

把持特徴を利用したユーザインタフェースの開発 Grip UI

近年、タッチパネルによるユーザインタフェースの普及により、画面と端末の大型化が進み操作性低下が指摘されるケースが増えている。そこで、「スマートフォンを持つ」という当り前の動作に着目し、各ユーザにとって自然な握り方を認識し、使いやすく片手操作時の利便性を向上させる新しい UI を試作開発した。本稿では、基本コンセプトおよび機能実装手法について解説する。

移動機開発部 つかもと まさかつ† ひぐち ゆうた
塚本 昌克 樋口 雄太
おかだ たかし
岡田 隆

1. まえがき

近年、タッチパネルによる直感的操作が受け入れられ、スマートフォンが爆発的に普及している。またこれに伴い、ディスプレイの大型化や高精細化が進み、端末筐体も大型化している。その結果として、持ちやすさや操作性に影響を及ぼし、指が届かないなど片手操作が難しくなっている。しかしながら片手で操作しなければならないシーンは数多くある。例えば満員電車や手荷物を持っていたり、子どもの手を引いているなどである。そのような時にやむを得ず片手で操作した結果、落としそうになったり、誤った場所をタッチしてしまうなどのケースが増え、ユーザの操作性に対する不満が増えつつある(図1(a))。

端末を小型化すれば、上記の不満は解消できるが、大容量コンテンツを大画面で楽しむメリットが損なわ

れてしまう。この課題に関して、片手操作作用にボタン配置やタッチ調整などのユーザインタフェース(以下、UI)改善を行う方法や、端末を振

るなどのモーション操作などが試みられている。しかしながら、指の可動域や周囲の状況による制限、両手操作との両立などの理由から十分な



操作性の改善には至っていない。また、持ち方によって起動する機能を変更するという研究も行われている [1] [2] が、片手操作性の改善にはつなげていない。

以上のような背景から、「スマートフォンを持っている（把持している）」という当り前の動作に着目し、各ユーザの握り方、いわゆる把持特徴を認識し、握る（Grip）ことで端末操作を行うことができる新しい UI を開発した（図 1（b））。開発した新しい UI により、端末の持ち方を変えたり周囲への影響を意識することなく、片手操作の操作性を向上させることが可能となった。

本稿では、把持特徴を利用した UI（以下、Grip UI）について、把持特徴の定義および端末実装時の要求条件、また、要求条件を満たすための技術課題とその解決手法を示す。さらに、この実現事例として今回試作開発した端末を紹介する。

2. Grip UI 要求仕様

2.1 把持特徴の定義

把持特徴を UI として活用していくためには、端末を把持している状態を数値化しておく必要がある。このため、本開発を始めるにあたり、握っていることの検知に必要な把持特徴を定義した。把持特徴の定義にあたり、把持特徴は動的であり、各個人一律ではないことを意識して検討した。

握っている状態の検知に対して、端末の「どの部分」を、「どれだけ強く」把持しているか区別する必要

がある。このうち、どれだけ強く把持したかは、「圧力の強度」をとらえることで検知できる。一方、どの部分を握ったかに関しては、端末の形状と個人の骨格特徴に依存し、かつ、端末に対し様に力がかかるわけではないため、単なる「接触面積」という指標では正確に検知することが困難となる。そこで「圧力の分布」により圧力の偏りをより詳細に取得することが重要となる。また、端末は常にユーザに持たれている状態が維持されるため、どのような状況で把持されたかを知るためには、「圧力の時間遷移」が UI 利用において必要な特徴量となる。

上記の理由から、以下に示す 3 つの特徴量にて把持特徴を定義した（図 2）。

- 圧力の強度
- 圧力の分布
- 圧力の時間遷移

「圧力の強度」とは、スマートフォンを把持した際に意図的に力を込めたかどうかなど、スマートフォンにかかる圧力の違いを取得するために必要な特徴である。

「圧力の分布」とは、端末にお

ける圧力分布を取得するために必要な特徴である。圧力がかかっている場所は、手の大きさ・筋力の付き方などにより個人差がある。そのため、端末における圧力分布を取得し、個人差を吸収する必要がある。

「圧力の時間遷移」とは、検出した圧力の時間変異を取得するために必要な特徴である。「握る」という動作は、動的な動作であり、個人によって力を込める指の順番も異なってくる。また、圧力検出における時間連続性を評価することで、意図的であるか、偶発的な圧力検出であるかの判断が可能となる。

これら 3 つの特徴量の定義により、Grip UI の実現に必要な情報を十分取得できるものと考えられる。

2.2 端末実装の要件

Grip UI は、取得した把持特徴により端末操作を行う機能である。端末実装に際しては、主に表 1 に示す要件を満足させる必要がある。

項番 1 および 2 は、ハードウェアに関する要件である。Grip UI は、スマートフォン利用を前提にしたユーザの利便性向上が開発の出発点で



図 2 「握っている状態を検知する」上で必要な要件

表 1 Grip UI の要求性能一覧

番号	項目	要求性能
1	ハードウェア	ユーザ属性(年齢, 性別, 体格など)にかかわらず, 把持特徴が取得できること
2		現存する端末と比較して, 持ち方が変わるサイズアップが発生しないこと
3	ソフトウェア	タッチパネルなど, 既存 UI と競合しないこと
4		アプリの修正を行わず, 利用できること
5		検出した把持特徴を端末操作以外に利用できること
6		個人ごとに適した握り方を登録, 利用できること
7	ユーザビリティ	握り方の登録が容易であること
8		Grip UI が実行された事をユーザへ提示すること
9		既存 UI と比較して, ユーザ感覚として反応が遅くないこと
10		現存端末の基本性能を損ねないこと

あり, ユーザの操作感が変わるような筐体サイズの変更を伴ってはならない。そのうえで, 2.1 節で定義した 3 つの特徴量をユーザ属性(年齢, 性別, 体格など)にかかわらず取得できるセンサが, 端末形状に影響を与えないように実装できなければならない。

項番 3 から 5 は, ソフトウェアに関する要件である。Grip UI は新しい UI として端末に実装するが, 有効な局面が既存 UI とは異なるため, タッチパネルやハードキーなどの既存の UI と併用されることとなる。よって, 既存 UI と共存できなければならない。また, UI は, すべてのアプリに関係する機能であるため, 既存アプリや 3rd Party^{*1}アプリなどへの拡張性を保証するためには, アプリの修正を行わないで Grip UI に対応させる枠組みの用意が必要である。さらに, 把持特徴自体はセンサで取得できる物理量であり, 端末操作以外への活用も期待で

きる。この拡張性が担保できる実装であることも望ましい。

項番 6 以降は, Grip UI を端末機能として利用するための要件である。項番 6, 7 は, 新しい機能をユーザに使いやすくするため, 項番 8, 9 は, ユーザに気持ち良い UI として訴求するため, 必要な要件である。

項番 10 は, 項番 1~9 の要件を満たすうえで前提となる要件である。本開発は, UI という端末の基本性能向上を目的としており, トレードオフとして損なわれる機能がないことは重要な検討事項である。

3. Grip UI 機能実装

3.1 触覚センサの最適化

本開発では, 2.1 節で述べた把持特徴のうち「圧力の強度」「圧力の分布」を効率よく検知するセンサとして, 多地点で圧力計測が可能な触覚センサ[3]~[5]を用いることとした。その中でも, 多地点の計測方式として, 圧力センサ素子がアレイ^{*2}状に

組まれている触覚センサを用いた。

本開発で用いた触覚センサの特性として, 圧力センサ素子単体の面積を大きくすると, 低い圧力の強度を検知でき, より高感度での実装が可能となる。一方, センサ素子のアレイ化という観点では, 圧力分布を詳細に把握するため, 圧力センサ素子単体の面積は小さいほうが望ましい。以上の観点から, 触覚センサを構成する圧力センサ素子サイズおよびアレイの最適化を行った(図 3)。本開発では, さらに, 端末形状に影響を与えないことも必要な要件であり, シート状の触覚センサを採用することとした。これらにより, 2.2 節にて示したハードウェア要件を満足させることは可能である。

実際の配置の最適化は, 触覚センサを構成する圧力センサ素子サイズ, センサ素子数, 配置箇所をパラメータとしてさまざまな試作機を準備し, 評価実験を繰返し行うことで最適化を図った。ここで, 素子数が増えるほど, センサ情報を取得するために必要な A/D 変換器や制御 IC に大きな負荷がかかることが問題となる。一般的に消費電流の削減のためにセンシング間隔を長くするなどの対処を行うが, 最適化に際しては, 2.1 節にて述べた「圧力の時間遷移」が十分に検知できるよう考慮した。

3.2 既存 UI との共存

新規センサ組込みに際しては, 既存機能に影響を与えないように AndroidTM*3 OS につなぎこむ必要がある。特に, UI の追加は, Android

*1 3rd Party: 移動端末用ソフトウェアを開発するソフトウェアベンダ。

*2 アレイ: センサ素子を配列状に組み合わせること。

*3 AndroidTM: 米国 Google, Inc. が提唱する携帯端末を主なターゲットとしたオープンソースプラットフォーム。AndroidTM は, 米国 Google, Inc. の商標または登録商標。

OS に対して割込みの形で実装されているため、新規センサを既存 UI デバイスと同じ位置づけで実装すると、競合が発生し操作性の低下を引き起こす可能性がある。

そこで、本開発では、図 4 に示すように、Android OS に用意されているセンサ追加の基本枠組みを利用した。アプリケーションフレームワーク^{*4}に実装される SensorManager^{*5}に対して、標準で用意されている拡張領域に触覚センサ用インタフェースを追加し、Android OS に対して加速度センサ^{*6}などと同等のセンサとして追加する形をとった。すでに搭載されているセンサ類と同じ分類として実装し、アプリケーションフレームワークより低いレイヤーでは既存 UI 機能との独立を図り、UI デバイスとしての競合を回避した。また、この実装を採用することで、センサ単体としてアプリケーションソフトウェアからの利用が可能となり、「握る」などの把持特徴を利用したアプリ開発など、端末操作以外への新規センサ利用の拡張性も確保した。

3.3 UI の統合

3.2 節で述べたように、触覚センサ情報は、Android OS の SensorManager にあがってくるため、SensorManager とアプリケーションを接続して、UI 機能を実現させる必要がある。

本開発では、この機能を実現するために、センサとアプリの間に特別な Service^{*7}を設け、Android 標準にて用意されている UI イベントへ

変換する処理を行わせることで対応した。これにより、Grip UI によるアプリへの入力も既存 UI による入力も、同じレベルの UI イベントとして扱うことができ、アプリの修正を行わなくても利用が可能になる。

図 5 は機能イメージを従来手法と対比して示したものである。図中左に示す従来の構成では、新規インタフェースから発行される Event^{*8}B を受け取るために、対応した EventListener^{*9}B を追加する必要がある。それに対し本開発では、図

中右に示すように、追加した UI スワップサービスにて、触覚センサの値からユーザがどのような状態で握っているか判断し、その状態に応じてユーザが必要とする機能に変換する処理を実行する。そして、特定キーの押下を意図する KeyEvent および特定アプリへの intent^{*10} 発行などの UI イベントへ対応させた。これらのイベント発行機能により、特定アプリの起動や、簡易的なアプリ操作を実現している。

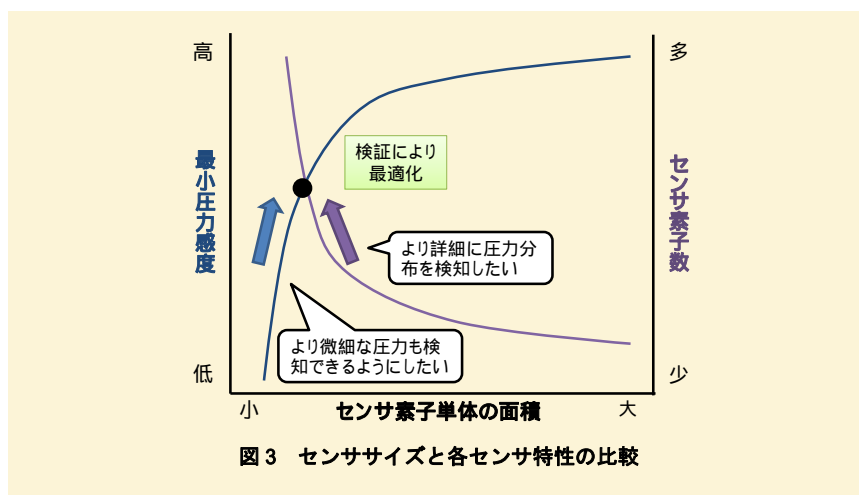


図 3 センササイズと各センサ特性の比較

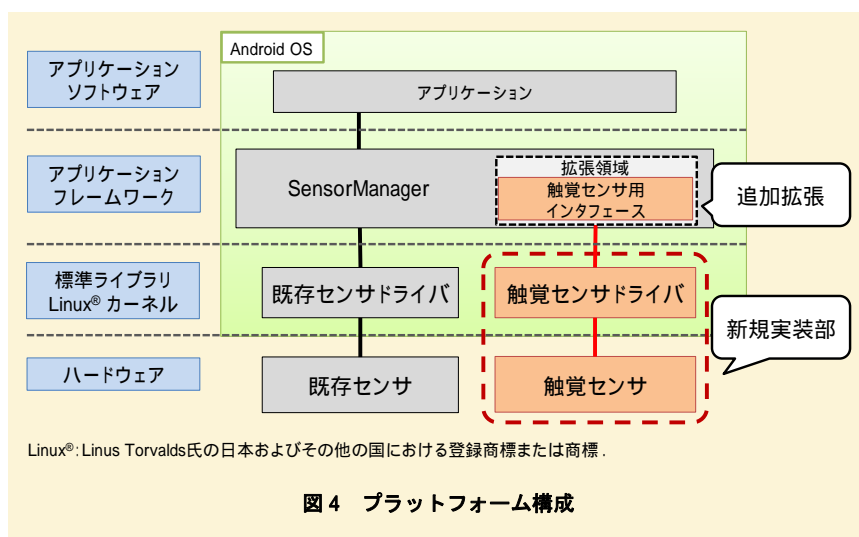


図 4 プラットフォーム構成

*4 アプリケーションフレームワーク：Android OS を構成するアーキテクチャの 1 つ。Android アプリケーションの標準機能を実装するために必要なコンポーネント群で構成されている。
*5 SensorManager：Android アプリ開発で利

用するクラスの 1 つ。各種センサの出力値取得など、搭載されているセンサ利用に際して呼び出す。
*6 加速度センサ：速度の変化を計測するセンサ。携帯端末に搭載することで、携帯端末の傾きや動きなどを検出することが可能と

なる。
*7 Service：Android アプリケーションを構成する要素の 1 つ。視覚的な画面を持たずバックグラウンドで動作する機能。
*8 Event：ボタン操作など、何らかの作用が起こった際にプログラムへ発信される通知。

4. 試作検証結果

4.1 Grip UI 搭載移動機

Grip UI 機能を搭載した試作機の外観を図 6 に示す。この図に示すように、触覚センサは、両側面および背面上部にアレイ状に配置した。表 2 に示すように端末サイズも 1mm 増以下であり、このサイズ増加は、商用段階で十分に吸収できるものである。

この試作機から得られる把持特徴の取得例を図 7 に示す。このように、触覚センサの最適配置を行った結果、2.1 節で述べた把持特徴を取得できる端末が実現できていることが確認できた。

4.2 Grip UI アプリケーション

Grip UI を実現するため、3.2 節および 3.3 節で述べた実装を行うとともに、表 1 で示したユーザビリティ要件を満たすため、ここでは、以下のような機能を実装した。

- ・ UI スワップサービスの起動 / 終了
- ・ UI スワップマップ登録
- ・ 把持テンプレート登録
- ・ 触覚センサパラメータ変更

各機能の設定画面例を図 8 に示す。まず、図 8 上部に示す UI スワップマップ登録機能は、握った入力に対し利用者が発行したい UI イベントを関連付け、UI スワップマップとして登録・編集を行う機能である。ここでは、あらかじめ用意された項目にチェックを入力することで、対

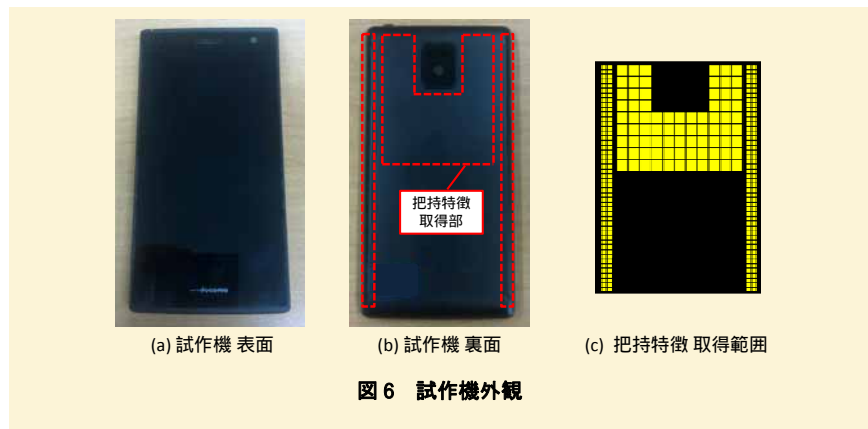
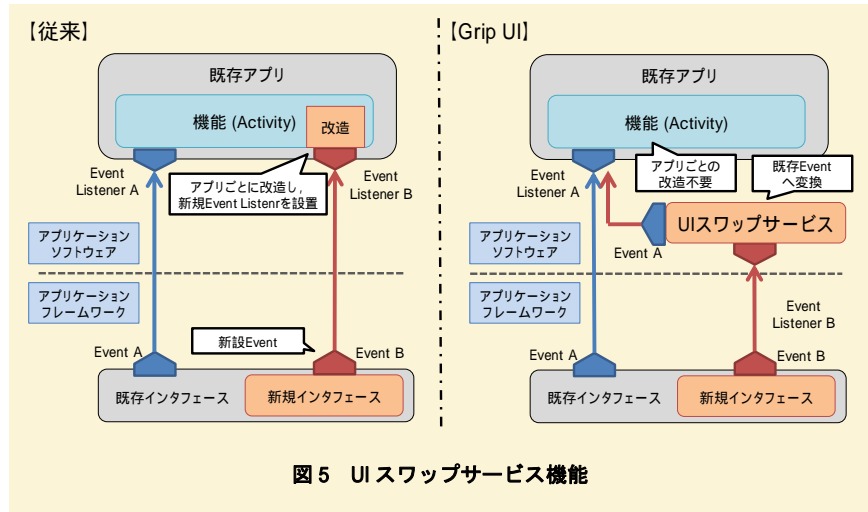


表 2 試作機およびベース機の端末寸法

	試作機	対ベース機比較
高さ	129.5 [mm]	+ 0.5 [mm]
幅	65.6 [mm]	+ 0.6 [mm]
厚さ	10.4 [mm]	± 0.0 [mm]

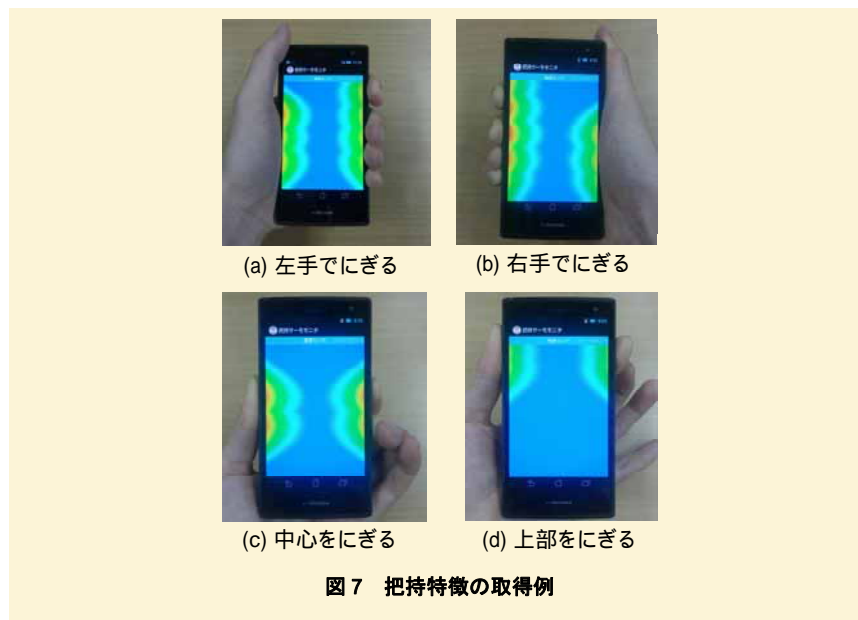
応する起動パターンをすぐ使用することができる。また、握り方、および発行イベントをユーザが自由に選択可能であり、ユーザ独自のカスタマイズを実現している。次に、図 8 下部に示す把持テンプレート登録機能は、握った入力を特定のパターンへ分類するための把持テンプレートの登録・編集を行う。ここでは、主

要な握り方があらかじめ登録されており、マップ登録する際に握り方を容易に指定することができる。また、ユーザが好きな握り方を独自に指定でき、握った位置や強さを画面で確認しながら登録することができる。これにより、ユーザごとに適した握り方や起動イベントを登録でき、各人が使いやすい操作を一連の流れ

*9 **EventListener** : 対応する Event が発生した際にプログラム上で呼び出されるメソッド。

*10 **intent** : Android OS が提供するプログラム間のパラメータ交換手段。アプリケーション内のコンポーネント間でのやりとりや、

アプリケーションをまたがるやりとりに利用される。



ができる。また、従来どおりホームボタンなどでアプリ一覧へ戻り、アプリアイコンを押下することで同様の操作を行うこともできる。このように、ユーザ別に握り方と対応するイベントをあらかじめ登録することにより、状況に応じた端末の操作をより簡易的に提供することができ、かつ通常 UI 操作とも共存が可能な仕組みが実現できた。

5. あとがき

「スマートフォンを握る」という単純なユーザ操作に着目した新しい UI の開発を行った。そして、試作を通して、端末形状を変えることなく、ユーザが端末を握っている状態を詳細に検知できることを確認した。そして、各ユーザにとって自然な握り方を、端末操作へ活用することを可能とした。

本機能は、今回取りあげた片手操作の利便性向上だけではなく、多様な応用先が考えられる。特に、「振る」「叩く」といったモーションと組み合わせることで操作の多様性はさらに向上していくものと考えている。また、端末を握るということ個人特徴ととらえることで、認証機能への応用についても研究がされており[6]、操作性の向上以外の付加価値を端末に与えることも可能になると考える。

今後は、「ユーザの握り方」情報を活用したさまざまな機能との連携により、新しい利用シーンやサービス創出を行っていきたい。



図8 Grip UI アプリケーション設定画面例

で設定できる仕組みとした。

4.3 使用例

図9では、ブラウザアプリを起動している状態から、メールアプリ起

動までの遷移を、Grip UI で操作した場合と通常の UI で操作した場合を示している。あらかじめにぎり方とアプリ起動を登録しておくことで、1回の操作でアプリ起動させること



文献

- [1] K.-E. Kim, W. Chang, S.-J. Cho, J. Shim, H. Lee, J. Park, Y. Lee and S. Kim : "Hand grip pattern recognition for mobile user interfaces," Proc. of the National Conference on Artificial Intelligence, Vol.21, No.2, pp.1789-1794, Jul. 2006.
- [2] H. Lee, W. Chan, J. Park and J. Shim : " New mobile UI with hand-grip recognition," Proc. of CHI'09 Extended

- Abstracts on Human Factors in Computing Systems, pp.3521-3522, ACM, 2009.
- [3] Nextinput Inc. : "ForceTouch." <http://www.nextinput.com/t/ForceTouch>
- [4] タッチエンス(株) : " ショッカキューブ™ ." <http://www.touchence.jp/cube/index.html>
- [5] 日本メクトロン(株) : " フレキシブル触

覚センサ."

http://www.mektron.co.jp/technology/sensor_fpc/

- [6] T. Iso, T. Horikoshi, M. Tsukamoto, T.Higuchi : "Personal feature extraction via grip force sensors mounted on a mobile phone : authenticating the user during key-operation," Proc. of the 11th International Conference on Mobile and Ubiquitous Multimedia, ACM, 2012.