

タッチ入力 of 柔軟な再配置を可能としたインタフェースの作成支援

加藤 邦拓 秋山 耀 宮下 芳明*

概要. 著者らはこれまでに、導電性インクを用いたタッチインタフェースの作成手法を提案してきた。これにより、タッチパネル上のインタフェースをディスプレイ外部に拡張し、実世界上に再配置することを可能とした。本稿では、こうしたインタフェースの作成支援システムを提案する。本システムにより、ユーザはインタフェースのコンポーネントをパネル上に配置するだけで、行いたい動作を実現するインタフェースを作成できる。今回プロトタイプシステムとして、ボタン、スライダ、ダイヤルの3種類を作成するシステムを実装した。

1 はじめに

ユーザはコンピュータを通じ、数多くのソフトウェアや、アプリケーションを使用することができる。インターネットを通じて、PC やスマートフォン、タブレット端末など様々な機器にアプリケーションをインストールでき、必要なものを容易に入手可能になった。しかし、こうしたアプリケーションのユーザインタフェースはそれぞれが決められた位置に固定され、配置を変えられないことが多い。

近年、スマートフォンやタブレット端末などの静電容量式タッチパネルを搭載したデバイスが広く普及している。タッチパネルは画面内の任意の位置にボタンなどのインタフェースを配置可能なため、アプリケーションごとに適したインタフェースを提供できるようになった。また静電容量式タッチパネルは、人体などの接地された導電体が接触することでタッチ認識を行っている。これを利用し、金属などの素材を介してタッチ入力を生じさせる手法がある。この手法を用い、実世界のスライダやボタンなどのタンジブルなインタフェースをタッチパネル上で使用するような、入力手法自体を拡張する手法も数多く提案されている [1]。また、上海問屋から販売された「フィンガータッチパッド」は、縞模様電極が配置されたインタフェースを画面端に貼り付けることで、特定箇所でのタッチ入力だけでなく、スクロールのような連続的な入力を可能としている [2]。

著者らはこれまでに、タッチパネル上に置いたり、貼り付けたりすることで使用可能なインタフェースを導電性インクによって作成する手法を提案した [3, 4]。これらにより、タッチパネル画面上のインタフェースをディスプレイ外に延長し、実世界上で再配置することが可能となった。

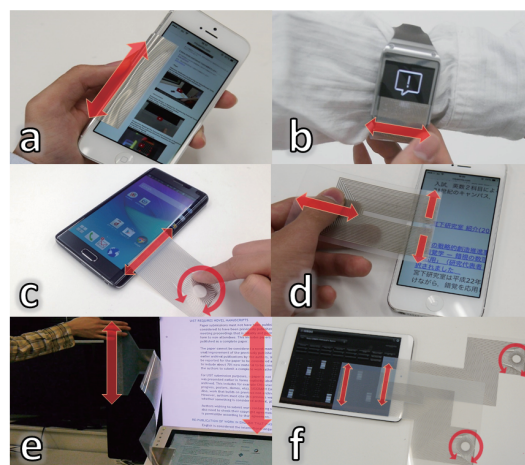


図 1. タッチ入力インタフェースの再配置例: a) スマートフォンの側面入力, b) スマートウォッチのバンドを用いた入力, c) 回転-スクロール操作変換, d) ズームインタフェース, e) 非タッチパネルディスプレイの側面入力, f) シンセサイザアプリ上での使用例

また、ExtensionSticker は、縞模様状に配置した導電性インクによって、連続的な入力を可能とする [4]。印刷パターンの形状を変えることで、例えばユーザの回転操作をスクロールのような直線操作に変換するといった、タッチ入力の拡張手法を実現した。図 1 に ExtensionSticker を用いたタッチインタフェースの拡張例を示す。

しかし、ExtensionSticker の作成には印刷する縞模様パターンの間隔や太さを適切に設定したり、配線同士が接触してはいけないなどの制約があり、それら全てを満たすパターンをユーザがデザインすることは容易ではなかった。本稿では、こうしたインタフェースをユーザ自身が容易に作成可能な環境構築を目指し、導電性インクを用いたタッチインタフェースのデザイン支援システムを提案する。

Copyright is held by the author(s).

* Kunihiro Kato, 明治大学大学院理工学研究科新領域創造専攻デジタルコンテンツ系, Yoh Akiyama, 明治大学理工学部情報科学科, Homei Miyashita, 明治大学大学院理工学研究科新領域創造専攻デジタルコンテンツ系

2 関連研究

Tan らはアプリケーションウィンドウを複数に分割可能とするシステム, WinCuts を提案している [5]. この研究は, 限られたデスクトップ画面のスペースを有効に扱うためのシステムであるが, アプリケーションのインタフェース部分のみをウィンドウごと切り出し, 別の場所に再配置することも可能としている. また, 大江らの Draw-to-Map では, アプリケーション上のスライダなどのインタフェースを, 実世界のデバイスに割り当てて操作する手法を提案している [6]. アプリケーションのインタフェース上でマウスドラッグするだけで, 操作の割り当てを可能とし, 実世界のスライダ操作に連動してアプリ内のインタフェースを操作できる.

3 提案システム

今回, プロトタイプとしてボタン, スライダ, ダイアルの 3 種類を作成可能なデザインシステムを実装した. 本デザインシステムは, インタフェースの作成を行うワークスペースと, ツールボックスからなる. ワークスペースは, グリッド状のパネルで構成されており, このパネル上に作成したいインタフェースのコンポーネントを配置していく. 各コンポーネントは, ユーザがタッチ操作を行う箇所 (入力部) と, タッチパネルに貼り付ける箇所 (出力部) に分かれており, ユーザはそれぞれに対して配置を行う.

まず, ユーザが作成したいインタフェースをツールボックスから選択することで, パネル上に入力部のコンポーネントが配置される. 入力部はパネル上のグリッドに沿って配置され, ドラッグ操作によって配置を決定する. その後, 同様に出力部の配置を行う. これらの操作完了後, 配置した入出力部が自動で配線され, 一つのインタフェースが完成する. また, 各コンポーネントは複数配置もできる. その際, 後から配置されるインタフェースの接続部は, 先に配置されたインタフェースのコンポーネントに重ならないように迂回して配線される. 複数のインタフェースを配置した状態でコンポーネントの移動を行う際, 他のインタフェースの接続部は, 入力部と出力部の配線が可能な範囲で自動的に迂回路を見つけ, 再配線される (図 2). デザインしたインタフェースは PDF として出力でき, そのまま印刷するだけで, タッチパネルに貼り付けて使用できる. 図 1f は, 本デザインシステムによって作成したインタフェースを印刷し, 既存のシンセサイザアプリ上で使用した例である.

4 まとめ

本稿では, 導電性インクによるタッチインタフェースの作成を支援するデザインシステムのプロトタイプ

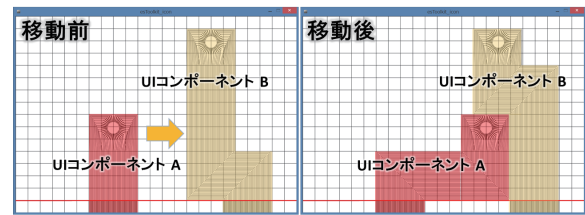


図 2. 接続部の自動配線

プを実装した. しかし, 現在のシステムでは 3 種類のインタフェース作成に留まっており, ユーザの様々な要望に応えるインタフェースの作成には至っていない. 今後はユーザが実現したい形のインタフェースをより綿密に実現可能なシステムの開発を行う. 例えば, ユーザが行いたい入力操作の動きと, それに対して実際にタッチパネル上で行われる操作の動きを描くだけで, その通りの動きを実現するためのパターンを出力するシステムへの発展が考えられる.

アプリケーション開発者は, 全てのユーザに対してそれぞれに適したインタフェースを提供することはできない. また, ユーザ側も完成されたアプリケーションを改変し, 自分の使いやすい形に拡張することはできないのが現状である. 著者らがこれまでに提案したタッチインタフェースの再配置手法はタッチパネル端末のアプリケーションに限らず, 家電製品など, あらゆるタッチインタフェースの拡張・再配置を可能とした. こうした様々な機器に対して, ユーザが望む形のインタフェースを自由に作成できる環境の実現を目指す.

参考文献

- [1] Sven Kratz, Tilo Westermann, Michael Rohs, Georg Essl: CapWidgets: tangible widgets versus multi-touch controls on mobile devices, In *Proc.CHI 2011*, pp.1351-1356, 2011.
- [2] 上海問屋: フィンガータッチパッド, <http://www.donya.jp/item/25894.html>
- [3] 加藤邦拓, 宮下芳明: 導電性インク複合機を用いたマルチタッチパターン生成手法, 情報処理学会研究報告 HCI, ヒューマンコンピュータインタラクション研究会報告 HCI-157, pp.1-6, 2014.
- [4] Kunihiro Kato, Homei Miyashita: Extension Sticker: A Method for Transferring External Touch Input Using a Striped Pattern Sticker, In *Adjunct Proc. UIST 2014*, pp.59-60, 2014.
- [5] Desney S. Tan, Brian Meyers, Mary Czerwinski: WinCuts: Manipulating Arbitrary Window Regions for More Effective Use of Screen Space, In *Proc.CHI EA 2004*, pp.1525-1528, 2004.
- [6] 大江龍人, 志築文太郎, 田中二郎: Draw-to-Map: 物理コントローラと GUI の対応付け手法, 情報処理学会研究報告 HCI, ヒューマンコンピュータインタラクション研究会報告 HCI-152, pp.1-8, 2013.