるため、本来画面下部に配置されているアイコンが画面上部に移動するなど、ユーザが混乱させられる可能性がある。一方本研究は画面のレイアウトに一切の変化を加えず、かつその操作はユーザがマウス操作において慣れ親しんだ間接操作と似たものとなっている。TouchShield[11]は、広い面積にてタッチを行った際に、その位置に物理キーを模したウィジェット(Home, Sleep, Back, Menu)を出現させることにより、指の届かない位置にあるこれらのキーの操作の代わりとする。本手法はこの手法と異なり実際に画面に存在するGUIを操作するため、このウィジェットに存在しないキーの操作も可能となる。

2.2 タッチパネル端末向けの操作切り替え手法

タッチパネル端末に対する特殊な動作を、端末の操作の切り替えに用いる手法が研究されている。特殊なタッチジェスチャを入力への切り替えとして用いる研究に、Fat Thumb[4]がある。この手法はタッチに用いた指の面積を入力モードの切り替えに用いる。また、MicroRolls[21]は親指を転がすようなジェスチャであり、その方向や速度によって様々な入力を行うことが可能となる。

携帯情報端末の握りの強弱を用いて操作の切り替えを行う研究が存在する。GripSense[6]は端末に組み込まれたジャイロセンサ及び端末のバイブレーション機能を用いて端末に対する握りの強弱を判定し、携帯情報端末の操作を切り替える。同様にForce Gestures[9]も、端末に取り付けた圧力センサを用いて端末に対する握りの強弱を判定している。

また、Forcetap[10]は携帯情報端末に組み込まれている加速度センサを用いて、Expressive Touch[18]はタブレットの4隅に取り付けられたマイクを用いてタップの強弱を判定した。TapSense[8]は携帯

情報端末に張り付けたマイクを用いて、画面に触れた指の部位を判定した。これらの手法はそれぞれ、特殊なタッチダウンが発生した際に操作を切り替えるべきか否かを判定している。

本稿における TouchOver は起動/終了にホームボタンのダブルタップを用いるが、[4][21][6][9][10][18][8]それぞれの操作を用いて操作の切り替えを行うことも可能である。具体的には、[4]を適用する場合、広い面積にてタッチを行った際にはタッチを含めて以降に発生するタッチイベントを転送し、そうでないタッチを行った場合には直接操作とする設計が考えられる。[21]の場合、タッチ後に画面上方向に指を転がすようなジェスチャを行った際に、以降のタッチイベントを転送する。[6][9]の場合、端末を強く握っている際に発生したタッチイベントを転送する。[10][18]の場合には強くタッチを行った際に、[8]の場合には指の先端を用いてタッチを行った際に、以降のタッチイベントを転送する。

3 TouchOver

TouchOver の要件は大端末に対する安定した操作、すなわち片手親指のみを用いた持ち替えを必要としない、画面の全領域に対する操作を可能とすることである。そこで本研究においては以下の2点をアプローチとする。

- ユーザは指が届かない領域に対しては、間接操作を用いて画面下半分の領域にて操作を行う(間接操作の設計)。
- 間接操作と直接操作とを任意に切り替える(切り替え動作の設計)。

3.1 間接操作の設計

本手法にて用いる間接操作は、画面下部のみを用いた画面上部の操作である。従来の間接操作手法の多くは画面に対するタッチダウンを用いてカーソルを出現させ、ドラッグ操作を用いてそのカーソルを目的の位置に移動させ、タッチアップにてカーソルが存在する位置をポインティングさせる。しかし、これらの手法はドラッグやロングタップといったタッチジェスチャを行うことができず、また、カーソルを動かすための

ドラッグ操作と端末において動作しているアプリケーションに対するドラッグ操作(例:スクロール)が競合する。

このような競合は操作の煩雑化の原因となりうるため、TouchOver においては、画面下部にて発生したタッチイベントを画面上部に転送する手法を用いる。すなわち画面の縦の長さを h としたとき、座標 (x, y) にて発生したタッチイベントを座標 $(x, y - h/2)$ に転送する。これにより、画面下部のみを用いてジェスチャを含む全ての操作を画面上部に対して行うことができる。なお、 $y < h/2$ の領域、すなわち画面上部にタッチダウンが発生した場合、TouchOver が起動している場合においてもその一連の操作(例:画面上部から画面下部へのドラッグ操作)を直接操作として扱う。

この手法を用いる場合、特に小さいターゲットを操作する際には、タッチダウン時にタッチイベントがどこに転送されるかのフィードバックが必要となる。そこで本手法においては、TouchOver が起動している間にホバーイベントが発生している際には図2aのようにタッチイベントの転送予想点を緑色のカーソルとして表示する。ホバーイベントとは、端末の画面から少し離れた位置に指があるときに発生するイベントである。図3に示すように、ホバーイベント発生時の指を画面に対して垂直におろした場合、ホバーイベントの発生した座標及びタッチイベントの発生する座標が一致する。すなわち、ホバーイベントが座標 (x_{hover}, y_{hover}) に発生している際には、座標 $(x_{hover}, y_{hover} - h/2)$ をタッチイベント転送予想点として利用することが可能となる。またタッチイベントが発生している際には、タッチイベントの転送先に赤色のカーソルを図2bのように表示することにより、タッチイベントが転送されている位置をフィードバックする。

このフィードバックにより、ユーザはホバーしている親指の移動を用いて先述の間接操作手法のように TouchOver を使用することができる。したがって、TouchOver は従来のカーソルを用いた間接操作手法と同程度の容易さで学習が可能である。

また、TouchOver が起動している間は、画面の下

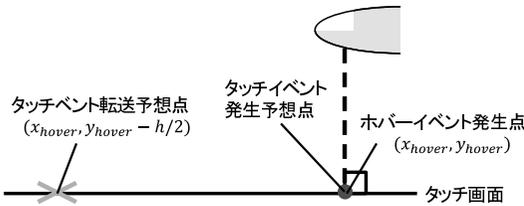


図3 ホバーイベント発生点、タッチイベント発生予想点及びタッチ点転送予想点の座標関係。

半分を赤く表示する。これはユーザに TouchOver の起動していること、及び、画面の上半分及び下半分の境界をフィードバックするためである。

3.2 切り替え動作の設計

本手法は、直接操作と間接操作を切り替える動作として、ホームボタンのダブルタップを用いる。先述のように、タッチパネル端末向けの操作切り替えに関する研究が多数存在する。しかし特殊なタッチジェスチャを TouchOver の起動/終了に用いた場合、アプリケーションに対するタッチジェスチャとの競合が発生する。大端末に対する握りの強弱を切り替えに用いた場合は手の疲労が操作の安定性に影響を及ぼす可能性があり、特殊なタッチダウンにて切り替えを行う場合には細かい操作が困難になることが考えられる。画面に対するタッチを必要とせず、また大端末を把持する手に力を込める必要のない操作として大端末をシェイクする操作が考えられる。これは携帯情報端末の素早い操作切り替えに適していると Wu ら [23] によって述べられており、TouchOver の起動/終了にも適している可能性がある。ただし、シェイク操作を本手法に単純に適用した場合には以下の問題が生じることが、我々が予備実験を行ったところわかった [25]。すなわち、1) 操作の間画面を見ることができないため切り替えのたびに操作対象を探す必要があり、結果として速度が低下する点、2) 大端末は重量があるため連続して切り替えを行うと手が疲労する点である。

そこで本稿においては、同じく大端末に対する片手操作手法である Reachability の起動/終了と同じ操作であるホームボタンのダブルタップを用いることとした。これは iPhone 6 及び 6 plus に実装されている

操作であり、従来のタッチジェスチャと同様にユーザにとって親しみやすい操作である。また、後述の比較実験においては Reachability と操作切り替えに関して同条件での比較を可能とする。本手法はホバーイベントの検出により操作性を向上させているが、現在 iPhone 6 及び 6 plus はホバーイベントを取得できない。したがって、我々は Android 端末上におけるホームボタンのダブルタップの検出を可能とするウィジェットを作成し、ホバーイベントを検出可能な Android 端末に装着した。

4 使用例

本節においては、Galaxy S4 SC-04E(Android 4.2.2, 69.8 × 136.6 × 7.9 mm, 1080 px × 1920 px, 5") にて実装したプロトタイプの使用例を述べる。なお、この端末はホバー検出が可能な端末であり、操作中使用者は利き手のみにて端末を把持した。

4.1 使用例 1: 動画サイトへのアクセス及び動画の再生

TouchOver を用いた動画サイトへのアクセス及び動画の再生手順を図 4 に示す。大端末のホーム画面に表示された動画サイトのアイコンに対して、ユーザの指が直接届かない場合を考える (図 4a)。この時ユーザは、ホームボタンのダブルタップを用いて TouchOver を起動する (図 4b)。画面にはホバーイベントを用いたタッチイベント転送先のフィードバックが緑色のカーソルとして表示されるため、このカーソルを左上のアイコンに合わせることでよりタッチ点の転送先を決定する。この状態にて画面をタップすることによってアイコンをタップする (図 4c)。同様にカーソルを画面上部の再生ボタンに合わせてタップすることにより、再生ボタンをタップする (図 4d)。カーソルを再生画面下部に合わせてドラッグ操作を行うことにより、シークバーを操作する (図 4e)。

この間手は大端末下部を安定に把持したままであり、大端末を安定に操作しているといえる。また、現在の動画を再生し終わり画面下部に表示されている別の動画にアクセスする際は、再びホームボタンをダブルタップすることによって TouchOver を終了さ

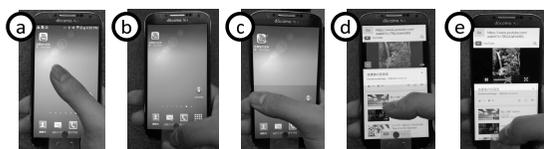


図4 動画サイトへのアクセス及び動画の再生。
 a: 指が直接届かない領域に表示された動画サイトのアイコン。b: ホームボタンのダブルタップによるTouchOverの起動。c: TouchOverを用いたアイコンのタップ。d: TouchOverを用いた再生ボタンのタップ。e: TouchOverを用いたシークバーの操作。

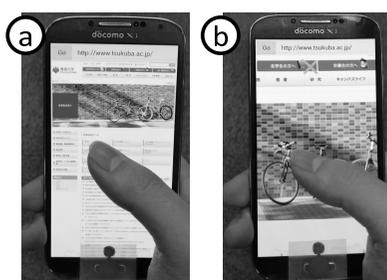


図5 ダブルタップを用いたブラウザの拡大。a: 画面上部に小さく表示されたメニュー。b: TouchOverを用いたダブルタップを行うことによる画面の拡大。

せ、直接リンクをタップすることによって次の動画にアクセスすることが可能である。

4.2 使用例2: ダブルタップを用いたブラウザの拡大

ブラウザを用いて情報を確認している際、1度の表示量を多くするために画面が縮小された状態にて表示される場合がある(図5a)。ブラウザ上においてはダブルタップイベントが発生した座標を中心に画面が拡大されるが、今回目的とするメニューは画面上部にあるため指で直接ダブルタップを行うことができず、また、このメニューはウェブページの最上段に記述されているため、画面のスクロールを用いて手元に移動させることもできない。そこでユーザは、図5bに示すようにカーソルをメニューに合わせてダブルタップを行う。これによりカーソルの位置にダブルタップイベントが転送されるため、画面下部にて行ったダブルタップによって画面上部の拡大が可能となる。

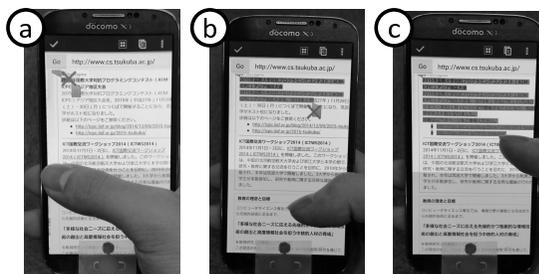


図6 ロングタップを用いた文字列選択。a: TouchOverを用いてロングタップを行うことによる文字列選択モードの起動。b: TouchOverを用いたドラッグ操作による選択ハンドルの移動。c: 指を用いた選択ハンドルの移動。

4.3 使用例3: ロングタップを用いた文字列選択

ブラウザに表示された文字列をコピーするため、表示されている文字列を選択する必要がある。ブラウザ上における文字列選択は、画面に対するロングタップを行い、表示された文字列選択ハンドルの範囲をドラッグ操作によって調節することによって行う。しかし今回対象とする文字列は画面上部に表示されており、直接指にてロングタップを行うのは困難である。そこで図6aに示すように、まずTouchOverを用いて先頭単語に対してロングタップを行い文字列選択モードを起動する。その後表示される文字列選択ハンドルにTouchOverを用いてタッチを行い、続くドラッグ操作にてハンドルを移動させる(図6b)。最後に直接指でハンドルを操作し、目的の文字列を選択する(図6c)。このように、TouchOverは画面の上半分領域を画面下部のみを用いて操作可能とする一方で、たとえ画面上部であってもユーザの指が届く範囲ならば直接指でタッチすることによって操作することも可能とする。

5 実験1: 基本操作の性能評価

基本的なタッチジェスチャであるポインティング及びドラッグの性能評価を行うため、比較実験を行った。比較対象には、既存手法である直接タッチ(Touch)、Reachability及び先行研究であるExtendedThumb[16]を用いた。

5.1 被験者

被験者は全員が普段からスマートフォンを使用している12名であり、TouchOverを使用したことがなかった。被験者は全て男性であり、全員が右利きであったが、普段スマートフォンの片手操作を行う際に左手を用いる被験者が1名いた。

5.2 実験機器

実験端末として、我々は Samsung Galaxy S4 SC-04E を用いた。また、ホームボタンのダブルタップを検出するためにウィジェットを作成し、実験端末に取り付けた。本手法に用いるウィジェットは加藤らの手法[13]にインスパイアされたものであり、導電インクとしての銀ナノ粒子インク[14]^{†1}及び専用紙^{†2}による紙片によって構成される。この導電インクにて描かれた図形はユーザのタッチを認識可能な領域として作用する。我々はまず、図7aに示すように導電インクを用いて専用紙にパターンを印刷する。このパターンは大小2つの円が線で結ばれた形である。その後この紙を30.0mm×18.0mmに裁断することにより紙片とし、図7bに示すように、端末の下端に、大きい円がホームボタンを覆い、かつ印刷面が裏面になるように取り付ける。これはタッチ面を滑らかにすると共に、パターンがユーザのタッチによりかすれることを防ぐためである。この大きい円をダブルタップすることにより小さい円の位置にダブルタップイベントが発生するため、ホームボタンのダブルタップが検出可能になる。

5.3 実験環境

実験環境を図8に示す。実験環境を統制するため、被験者には椅子に座り、普段片手でスマートフォンを操作する手を用いて端末を把持するよう指示した。また、端末を持つ手の肘から先を机や体で支えることができないよう指示した。

被験者には実験にあたり、速度よりも精度を優先するよう教示を行った。これはTouchOverの設計指針である安定した片手操作に従うものである。直接タッ

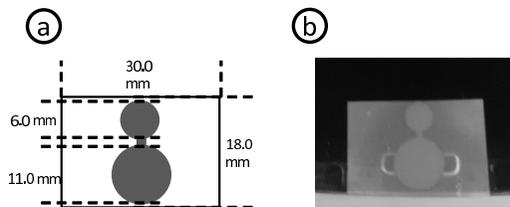


図7 a: ウィジェットに印刷したパターン。
b: ウィジェットを端末に取り付けた様子。

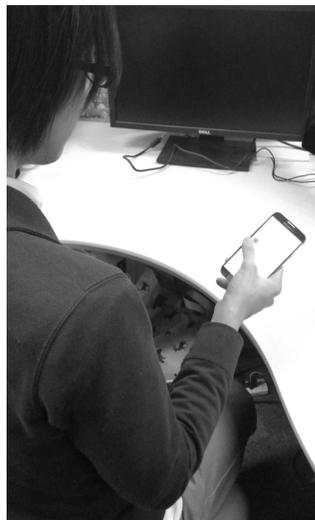


図8 実験環境。

チにて操作を行う際、指がターゲットに届かない場合には端末を把持する手のみを用いて端末を持ち替えるよう被験者に指示した。

5.4 タスク

被験者はそれぞれの操作手法について、ポインティングタスク及びドラッグタスクを行った。ポインティングタスクにおける1試行は、図9aに示すターゲットをタップすることである。ターゲットの大きさはホーム画面におけるアイコンの大きさにならった。また、ドラッグタスクにおける1試行は、図9bに示すシークバーを目標の値に合わせることである。目標値はランダムに決定され、スクロール操作の実験を行った先行研究[5]と同様に赤い線で目標値となる範囲を示した。

各セッションは被験者が画面をタッチすることによって開始された。被験者が試行を正しく完了するご

^{†1} NBSIJ-MU01

^{†2} NB-TP-3GU100

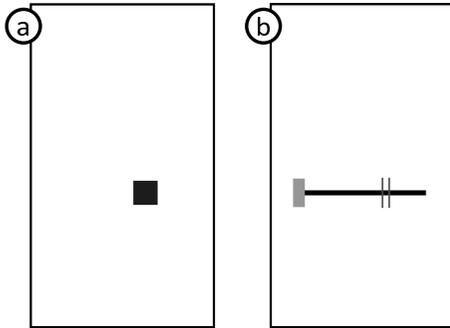


図9 a: ポインティングタスクにおけるターゲット。
b: ドラッグタスクにおけるシークバー。

とに、ランダムに選ばれた新たなターゲットが表示され、ポインティングは24試行、ドラッグは6試行を行うことで終了する(1セッション)。各セッションの間には休憩を取ることを許可した。また、セッション中それぞれのターゲットは全て異なる領域に表示された。具体的に、ポインティングタスクでは画面を4×6に分割した24領域にターゲットが、ドラッグタスクでは画面を1×6に分割した6領域にシークバー(上からA~F)が表示された。

これらのタスクはポインティング、ドラッグの順に行われ、操作手法の使用順は、手法間の影響をなくすため被験者ごとに重複のないランダムとした。この順番はポインティングタスク、ドラッグタスク、後述の実験2において全て同じであった。本実験は学習に1セッション、習熟に2セッション、本番に4セッションの計7セッションから構成されたため、被験者は実験1において合計で840回の試行(4×(24+6)×7)を行った。

被験者にはタスク終了後にアンケートに回答してもらった。実験1、実験2、アンケートの回答を合わせて被験者1人あたりの実験時間は合計約2時間であり、拘束時間に対する謝礼として1人あたり1640円が支払われた。

5.5 実験結果

5.5.1 エラー率

ポインティングタスクにおける結果を図10に、ドラッグタスクにおけるエラー率を図11にそれぞれ示す。

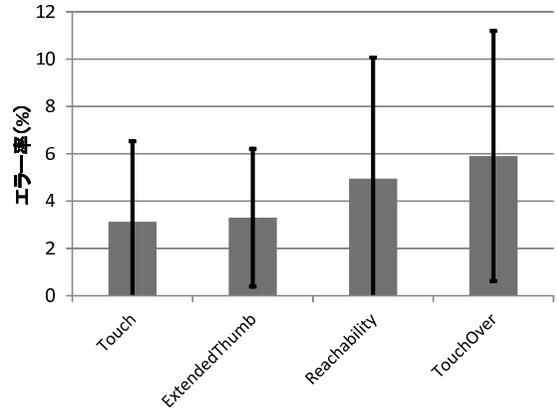


図10 ポインティングタスクにおけるエラー率。

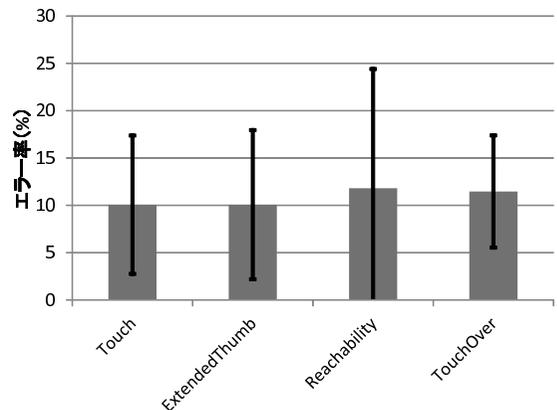


図11 ドラッグタスクにおけるエラー率。

ポインティングタスクにおけるエラー率は3.13%~5.90%であった。これらの結果において、手法間に主効果は見られなかった($F_{3,33} = 2.22, p = .104$)。また、ドラッグタスクにおけるエラー率は10.07%~11.80%であった。これらの結果においても、手法間に主効果は見られなかった($F_{3,33} = 0.14, p = .935$)。

5.5.2 操作時間

ポインティングタスクにおける結果を図12に、ドラッグタスクにおける結果を図13にそれぞれ示す。ポインティングタスクにおける操作時間はTouchが878.23 ms, ExtendedThumbが1196.38 ms, Reachabilityが1114.68 ms, TouchOverが1373.26 msであった。これらの結果において、手法間に主効果が見られた($F_{3,33} = 42.87, p = .000$)。多重比較を行っ

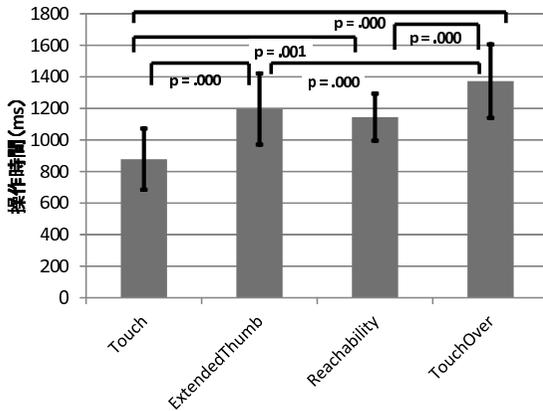


図 12 ポインティングタスクにおける操作時間。

たところ、直接タッチが他に比べて有意に速く ($p = .000$, $p = .001$, $p = .000$), TouchOver は他に比べて有意に遅かった ($p = .000$, $p = .000$). また、ドラッグタスクにおける操作時間は Touch が 2109.62 ms, ExtendedThumb が 2577.03 ms, Reachability が 2490.02 ms, TouchOver が 2675.69 ms であった。これらの結果においても手法間に主効果が見られた ($F_{3,33} = 10.34$, $p = .000$). 多重検定を行ったところ、直接タッチが他に比べて有意に速かった ($p = .000$, $p = .001$, $p = .000$).

5.6 考察

実験の結果より、ポインティングのような単純な操作においては、たとえ端末が大きくとも直接タッチによって操作を行う方がよい性能が期待できることが判明した。そこでより複雑な操作を行ったドラッグタスクの結果について、TouchOver の目的である安定性を測るために、各シークバーごとのエラー率を調べた。結果を表 1 に示す。

各数値は直接タッチの値との差によって色分けされており、赤色は直接タッチよりエラー率の低いシークバー、黄色は直接タッチと同じエラー率のシークバーを示す。これらの結果において、手法及び領域の交互作用における単純主効果がシークバー A に見られた ($F_{3,198} = 3.59$, $p = .014$). 多重比較を行ったところ、シークバー A におけるエラー率は直接タッチの値が他の手法の値に比べて有意に高かつ

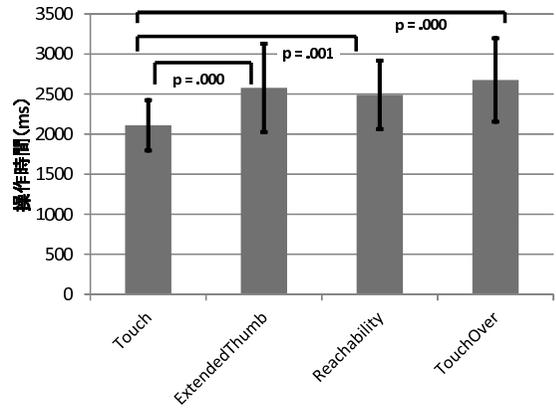


図 13 ドラッグタスクにおける操作時間。

表 1 シークバーごとのエラー率

	Touch	ExtendedThumb	Reachability	TouchOver
A	27.08	8.33	10.42	8.33
B	14.58	6.25	14.58	12.50
C	6.25	22.92	10.42	18.75
D	6.25	8.33	10.42	6.25
E	2.08	6.25	10.42	8.33
F	4.17	8.33	14.58	14.58

た ($p = .006$, $p = .006$, $p = .015$). この結果は、TouchOver は他の手法同様、親指から最も遠い領域において複雑な操作を行う際に安定して操作が可能であることを示している。一方でシークバー F においては、Reachability 及び TouchOver のエラー率が直接タッチの 3 倍の高さを示した。これはホームボタンのダブルタップを実現するために我々が装着したウィジェットが操作を阻害したためであると考えられる。

また、シークバー C は直接タッチ以外のエラー率が高いという結果が示された。これは表示位置が画面上半分の領域ではあるが、被験者の指が比較的届く位置であるため混乱が生じた結果であると考えられる。すなわち、シークバー C については、直接タッチ、TouchOver のどちらを用いても操作が可能であるため、ユーザはどちらの手法を用いるべきか混乱し、これがエラー率が高くなった原因であると考えられる。これはそれぞれの手法をより長く使うことによって、被験者が直接タッチとそれぞれの手法を使い分ける領域の境界を学習することで改善される可能性があるため、今後調査する必要がある。

6 実験2：連続操作における性能評価

実験1において、画面上部に対して複雑な操作を行う際の TouchOver の安定性が示された。そこで、この安定性が実環境における操作時間に与える影響を調査するため、実験2を行った。実験2においては、連続操作における TouchOver の性能を他手法と比較する。今回は実環境を想定した連続操作として、複数のシークバーを目標値に合わせた後にターゲットをポインティングするというタスクを行った。これは、Android 端末における音量の設定画面を模したものである。比較対象となる手法は実験1と同一であり、また、被験者、実験機器及び実験環境も実験1と同条件にて行った。

6.1 タスク

本タスクは図14aに示す、6本のシークバーの位置調整及び決定ボタンを模したターゲットのポインティングからなる。このシークバーは実験1に用いたものと同じものであり、ターゲットはポインティングするのに十分な大きさで画面右下に表示された。

はじめに、被験者は任意の順番にてシークバーの位置合わせを行った。全てのシークバーを合わせ終えた後、被験者はターゲットをポインティングした。この時位置のずれているシークバーがあった際には、図14bに示すように、そのシークバーの左に赤くフィードバックを表示した。この一連の操作を1つのセッションとし、各セッションは図14cに示すターゲットをポインティングすることで開始された。このターゲットはセッション中表示されているものと同一のものであった。本実験は学習に2セッション、習熟に2セッション、本番に4セッションの計8セッションから構成された。

6.2 実験結果

本実験結果を図15に示す。1セッションの操作時間は Touch が 12454.58 ms, ExtendedThumb が 16618.71 ms, Reachability が 13865.04 ms, TouchOver が 16522.83 ms となった。これらの結果において、手法間に主効果が見られた ($F_{3,33} = 12.29, p = .000$)。

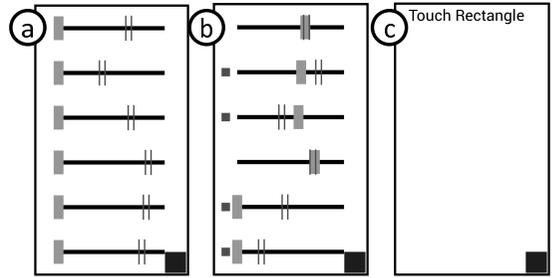


図14 a：シークバー及びターゲット。b：ハイライトされたシークバー。c：開始画面。

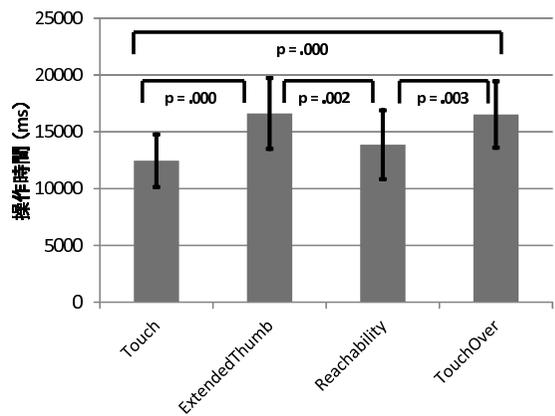


図15 連続操作における所要時間。

多重比較を行ったところ、Touch 及び Reachability の操作時間が ExtendedThumb 及び TouchOver の操作時間に対して有意に短かった。

6.3 考察

実環境を想定した場合、TouchOver や ExtendedThumb よりも直接タッチや Reachability の方が操作時間が短いという結果が得られた。この結果はカーソルを目標まで動かす時間が原因である。すなわち、カーソルの移動に対して端末の持ち替えの時間が短いことを示している。今回実験にて使用した端末は、手の大きさや持ち方によっては画面のほとんどの領域に対して指が届くため、より大きな端末を用いれば TouchOver の操作時間の方が短くなる可能性がある。

また Reachability については、切り替え動作がホームボタンのダブルタップのみで済み、カーソルの移動

などの操作が必要ないため、操作時間が短くなったものと考えられる。しかし実験後のアンケートにおいて、画面下部のコンテンツが隠れてしまうことを問題視するコメントが1件あった。したがって、画面の全領域を見ながら画面上部の操作を行わなければならない実環境(カメラアプリケーションなど)について、更に調査を行う必要がある。

ExtendedThumb については実験1とは異なり TouchOver との差が小さくなった。これは、ExtendedThumb が操作を行うたびにダブルタップを用いる必要があるため、本実験のように操作対象が画面上部に集中している場合には TouchOver の操作時間が短くなるためであると考えられる。被験者からも、ExtendedThumb のダブルタップの煩雑さを問題視するコメントが1件あった。今後はこの仮説を検証するために、この条件をより顕著にした実験を行い詳細な調査を行う必要がある。

7 今後の課題

プロトタイプの使用及び実験によって得られた知見より、今後の課題を述べる。

7.1 画面左側に対する操作

プロトタイプの使用及び実験中に、画面左端の操作を行う際わずかではあるが持ち替えの必要が生じた。この問題に対しては、タッチ点転送の方向を上方向のみから、たとえば上、左、左上の3種に拡張することによって改善が可能であると考えられる。これは Ninja Cursors [15] と同様にアイコンとカーソルの平均距離を縮める効果があるため、より小さな領域のみを用いて大端末の操作が可能となる。

7.2 カーソルを用いることによる利点

実験後のアンケートにおいて、画面下部に対する操作が指で隠れてしまうために困難であったと回答した被験者が1名いた。これに対して TouchOver はカーソルを用いた間接操作手法であるため、画面上部に対しては Shift [22] や Tyler らの研究 [7] などと同様に指によって操作対象が隠れる問題を解決すると考えられる。

7.3 操作切り替え手法の検討

現在、TouchOver の切り替えにはホームボタンのダブルタップを用いている。しかし、ホームボタンのダブルタップを検出するために取り付けられたウィジェットが操作を阻害するという問題が明らかになった。この問題に対し我々は、大端末をシェイクする動作を用いて TouchOver の起動/終了を行うことを考えている。シェイク操作は大端末に組み込まれたセンサのみを用いて実装可能であり、画面に対する操作を阻害することがないため、より高い精度にて操作が可能となると考えられる。ただし3.2節に述べたように、シェイク操作には速度を低下させる可能性及び疲労が問題となる可能性もある。従って、今後はシェイク操作に関して速度、疲労、精度の関係を調査する必要がある。

7.4 横方向把持時における TouchOver

今回は大端末を縦方向に把持した状態にてプロトタイプの使用及び実験を行ったが、大端末を横方向に把持した状態においても同様に持ち替え問題は発生する。例として YouTube などの動画サイトを閲覧する際、動画のアスペクト比によっては大端末を横方向に把持して閲覧する場合がある。この場合シークバーの操作は画面長辺の方向となり、指が届かない領域に対して操作を行う必要も存在する。このとき TouchOver は、例えば右手で大端末を把持する場合には画面の右半分領域のみを用いて全領域の操作を行うことが可能となる。このような場合における TouchOver の性能評価も今後行う必要がある。

8 まとめ

大端末向けの片手操作手法である TouchOver を示した。これは大端末を片手のみにて把持した際の操作を安定して行う、すなわち片手親指のみを用いた持ち替えを必要としない、画面の全領域に対する操作を行うための手法であり、画面下部にて発生したタッチイベントを画面上部に対して転送することによって実現される。このときタッチ点の転送先のフィードバックとしてホバーイベントを用いたカーソルの表示を行い、TouchOver を起動/終了するための動作

としてホームボタンのダブルタップを用いた。また、TouchOver を Galaxy S4 SC-04E 上にて動作するプロトタイプアプリケーションとして実装し、その使用例を述べ、被験者実験によってその性能を評価した。

実験の結果、単純な操作においては直接タッチが有意に速いが、画面上部に対するエラー率が高いという結果が得られた。また、ExtendedThumb は画面上部に操作対象が画面上部に集中している場合に TouchOver と比べ相対的に操作時間が遅くなり、Reachability は画面の全領域を見なければならぬ操作を行う際の問題がコメントによって得られた。これらから TouchOver の有用性が示唆された。

今後は転送先の拡張や新たな切り替え動作を検討するとともに、横方向把持時の性能の評価を行う。

参考文献

- [1] How to use reachability on the iphone 6 and 6 plus - cnet, <http://www.cnet.com/how-to/how-to-use-reachability-on-iphone-6-6-plus/>.
- [2] Baudisch, P., Cutrell, E., Robbins, D., Czerwinski, M., Tandler, P., Bederson, B. and Zierlinger, A.: Drag-and-Pop and Drag-and-Pick: Techniques for Accessing Remote Screen Content on Touch- and Pen-operated systems, in *Proc of Interact Conference*, 2003, pp. 57–64.
- [3] Bezerianos, A. and Balakrishnan, R.: The Vacuum: Facilitating the Manipulation of Distant Objects, in *Proc of CHI '05*, ACM, 2005, pp. 361–370.
- [4] Boring, S., Ledo, D., Chen, X. A., Marquardt, N., Tang, A. and Greenberg, S.: The Fat Thumb: Using the Thumb's Contact Size for Single-handed Mobile Interaction, in *Proc. of MobileHCI '12*, ACM, 2012, pp. 207–208.
- [5] Cockburn, A., Savage, J. and Wallace, A.: Tuning and Testing Scrolling Interfaces That Automatically Zoom, in *Proc. of CHI '05*, ACM, 2005, pp. 71–80.
- [6] Goel, M., Wobbrock, J. and Patel, S.: GripSense : Using Built-in Sensors to Detect Hand Posture and Pressure on Commodity Mobile Phones, in *Proc. of UIST '12*, ACM, 2012, pp. 545–554.
- [7] Gunn, T. J., Zhang, H., Mak, E. and Irani, P.: An evaluation of one-handed techniques for multiple-target selection, in *Proc. of CHI EA '09*, New York, NY, USA, ACM, 2009, pp. 4189–4194.
- [8] Harrison, C., Schwarz, J. and Hudson, S. E.: TapSense: Enhancing Finger Interaction on Touch Surfaces, in *Proc. of UIST '11*, ACM, 2011, pp. 627–636.
- [9] Heo, S. and Lee, G.: Force Gestures: Augmented Touch Screen Gestures Using Normal and Tangential Force, in *Proc. of CHI EA '11*, ACM, 2011, pp. 1909–1914.
- [10] Heo, S. and Lee, G.: Forcetap: Extending the Input Vocabulary of Mobile Touch Screens by Adding Tap Gestures, in *Proc. of MobileHCI '11*, ACM, 2011, pp. 113–122.
- [11] Hong, J. and Lee, G.: TouchShield: A Virtual Control for Stable Grip of a Smartphone Using the Thumb, in *Proc. of CHI EA '13*, ACM, 2013, pp. 1305–1310.
- [12] Karlson, A. K. and Bederson, B. B.: One-handed Touchscreen Input for Legacy Applications, in *Proc. of CHI '08*, ACM, 2008, pp. 1399–1408.
- [13] Kato, K. and Miyashita, H.: Extension Sticker: A Method for Transferring External Touch Input Using a Striped Pattern Sticker, in *Proc. of UIST'14 Adjunct*, New York, NY, USA, ACM, 2014, pp. 59–60.
- [14] Kawahara, Y., Hodges, S., Cook, B. S., Zhang, C. and Abowd, G. D.: Instant Inkjet Circuits: Lab-based Inkjet Printing to Support Rapid Prototyping of UbiComp Devices, in *Proc. of UbiComp '13*, New York, NY, USA, ACM, 2013, pp. 363–372.
- [15] Kobayashi, M. and Igarashi, T.: Ninja Cursors: Using Multiple Cursors to Assist Target Acquisition on Large Screens, in *Proc. of CHI '08*, New York, NY, USA, ACM, 2008, pp. 949–958.
- [16] Lai, J. and Zhang, D.: ExtendedThumb: A Motion-based Virtual Thumb for Improving One-handed Target Acquisition on Touch-screen Mobile Devices, in *Proc of CHI EA '14*, ACM, 2014, pp. 1825–1830.
- [17] Parhi, P., Karlson, A. K. and Bederson, B. B.: Target Size Study for One-handed Thumb Use on Small Touchscreen Devices, in *Proc. of MobileHCI '06*, ACM, 2006, pp. 203–210.
- [18] Pedersen, E. W. and Hornbæk, K.: Expressive Touch: Studying Tapping Force on Tabletops, in *Proc. of CHI '14*, ACM, 2014, pp. 421–430.
- [19] Roth, V. and Turner, T.: Bezel Swipe: Conflict-free Scrolling and Multiple Selection on Mobile Touch Screen Devices, in *Proc. of CHI '09*, ACM, 2009, pp. 1523–1526.
- [20] Roudaut, A., Huot, S. and Lecolinet, E.: Tap-Tap and MagStick: Improving One-handed Target Acquisition on Small Touch-screens, in *Proc. of AVI '08*, ACM, 2008, pp. 146–153.
- [21] Roudaut, A., Lecolinet, E. and Guiard, Y.: MicroRolls: Expanding Touch-screen Input Vocabulary by Distinguishing Rolls vs. Slides of the Thumb, in *Proc. of CHI '09*, ACM, 2009, pp. 927–936.
- [22] Vogel, D. and Baudisch, P.: Shift: A Technique for Operating Pen-based Interfaces Using Touch, in *Proc. of CHI '07*, ACM, 2007, pp. 657–666.
- [23] Wu, Y., Mei, T., Yu, N. and Li, S.: Accelerometer-based Single-handed Video Browsing on Mobile Devices: Design and User Studies, in *Proc. of ICIMCS '12*, ACM, 2012, pp. 157–160.

- [24] Yu, N.-H., Huang, D.-Y., Hsu, J.-J. and Hung, Y.-P.: Rapid Selection of Hard-to-access Targets by Thumb on Mobile Touch-screens, in *Proc. of MobileHCI '13*, ACM, 2013, pp. 400–403.
- [25] 大西主紗, 志築文太郎, 田中二郎: TouchOver: 大画面を備える携帯情報端末を楽に操作するための片手親指操作手法, 第 22 回インタラクティブシステムとソフトウェアに関するワークショップ (WISS2014), 日本ソフトウェア学会, 2014, pp. 85–90.
- [26] 土佐伸一郎, 田中二郎: LoopTouch: 画面ループを用いたモバイル端末片手操作手法, インタクション 2013, 情報処理学会, 2014, pp. 175–182.



大西主紗

2014 年筑波大学情報学情報科学類卒。同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博士前期課程在学中。モバイルインタクションに関する研究に興味を持つ。情報処理学会会員。



志築文太郎

1994 年東京工業大学理学部情報学科卒業。2000 年同大学大学院情報理工学研究科数理・計算科学専攻博士課程単位取得退学。博士(理学)。現在、

筑波大学大学院システム情報系准教授。ヒューマンインタフェースに関する研究に興味を持つ。日本ソフトウェア科学会, ACM, IEEE Computer Society, 電子情報通信学会, ヒューマンインタフェース学会各会員。



田中二郎

1975 年東京大学理学部卒業。1977 年同大学大学院理学系研究科修士課程修了。1984 年米国ユタ大学大学院計算機科学科博士課程修了。ユタ大学では関数型プログラミング言語の並列実装に関する研究に従事。PhD. in Computer Science。1985 年から 1988 年に(財)新世代コンピュータ技術開発機構にて並列理論型プログラミング言語の研究開発に従事。1993 年から筑波大学に勤務。現在、筑波大学システム情報系教授。ユビキタスコンピューティングや未来の情報環境の構築に興味を持つ。ACM, IEEE, 情報処理学会, ヒューマンインタフェース学会, 電子情報通信学会各会員。