

# 指の近接位置情報を利用した ユーザ操作支援機能の開発

スマートフォンはタッチパネルを用いることで直感的に操作できるという特長があるが、タッチ前に操作対象をフォーカスする手段がないため、選択前に処理内容を確認することが難しいという課題がある。これを解決するために、指の近接位置情報を検出し、その情報を利用したアシスト機能を開発し、2013年夏モデルに搭載した。本稿では指の近接位置情報の検出手法、および2013年夏モデルに搭載したユーザ操作支援機能について、ソフトウェアを中心に解説する。

移動機開発部 多賀谷 昌志 岡田 たかし  
先進技術研究所 青柳 禎矩

## 1. まえがき

近年市場に投入される移動機端末はスマートフォンが主流となっており、操作方法が従来のフィーチャーフォンと大きく異なっている。フィーチャーフォンは表示部のディスプレイと操作部のハードキーが個別に搭載されている。これに対し、スマートフォンは画面上に操作部と表示部が一体化した構造であり、ユーザは表示されたアイコンなどを直接指で触れることで操作を行う。これによりユーザはディスプレイ上に表示されたアプリケーションや情報を直感的に操作できるという特長がある。その反面、操作対象をフォーカス<sup>\*1</sup>する手段がないため、アイコンの選択前に処理内容を確認するのが難しいという課題がある(図1)。

フォーカスを行うにはフィーチャ

フォンで使われているハードキー(カーソルキー)以外に、マウスやスタイラス(タッチペンなど)を利用する方法が一般的である。しかしながら、スマートフォンへのデバイスの追加が必要となるだけでなく、表示されたアイコンなどを直接触れる直感的操作という利点の喪失も招く。このため、「指で触る」という

基本動作に基づく指の位置情報を用いてフォーカスすることが有効な解決手段になるものと考えられる。

このような背景から、指の近接位置情報を利用したユーザ操作支援機能を開発し、2013年夏モデルに搭載した。本機能は、指がタッチパネルに接触していない近接状態にあっても指の座標が取得できるタッチパ

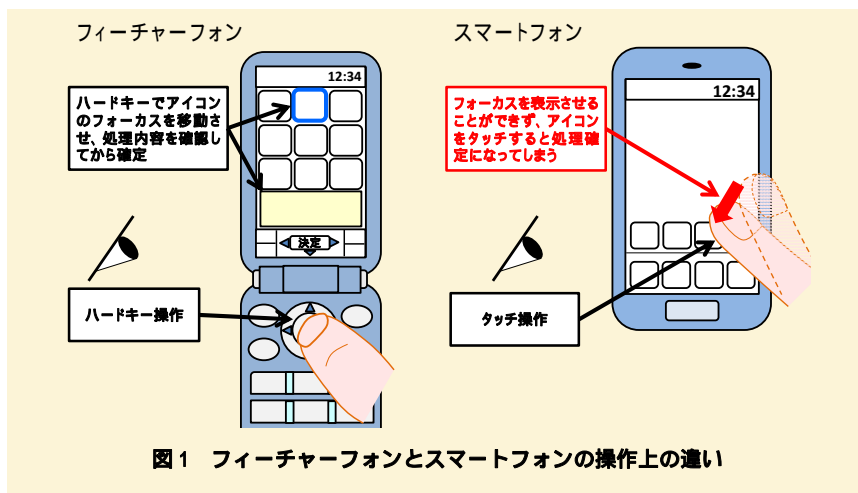


図1 フィーチャーフォンとスマートフォンの操作上の違い

©2013 NTT DOCOMO, INC.  
本誌掲載記事の無断転載を禁じます。

\*1 フォーカス：ボタンなどが選択され入力や決定がされる状態。

ネルを活用し、座標に応じてフォーカスを移動させ、対応した付加情報を表示するなどの処理をする。これにより、直感的操作の利点を損なうことなくユーザの操作性・利便性を向上させる機能である。

本稿では、今回開発した指の近接位置情報を利用したユーザ操作支援機能について説明する。

## 2. 指の近接位置情報の取得

指の近接位置情報を取得する方法には、主に表 1 のような 3 つの方法がある。本開発においては、タッチパネル全面での近接位置情報が取得できること、かつ周囲の光などの影響を受けにくく常に安定した操作が期待できること、そしてなによりも、新規でデバイスの追加が不要であること、これらの理由から、静電容量方式タッチパネルを採用した。

このタッチパネルでは、通常のタッチパネルよりも、小さい静電容量の変化量を検出することができ、1cm 程度タッチパネルから指が離れた状態でも指の座標検出が可能である。これにより、アプリケーションはあらかじめ、ユーザがタッチしようとしている箇所を取得しフォーカスすることが可能になる(図 2)。なお、Android™\*2.4.0 以降では、この近接位置情報を取得するための API (Application Programming Interface)\*3 が標準で実装されており、アプリケーションはこれを利用して開発することが可能である。具体的には標準 API の `getAction()`

メソッドでは、近接情報取得はタッチ検出と同様の仕組みで検出可能であり、タッチ検出であれば `ACTION_MOVE`、近接検知であれば `HOVER_MOVE` のイベントで区別して取得することが可能である。したがって、プリインストールのアプリケーションでなくとも、標準 API を使えば、どのアプリケーションでも指の近接による操作が可能となる。

一方、タッチパネルの検出感度を高めたことにより、ディスプレイなど移動機端末内部からの電磁ノイズなどを受けやすくなり、座標ジッタ

\*4 が増加する。この座標ジッタが大きくなると、ユーザが意図しない対象をフォーカスしてしまい、逆に操作性低下の原因を招いてしまう。

この課題に対して、タッチパネル出力に近接位置情報検出フィルタを挿入することで対策を図ることが考えられた。スマートフォンメーカー各社は、タッチパネルの座標ジッタの特性を調査した結果から、ジッタ抑圧と追従性を両立させるためにフィルタ定数の最適化を行い、かつ、意図した位置と検出位置とのずれを吸収するための位置補正を行った。

表 1 近接座標取得方法の比較

	超音波センサ方式	光センサ方式 インセルタッチパネル	静電容量方式 タッチパネル
座標取得方法	超音波を指にあて、その反射波から座標を計測する	ディスプレイのセルに配置した光学センサの受光強度から座標を計測	タッチパネル上の静電容量値の変化量から座標を計測する
検出範囲	超音波の指向性有 (×)	タッチパネル全面 ( )	タッチパネル全面 ( )
検出距離	5cm 以上可能 ( )	周囲の光に影響を受ける (×)	1cm 程度 ( )
追加デバイス	超音波送信機と複数のレーザーが必要 (×)	従来からデバイスは増えない ( )	従来からデバイスは増えない ( )

×：本開発への適性を示す

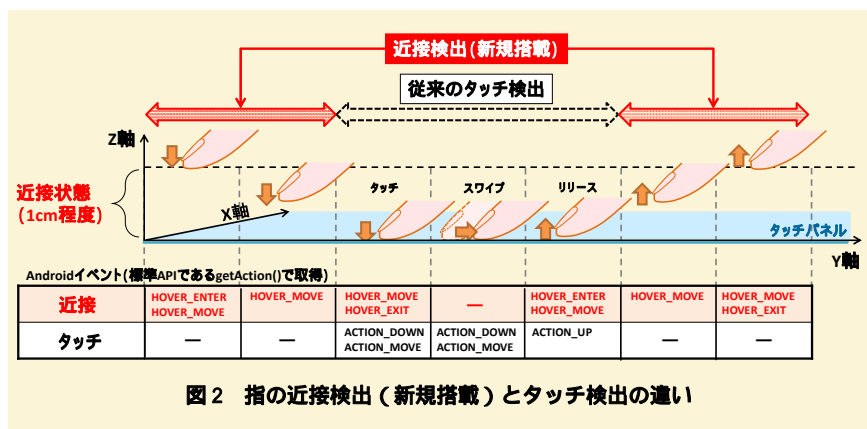


図 2 指の近接検出 (新規搭載) とタッチ検出の違い

\*2 Android™：米国 Google, Inc.の商標または登録商標。

\*3 API：OS やミドルウェアなどが提供する機能を、上位のソフトウェアが利用するためのインタフェース。

\*4 座標ジッタ：タッチパネル上に指を近接させた際に、接近した指が停止した状態であっても、検知される座標が小刻みに変化する現象。

このフィルタの評価に際しては、写真1に示すようなタッチパネルの評価ロボットを用いた。

またこれを用いて、直線移動、曲線移動といった同一動作を繰返し、かつユーザが対象を選択する際の複雑な動作を実行させた。そして、検出距離、検出位置、フォーカスの追従性などのタッチパネルの性能がユーザ体感的に最適となるように各種パラメータの調整を行った。またさ

らに、ユーザの一般的な利用シーンを考慮してタッチパネル上に水滴がある環境、液晶フィルムを装着している環境、皮脂などが付着している環境下での評価も実施し、ユーザの操作感に影響を与えないことも確認した。

### 3. アプリケーションの開発

2013 年夏モデルでは、指の近接

位置情報を利用したアシスト機能を、図3に示すように、Pallet UI（待受け/ホーム画面）、ソフトキーボード、パターン入力によるロック解除機能、ブラウザなど複数のアプリケーションに対応させた。

ここでは、これらの中で、メール作成などの入力作業においてユーザの利用頻度が高く、かつユーザ操作感に一番大きく影響を与えるソフトキーボードについて指の近接位置情報を用いた操作感向上手法を説明する。

#### 3.1 指の近接位置情報を用いたアシスト機能

タッチパネルで検知された指の近接位置情報を用いてユーザの入力をアシストする機能としては、表2に示すように3つの方法が考えられる。

1 つめはガイド表示であり、その部分の情報/機能を見やすい位置に

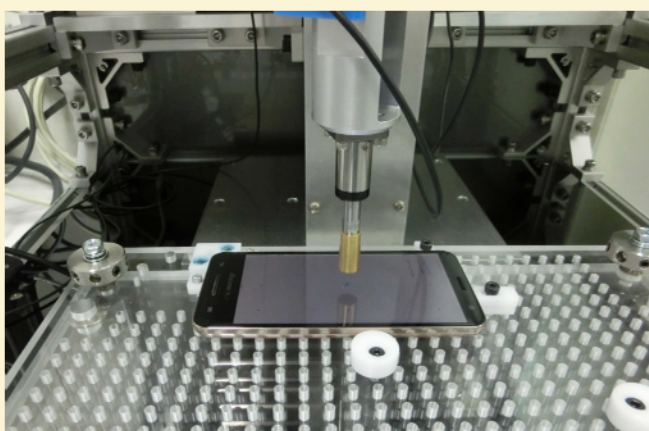


写真1 タッチパネルの座標ジッタ特性の計測の様子



図3 指の近接位置情報を利用したアシスト機能

表示し、指に隠れてしまう場合に効果を発揮する。2 つめは、タッチ検出範囲の拡大であり、タッチされる場所を推定し、検出される範囲を拡大して指のブレを吸収する方法である。表示が小さいものを操作する際に効果を発揮できる。3 つめの拡大表示は、タッチされる場所を推定してその周囲を拡大して見やすく表示することである。ブラウザなどの細かい表示を一時的に拡大して見やすくする際に効果を発揮する。

スマートフォンでは一般的にディスプレイに表示された QWERTY キー\*5 やテンキー\*6 をタッチして文字入力を行う。今回の開発において、キーボードの実装にはこれら 3 つの基本的なアシスト機能に基づき実装方法を検討した。

### 3.2 QWERTY キー入力

QWERTY キーはキーの数が多く 1 つひとつのキーの面積が小さい。また、押そうとしているキーが自分の指で隠れて見えなくなってしまう。このため、押し間違いによる誤入力の可能性が高くなる。

この問題点を解決する方法として、後者のキーが隠れてしまう問題に対しては、ガイド表示が有効である。一方前者の面積が小さい問題に対しては、検出範囲拡大と拡大表示の 2 つの手法が適用できる。このうち、拡大表示に関しては、キーボードのように連続操作が行われる場合、複数キーを拡大させることで指の下以外のキーの中心が、拡大する場合としない場合とずれてくるため、キー

ボード自体の視認性が大幅に損なわれてしまう可能性があった。このような理由から、ここでは、検出範囲を拡大する方法を用いた（表示上はキーを 1 つ拡大表示）。

以上のような検討結果に基づき、QWERTY キー入力では、ガイド表示と検出範囲拡大を組み合わせてアシスト機能を実装した。具体的には、図 4 に示すように、指が近接状態であることを検知したら、座標値から指の下のキーを推定し、指の下のキ

ーの検出範囲（および表示）拡大と、指の下にあるキーが何であるか指の上にガイド表示をする。

### 3.3 テンキー入力

テンキー入力では、QWERTY キー入力と異なり、キー自体の面積が小さいという問題はないが、フリック\*7 入力\*7 が一般的に利用されており、タッチ後に 2 回目の操作が求められる。そのため、方向と割り当てられている文字（上下左右で 4

表 2 近接座標を用いた入力アシストの比較

	ガイド表示	タッチ検出範囲の拡大	拡大表示
アシスト内容	タッチされる部分を推定し、情報/機能を見やすい位置に表示する。	タッチされる部分を推定し、検出される範囲を拡大してユーザのブレを吸収する。	タッチされる場所を推定してその周囲を拡大して見やすく表示する。
対象範囲	タッチされる部分	タッチされる部分	タッチされる部分の周囲を含む一定範囲領域内
有効なケース	対象が指などで隠れてしまう場合	表示が小さいものを操作する場合	細かい表示を一時的に拡大して見やすくする場合
イメージ			

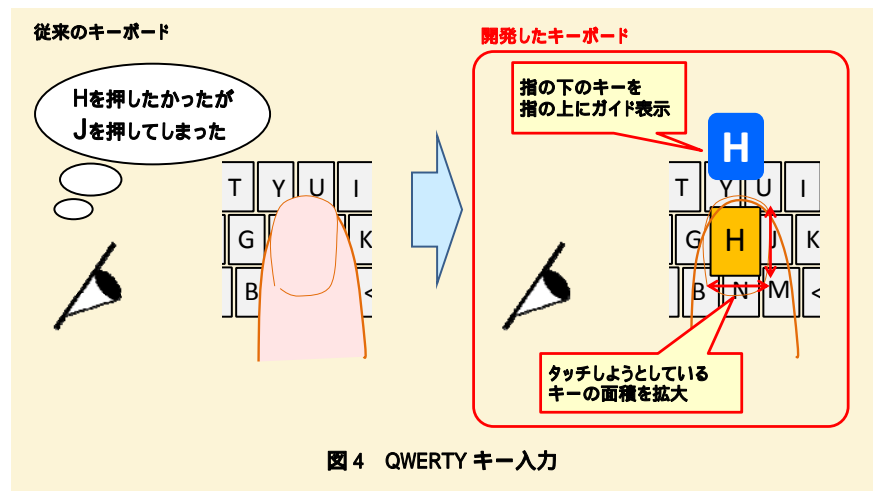


図 4 QWERTY キー入力

\*5 QWERTY キー：アルファベットの入力用として広く利用されているキー配列。

\*6 テンキー：数字入力に特化したキー配列。

\*7 フリック：指を弾くように移動させて入力する方法。

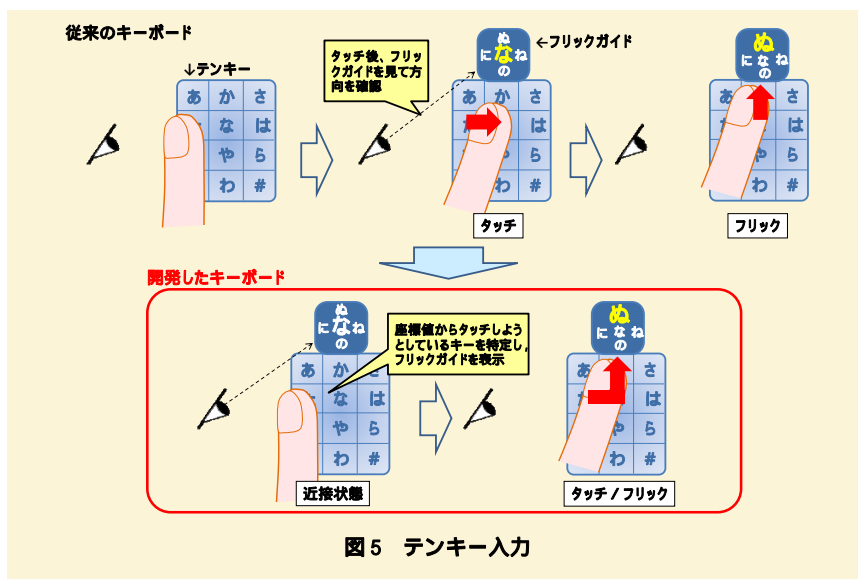
文字割り当てられている)の関係を知らない状態では、タッチし表示されたフリックガイドで方向を確認してからフリックするため、入力に時間がかかってしまう点が課題となる。

この課題に関しては、図5に示すように、指の下のキーのフリックガイドを指が近接した段階で表示する機能を実装した。この機能を利用することにより、ユーザはフリックの方向と割り当てられているキーを知らなくとも、タッチからフリックを一連の動作で行うことができ、入力の時間を短縮することが可能になるものと考えられる。

#### 4. あとがき

本稿では、2013 夏モデルで搭載した指の近接位置情報を用いたアシスト機能に関して、指の近接位置情報を取得するタッチパネルの実現方法、およびアシスト機能を実装したアプリケーションについて、ソフトウェアの実装手法を中心に解説した。

これらの機能実装により、スマー



トフォンに慣れていないユーザは入力時間を短縮し、熟練したユーザも誤入力を減らすことが可能になったものと考えている。しかしながら、2013 夏モデルにおいて初めて実装した機能であるため、実使用感を踏まえて今後改良を加えていく必要がある。なお本稿では Android API での開発を行ったが、HTML5<sup>\*8</sup>で記述されたアプリケーションでも、指の近接位置情報を取得できる HTML タグや JavaScript<sup>\*9</sup> のメソ

ッドを利用することで、同様に開発が可能である。

今後は、タッチパネルの近接位置情報取得に関する精度向上の技術開発動向を注視しつつ、さらなる精度向上を図る。また、アプリケーションは、ゲームのようなエンターテイメントにも適用分野を広げていき、将来的には 3 次元の直感的な操作ができる UI が一般化していくよう利用方法の創出を進めていく。

\*8 HTML5: WHATWG および W3C が策定を進めている HTML の改訂版。

\*9 JavaScript: Web ブラウザなどでの利用に適したスクリプト言語。  
JavaScript は、Oracle Corp. およびその子会社、関連会社の米国およびその他の国

における登録商標または商標。