

# マルチユーザ認識可能なテーブルトップインタフェース “Suzuri”の開発

吉谷 拓真<sup>1</sup> 新井 イスマイル<sup>1</sup>

**概要：**テーブルトップインタフェースは複数人での共同作業環境として、もしくは公共空間におけるデジタルインタフェースとして直感的で有用なものである。しかしながら今日のテーブルトップインタフェースでは複数人のユーザが同時に操作をする際に、どの操作が誰によるものなのかを認識する術を持たない。これからのテーブルトップインタフェースにはグローバルな個人認識可能性と使用に際しての手軽さの両方が求められるが、既存研究の中には両者を現実的に満たすものは存在しない。本研究においてはスマートフォンにより個人を認証、AR技術を用いてスマートフォンを認識し3D深度センサを用いた手のトラッキングによって操作と個人を関連付け手軽に使用することができるテーブルトップインタフェース“Suzuri”の開発を行う。

**キーワード：**テーブルトップインタフェース, マルチユーザ, スマートフォン

## 1. はじめに

テーブルトップインタフェースは複数人での共同作業環境として、もしくは公共空間における情報提供インタフェースとして直感的で有用なものである。しかしながら今日の主なテーブルトップインタフェースは単なる巨大なマルチタッチディスプレイであり、複数人が同時に操作することは可能であってもどの操作が誰によるものかを判別できない。より自然なインタラクションを実現するためには、操作と操作主を関連付ける必要がある。

異なる人による操作をそれぞれの操作主と関連付ける研究はDietzら[1]により初めて実現されたが、操作主は個人と関連づいたわけではない匿名ユーザである。つまり「この操作はBobによるものです」「この操作はAliceによるものです」といったように、特定のユーザによる操作として識別するのではなく「この操作は椅子Aに座った人によるものです」「この操作は椅子Bに座った人によるものです」といったような、その場限りの区別でしかない。テーブルトップインタフェースを共同作業空間として考えた場合、各ユーザごとの作業履歴もしくは各ユーザ独自の設定といったものが保存されている方が自然である。また、テーブルトップインタフェースを公共空間における情報提供インタフェースとして使用する場合には、ユーザの嗜好に合わせた情報を提供した方がよりユーザの満足度は高くなる

と考えられる。操作主の区別というのはその場限りのものではなく、グローバルで永続的な識別である必要がある。

操作主を識別する方法としては、あらかじめ手の輪郭を登録しておくといったもの[2], [3]や、靴の特徴を登録しておくといったもの[4]、手に特別な装置をつけるもの[5]などが研究されているが、いずれもグローバルな認証としては弱かったり、非日常的な装置の装着が必要であったりと現実的とは言いがたい。そこで本研究では認証として確実であり、また人々が日常的に携帯・使用しているものであるスマートフォンに着目した。スマートフォンなどのモバイルデバイスをテーブルトップインタフェースと結びつける研究はLeeら[6]、Wilsonら[7]、Schmidtら[8], [9]によって行われているが、これらはモバイルデバイスをインタラクションの要素として中心に据えており、本研究でのモバイルデバイス(スマートフォン)の使用法とは一線を画するものである。

本稿2節では今後必要となるテーブルトップインタフェースについて考察し、関連研究についての紹介とその課題を述べていく。“Suzuri”とは本研究で提案するテーブルトップインタフェースの名称である。スマートフォンとAR技術を組み合わせた個人認証、深度画像を用いた手のトラッキングを行うことにより個人とタッチ操作を自然な方法で結びつけるものであり、3節でコンセプトを示す。また、4節においてその設計、5節においてその実装の詳細を示し、6節において様々な観点からの評価を行う。7節において本研究のまとめと今後の展望を述べる。

<sup>1</sup> 明石工業高等専門学校 電気情報工学科, Department of Electrical and Computer Engineering, Akashi National College of Technology

## 2. 関連研究とその課題

まず前提として、テーブルトップインタフェースの役割を複数人での共同作業空間と公共空間における情報提供インタフェースとして定義する。前者の定義においては個人の識別可能性が求められる。個人識別可能であると、作業履歴や作業状態、個人ごとの設定などを保存することが可能となり、より自然かつ便利であるからである。後者の定義においてテーブルトップインタフェースには特別な装置や姿勢などの制約がなくカジュアルに手軽に使えることが求められる。以下、個人の識別可能性と使用の手軽さの観点から関連研究とその課題を考察していく。

操作と操作主を関連付ける研究としては Dietz らによる DiamondTouch[1] が先駆けである。この研究で使用されるタッチディスプレイは椅子に座った人が画面をタッチするとテーブル-人-椅子でできあがる回路の容量が変化することを利用してタッチを検出している。画面の位置ごとに異なる周波数の電流信号を流しており、回路に流れる電流の周波数解析によって複数のタッチポイントを求めることができる。それぞれの回路を別々に監視しているため、「どの椅子に座ってる人がどのタッチをした」ということが機構的に分かるようになっている。しかしながら DiamondTouch は使用に際して必ず椅子に座る必要があり、使用姿勢に大きな制約がある。これは公共空間での利用を考えた際には欠点となる。また、DiamondTouch はある操作がどの椅子に座った人によるものなのか以上のことは分からないので、共同作業空間としての利用を考えた際に重要なポイントである個人識別性を十分には満たしていない。

テーブルトップインタフェースにおいて、手をトラッキングするというアイデアを実装した例としては Dohse らによるもの [10] が挙げられる。この研究において Dohse らはテーブルの上方に設置されたカメラ画像から肌色部分を検出し、それによって手のトラッキングを実現した。画面に偏光フィルタを貼ることにより画面に肌色が映っても誤認識なく手をトラッキングできるようにする、といった工夫が行われている。この研究でも個人と手の簡潔な関連付け手法は示されていない。また、この方法だと手袋をしている場合に手を認識不可能である。

個人と手を関連付けた研究としては Schmidt ら [2], [3], [8], [9] や、Richter ら [4] によるものがある。Schmidt らの HandsDown[2] は手の輪郭を認証キーにして登録されているユーザと登録されていないユーザを区別している。半透明のすりガラスのようなテーブルトップに下からプロジェクタで映像を投影、またそこにカメラも設置されておりテーブルトップを下から監視しているタイプのテーブルトップインタフェースであり、テーブルトップに手を置くとその輪郭で個人を識別し、その個人としての操作が可能となっている。また HandsDown のシステムを基に

した IdLenses[3] は片方の手でスコープをつくり、そのスコープ内は自分専用の空間であるとし、そこで起きたタッチをその個人に関連付けるというものである。手の輪郭を生体認証キーとして使用するアイデアは非常に面白いはあるが、どの程度の数のユーザを識別できるのかということに関しては未知数である。同論文内では 500 人を識別できたことが述べられているが、この数は一つの施設内のある一定の集団内での使用には問題無いと考えられるが公共空間における情報提供インタフェースとしてのテーブルトップインタフェースを考えた場合、不特定多数の人物を識別する必要があり十分な数であるとは言えない。Richter らの Bootstrapper[4] も同様の仕組みのテーブルトップインタフェースを使用し、テーブルのユーザが立つ一端に下向きに深度センサ付きカメラ (Microsoft Kinect<sup>\*1</sup>) を設置、得られたカラー画像と深度画像から靴を抽出し、その靴によって個人を識別するというインタフェースである。Bootstrapper の問題点は HandsDown などと同様、使用人数に限りがあるということである。また、公共空間での使用を考えた場合、大量生産されている靴では個人を一意に識別することは不可能である。またその日の気分や天候により複数の靴を履き替える人もいるなど、靴と個人の結びつきは強固なものとは言い難い。

手を個人と関連付ける確実な方法としては軽量デジタルデバイスの使用が考えられる。軽量デジタルデバイスの補助によってそのデバイスが発する固有 ID と個人とを結びつけることで、手の輪郭や靴といった不安定な要素での認証ではなく確実な認証が可能となる。手に腕時計のように特殊な装置を装着しこれを実現した例 [5] などがあるが、このような特殊なデバイスの装着をユーザに要求することは手軽さと現実性を損なう。

個人を一意に識別でき、かつ手軽で誰もが持っているデジタルデバイスとして真っ先に思いつくのは携帯電話である。携帯電話をテーブルトップインタフェースと結びつけた例としては Lee らの u-Table[6] や、Wilson らの BlueTable[7]、Schmidt らの PhoneTouch など [8], [9] が挙げられる。u-Table では携帯電話の裏側にタグマーカを貼り付け、テーブル下に設置されたカメラでその位置を把握、携帯電話の周りに操作メニューなどを表示するシステムである。この方法では予め携帯電話の裏側にタグマーカを貼り付ける必要があり、現実的とは言えない。BlueTable ではテーブル上方に設置されたカメラ画像から、画像処理によって携帯電話の位置を認識する。携帯電話は Bluetooth による無線通信でホストシステムと接続され、固有のパターンの信号を携帯電話の赤外線通信機能を利用し赤外線としてテーブル上に照射する。カメラでそのパターンを認識することで携帯電話とテーブルトップインタフェー

\*1 <http://www.xbox.com/en-US/kinect/>

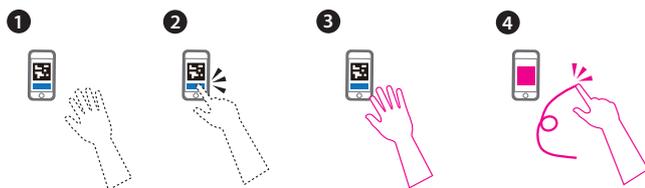


図 1 Suzuri の使用方法

スが接続される。近年では赤外線通信を搭載した端末が少なくなってきたことを考えると、この手法は将来汎用性を損ねる可能性があり、また同研究ではどのタッチが誰のタッチであるかという問題は主題ではないので当然解決されていない。PhoneTouch などの研究 [8], [9] において Schmidt らは従来のマルチタッチに加え携帯電話をスタイラス代わりに使用可能なテーブルトップインタフェースを提案した。同研究においては携帯電話によるタッチと手によるタッチは切り離されて処理されているため、手でなされたタッチに関しては、どのタッチが誰のタッチであるかの認識は実現できていない。携帯電話でのタッチは個人と識別することも可能であると考えられるが、その場合シングルタッチになってしまい自然な操作ができるとは言い難い。

以上の議論をまとめると、現状としては手によるタッチ操作を個人と確実に関連付けて識別するマルチタッチテーブルトップインタフェースは存在しない。公共空間で使用可能であるために、我々にはそのようなテーブルトップインタフェースを実現する仕組みが必要である。

### 3. Suzuri の提案

前節の議論を踏まえると、現状のテーブルトップインタフェースはマルチユーザであっても手軽さを欠いていた、公共空間での使用に現実的な難点があったりすることが見えてくる。そこで本研究では自然かつ現実的に手によるタッチ操作と個人を関連付けることが可能なテーブルトップインタフェースである“Suzuri”を提案する。

#### 3.1 Suzuri のコンセプト

Suzuri は公共空間に置かれることを前提としたマルチユーザテーブルトップインタフェースである。特定の施設や集団内での利用のみならず、公共の場で不特定多数のユーザに使用されることを想定している。そのため個人の認証はユーザ数が大規模であっても確実に実行される必要がある、また公共空間で手軽に使用可能であるために非日常的な特殊装置の装着が不必要である必要がある。そこで Suzuri はユーザの認証にスマートフォンを用い、個人と関連づいたタッチ操作を自然に行うため図 1 のような使用感をもたせる。図 1 の各ステップの説明を以下に記す。

(1) スマートフォンで指定 URL にアクセスし、Suzuri のサイトでログインをすると AR マーカが表示される。

(2) スマートフォンをテーブルトップに静置し、マーカの下ボタンをタップする。  
(3) ボタンをタップした手はそのスマートフォンの持ち主の手であると認識される。  
(4) 以後その手で行ったテーブルトップインタフェースへの操作はそのスマートフォンの持ち主による操作であるとして処理される。

テーブルトップ上に複数のスマートフォンを置かれた場合でも同様にそれぞれの手はそれぞれのスマートフォンに対応して認識されるので、個人とタッチ操作の自然な関連付けが可能となる。この操作感はスマートフォンを書道の硯、手を書道の筆と見立てると理解しやすいことから“Suzuri”と命名した。次項以降で、このコンセプトが技術的にも実用的にも現実的であることを示す。

#### 3.2 スマートフォンを用いた個人認証と集合知の情報サービス

近年のスマートフォンの台頭はユビキタス社会 [11] を現実的なものとしてきている。皆が携帯し、様々なセンサを搭載し、高度な Web 体験も可能で直感的なインタフェースを備えたスマートフォンは現実世界と Web 上の仮想世界の直感的な架け橋であるといえる。常に携帯されることによりスマートフォンは個人の日常と深く関係を持ち、単に個人を識別する道具としての利用にとどまらず、行動履歴などから嗜好を割り出すといった近年の Web サービスによく見られる集合知的な情報サービス提供のための情報源としての利用も考えられる。スマートフォンをテーブルトップインタフェースにおける個人認証のキーとすることで、手軽かつ確実な個人認証といった利点のみならず、個人や環境に特化した、現実世界と仮想世界のより直感的でシームレスな連結を可能とする。

昨今の集合知を用いた Web サービスのレコメンデーションなどは基本的には個人単位である。複数人が集まって同時に操作するテーブルトップインタフェース上であれば、それら個人と個人の関係性（家族、友人、恋人など）も考慮した、よりコンテキストアウェアネスの高い情報サービスの提供が可能となる。このような情報サービスは主にショッピングモールなどの公共空間での利用の際に価値を持つと考えられる。

#### 3.3 AR 技術を用いたスマートフォンの認識

Suzuri ではテーブルトップに置かれたスマートフォンの位置を個別に認識する必要がある。そのためにスマートフォン画面上にマーカを表示するのであるが、このマーカには AR 用のマーカを使用する。スマートフォンはテーブルトップに無造作に置かれることを想定するのでその向きなどがバラバラであるのだが、AR マーカであればそのような回転や歪みにロバストであるので実装が容易になる。

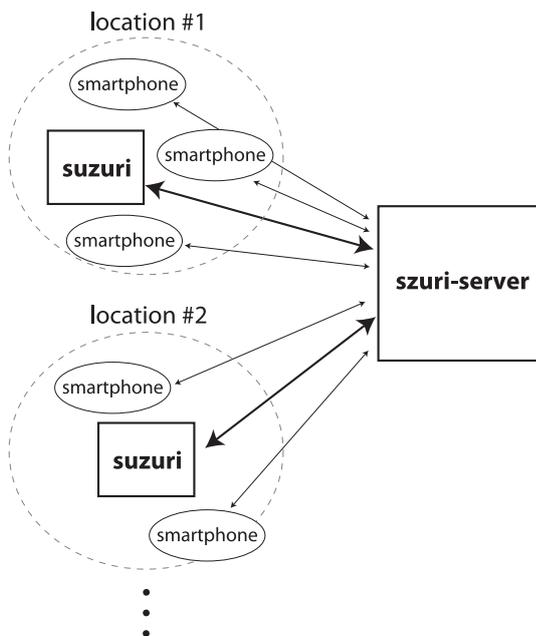


図 2 大局的な構成

### 3.4 深度センサを用いた手の認識

3次元深度情報を用いることで前景と背景の切り分けが容易かつ確実なものとなるので、Suzuri ではこのようにして得られた前景から手を検出しトラッキングを行う。

## 4. 公共空間における Suzuri ネットワークと本研究での実装箇所

### 4.1 公共空間に散在する Suzuri

3.1項で述べたように Suzuri は公共空間に設置されることを前提としたテーブルトップインタフェースである。街中の様々な場所に置かれることを想定しているため、大局的な構成としては図 2 のようになる。図中において suzuri とは各テーブルトップインタフェースであり、その上に複数のスマートフォンが乗っている。suzuri もスマートフォンも suzuri-server という中央サーバに接続され、suzuri-server 上ではテーブル (suzuri) とスマートフォンの対応関係の処理や、タッチ情報や個人情報のやり取りが行われる。

### 4.2 本研究で実装する構成

本研究では前節で述べられた大局的な構成の中の 1 ノードを実装する。本研究で実装する構成を図 3 に示す。スマートフォンと suzuri-server の間ではリアルタイムの双方向通信が行われる。suzuri は suzuri-camera と suzuri-table から成り、suzuri-camera は手のトラッキングとマーカの認識やタッチと手の関連付け処理を、suzuri-table はタッチの単純な検出と情報の表示を行い、それぞれリアルタイム双方向通信で情報のやり取りを行う。

#### 4.2.1 suzuri-server

suzuri-server は Web サーバである。ユーザはスマートフォンから suzuri-server に接続し、ユーザ固有 ID を用い

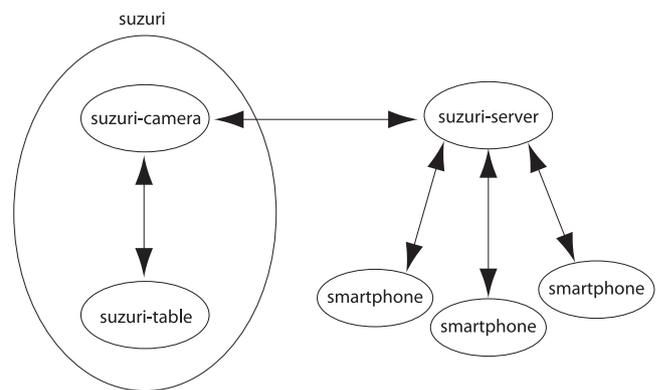


図 3 本研究で実装する構成

て Suzuri システムにログインする。表示されるマーカはログインセッションごとに異なるパターンが表示され、少なくとも同一テーブル上ではユニークである。スマートフォンから suzuri へ、またはその逆への情報のやり取りをリアルタイムに行う。また、ユーザの情報をデータベースにストアする役割も持つ。

#### 4.2.2 suzuri-camera

suzuri-camera はテーブル上を監視するシステムである。スマートフォン上に表示されるマーカの認識とトラッキング、ユーザの手の認識とトラッキングを行う。また、suzuri-table から送られてくるタッチ情報と、suzuri-server からログイン状況を得ている個人の関連付けの作業も suzuri-camera が行う。suzuri-camera は Suzuri の中心的なシステムである。

#### 4.2.3 suzuri-table

suzuri-table は各タッチ操作に ID を割り振り、suzuri-camera に送信する。suzuri-camera からはタッチ ID に個人 ID を加えた情報を送り返してくるので、その拡張されたタッチ操作情報を基にディスプレイにアプリケーションの目的に応じた表示を行う。つまり、テーブルデバイス上のアプリケーションのベースとなる共通的な呼称である。ここで作成可能なアプリケーションには基本的には制約は無いが、suzuri-camera また suzuri-server と通信を行う際には所定の形式に則る必要がある。

## 5. Suzuri の実装

本章では Suzuri を実装方法について述べる。5.1 項に Suzuri の機器構成を示す。4 節で述べられた設計を基に suzuri-camera, suzuri-server, suzuri-table に分けてそれぞれ 5.2 項, 5.3 項, 5.4 項にソフトウェア実装の詳細を述べる。

### 5.1 機器構成

表 1 に使用機材一覧を、図 4 に Suzuri の機器構成を示す。suzuri-camera と suzuri-server は MacBook Air 上で動作させ、suzuri-table は VAIO Tap 20 上で動作させる。VAIO Tap 20 がテーブルトップインタフェースのテーブ

表 1 使用機材

名称	規格	製造会社名
MacBook Air	Mac OS X v10.7.5 / Intel Core i5 1.6GHz / RAM 4GB	Apple Inc.
Kinect		Microsoft
VAIO Tap 20	Windows 8 / Intel Core i5 1.7GHz / RAM 4GB / 20 インチ 10 点タッチディスプレイ	ソニー株式会社
ホワイトラック	90x45x150 [cm]	コーナン商事
iPhone 5	iOS 6.1	Apple Inc.
iPhone 4	iOS 6.0	Apple Inc.
iPod touch	iOS 6.0	Apple Inc.
Galaxy Nexus	Android 4.1.2	サムスン電子

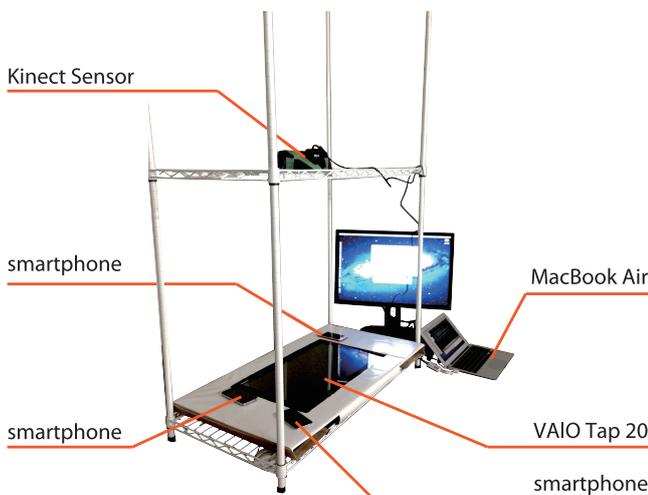


図 4 Suzuri の外観

ルトップの部分であり、その上方にはテーブルとその周辺を望むように Kinect センサを下向きに配する。

## 5.2 openFrameworks による suzuri-camera の実装

suzuri-camera の実装には openFrameworks<sup>\*2</sup> という C++ のメディアアート用のフレームワークを用いた。様々な機能がアドオンという形で提供されており<sup>\*3</sup>、マルチメディアシステムの構築において迅速かつ効率的な実装が可能となるからである。

### 5.2.1 ofxKinect による深度情報の取得

openFrameworks で Kinect を活用する際には ofx-OpenNI<sup>\*4</sup> と ofxKinect<sup>\*5</sup> の 2 つの選択肢があるが、今回の実装ではより軽量の ofxKinect を用いて RGB 画像と深度情報を取得することにした。

### 5.2.2 ofxBlobTracker による輪郭の抽出

設定された閾値以上の距離の部分の深度情報データを用

<sup>\*2</sup> <http://www.openframeworks.cc/>

<sup>\*3</sup> <http://ofxaddons.com/>

<sup>\*4</sup> <https://github.com/gameoverhack/ofxOpenNI.git>

<sup>\*5</sup> <https://github.com/ofTheo/ofxKinect.git>

いてカットしたデータから二値画像を生成し、OpenCV<sup>\*6</sup> で輪郭抽出を行いプロブを抽出する。このプロブをトラッキングすることによって手のトラッキングとする。プロブの形状から鋭利な角度を持った部分を指と認識する。これらの処理には ofxBlobTracker<sup>\*7</sup> を用いた。

### 5.2.3 ofxARToolkitPlus によるマーカの認識

RGB 画像をモノクロ画像に変換し、さらに明度に閾値を設定することで二値画像を得る。この二値画像から AR マーカを認識し位置を取得する。マーカの認識には ARToolKit の改良版である ARToolkitPlus を openFrameworks のアドオン化した ofxARToolkitPlus<sup>\*8</sup> を使用した。BCH 符号化による誤り訂正をサポートしており、マーカは 8x8 の白黒のパターンで全部で 4096 種類ありそれぞれのマーカには ID が割り振られている。

### 5.2.4 ofxOsc による OSC 通信の実装

OSC (OpenSound Control<sup>\*9</sup>) は MIDI<sup>\*10</sup> に代わる通信プロトコル [12] であり、マルチメディアの現場で機器間の情報のやり取りに広く使われている。ofxOsc は openFrameworks に組み込まれている OSC 通信のためのアドオンであり、suzuri-camera では ofxOsc を使用した。

## 5.3 Node.js による suzuri-server の実装

Node.js<sup>\*11</sup> はイベントドリブンな入出力を扱う、サーバサイド JavaScript 環境である。suzuri-server の実装にはリアルタイム処理を記述しやすい Node.js を使用した。

### 5.3.1 Socket.IO, MongoDB によるリアルタイム Web アプリ

suzuri-server ではスマートフォンと suzuri-server の通信に Node.js のライブラリで WebSocket によるリアルタイム通信を容易に実装可能な Socket.IO<sup>\*12</sup> を使用した。また、suzuri-server のデータベースとして MongoDB<sup>\*13</sup> を使用した。

### 5.3.2 Passport による Facebook 認証

ユーザは Suzuri Web サイトにアクセスした後、Facebook でのログインを求められる。この Facebook 認証の部分は Passport<sup>\*14</sup> という Node.js のライブラリを使用して実装した。

### 5.3.3 node-osc による OSC 通信の実装

suzuri-server と suzuri-camera の間の通信には OSC を用いている。suzuri-server の側では Node.js で OSC 通信

<sup>\*6</sup> <http://opencv.org/>

<sup>\*7</sup> <https://github.com/patriciogonzalezvivo/ofxBlobTracker>

<sup>\*8</sup> <https://github.com/fishkingsin/ofxARtoolkitPlus.git>

<sup>\*9</sup> <http://opensoundcontrol.org/>

<sup>\*10</sup> <http://www.midi.org/>

<sup>\*11</sup> <http://nodejs.org/>

<sup>\*12</sup> <http://socket.io/>

<sup>\*13</sup> <http://www.mongodb.org/>

<sup>\*14</sup> <http://passportjs.org/>

を可能にする node-osc<sup>\*15</sup>を使用した。

#### 5.4 Cinder による suzuri-table の実装

suzuri-table の実装には Cinder<sup>\*16</sup>という openFrameworks に似た C++のメディアアート用フレームワークを使用した。suzuri-table は Windows の端末で実装したので、Mac での開発が主な openFrameworks では動作の不具合が生じる懸念があったからである。また、Cinder は公式にマルチタッチへの対応を謳っているため suzuri-table の実装は Cinder で行うことにした。

##### 5.4.1 OSC ブロックによる OSC 通信の実装

Cinder も OSC 通信用のライブラリが用意されており、suzuri-table の OSC 通信部には Cinder の OSC ライブラリである OSC ブロックを使用した。

##### 5.4.2 multiTouch API によるマルチタッチの実装

マルチタッチの実装は Cinder の multiTouch API を用いて実装した。各タッチ操作には ID が割り振られ、タッチの開始、移動、消失の間は固有であることが保証される。suzuri-table は各タッチイベント発生時にタッチ ID とその座標を suzuri-camera に送信している。

#### 5.5 メッセージのやりとり

##### 5.5.1 手の認識までの流れ

client がマーカー直下のボタンをタップすると tap イベントが発火し suzuri-server へ伝わる。この情報はそのまま suzuri-camera へ送信され suzuri-camera はマーカーに最も近い指が属する手をそのユーザの手と認識する。手を認識した際 suzuri-camera は suzuri-server へ認識したユーザの ID と、現在トラッキングされている手の本数を suzuri-server へ送信する。また、手を見失った場合も同様である。この情報はそのまま client へ届けられるので、client は自分の持ち主の手が認識されたかどうかということがリアルタイムで分かる。また、suzuri-camera はマーカーを認識した際と消失した際に suzuri-server へそのマーカー ID を伝えているので、client は自分のマーカーが現在トラッキングされているのか否かということが分かる。

##### 5.5.2 タッチとユーザの関連付けの流れ

suzuri-table 上でのタッチ操作はそれぞれ began/moved/end イベントを持ち、それぞれタッチの開始時、移動時、終了時に発火する。これらのイベント発火時に suzuri-table はその旨をタッチ ID とタッチ座標とともに suzuri-camera へ伝える。began イベントを受け取った suzuri-camera はそのイベントが発生した場所に最も近い手をその操作主として処理する。この手がユーザと関連付けられている場合、suzuri-table へタッチ ID とともにユーザ ID を送信することで suzuri-table にタッチとユー

ザの関連性を伝える。もしその手が未だ匿名ユーザである場合、ユーザ ID としての形式が外れた情報を送ることで suzuri-table にそのタッチは誰とも結びつかなかったということを知らせる。

#### 5.6 お絵かきアプリケーションの実装

本研究では Suzuri の概念を用いてお絵かきアプリケーションを実装した。

##### 5.6.1 仕様

本研究で実装したお絵かきアプリケーションの仕様を以下に列挙する。

- 指でマーカーの下にあるボタンをタップすると画面が遷移し線の色と太さを変えることができる。
- スマートフォンと関連づいてトラッキングされた手によって描かれた線はそのユーザが設定した色・太さで描画される。
- スマートフォンに関連づいてトラッキングされていない手によって描かれた線は色を持たず、タッチが終わると同時に消失する。
- 複数の人がそれぞれ異なる色、太さの線を同時に描くことができる。
- 手のトラッキングが切れるとスマートフォンの画面はマーカーの画面に戻る。

##### 5.6.2 操作の流れ

- (1) まずスマートフォンで Suzuri の Web サイトにアクセスし facebook アカウントを用いてログインする。
- (2) ログインに成功すると自動的にマーカーページ (図 5) へ遷移する。
- (3) マーカー下の青いボタンを指先でタップする (図 6) ことで手のトラッキングがはじまる。手のトラッキングに成功すると線の属性設定画面へ自動的に遷移し、それぞれが設定した色・太さで絵を描くことができる (図 7)。

#### 6. 評価

##### 6.1 Suzuri と既存研究の機能面での比較

Suzuri は既存研究に比べ以下のような機能的に優れた点を有する。

- 操作と操作主を個人レベルで特定可能である。
- 個人の識別が特定の集団内にとどまることはない。
- 個人の認証方法がシンプルで直感的かつ確実である。
- スマートフォンの画面を補助のディスプレイとして使用可能である。
- 特殊な装置を必要としない。
- 使用姿勢に制約がない。

##### 6.2 トラッキング可能な手の本数

原理的には何本であっても手をトラッキングすることは

\*15 <https://npmjs.org/package/node-osc>

\*16 <http://libcinder.org/>



図 5 マーカページ

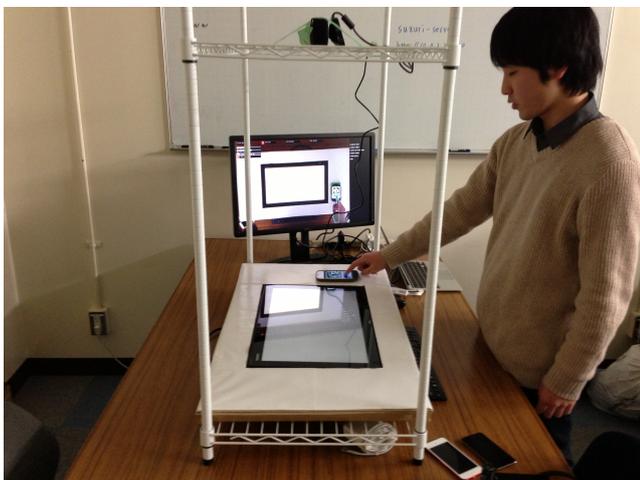


図 6 マーカ直下の青いボタンをタップし手を認識させる

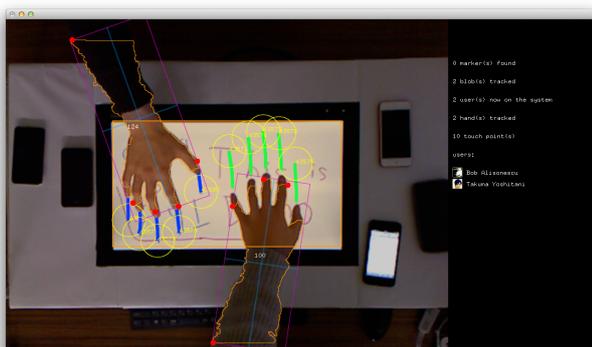


図 7 2人のユーザがそれぞれ5本の指で線を描いている様子

可能である。この点は OpenCV の ContourFinder の性能に全てを負う。

### 6.3 Kinect センサとテーブルの距離について

Kinect センサをテーブルから離すことで視野が広がり手のトラッキング範囲が増えるが、マーカが相対的に小さく

なり認識精度が落ちてしまう。試行錯誤の結果 Kinect センサとテーブルの間の距離を 690[mm] よりも大きくするとマーカの認識率が極端に落ちることが判明した。よって本研究の実装では Kinect センサとテーブル間の距離は 690 [mm] に設定してある。これだけの距離が離れている場合、使用可能なマルチタッチディスプレイはワイドディスプレイであれば最大 32 インチ程度となる<sup>\*17</sup>。それ以上に大きなディスプレイを使用する場合には、マーカの認識精度を上げる工夫をした上で Kinect センサをもっと上に持ち上げるか、もしくは複数の Kinect センサを一定の間隔で並列させ撮影エリアを拡大するといった方法が考えられる。

### 6.4 AR マーカのパターン数について

低解像度の RGB カメラでもマーカを認識するためにはマーカのパターンをシンプルにする必要があるが、その場合必ずとマーカの種類の限定的になる。ユーザとマーカをグローバルに 1 対 1 対応させ場合にこのことは重大な問題となるが、マーカはテーブル上でユニークであるという制約のみで解決する。しかしながら図 2 のような構成の場合はスマートフォンはテーブルとは直接的な接続を持たないために現在どのテーブル上に置かれているのかということが分からない。このことに対処する方法としては、無線 LAN のフィンガープリントを取得することでどのテーブル上に置かれているかを推定する方法、もしくはテーブル自体を無線 LAN のアクセスポイントにしてスマートフォンとテーブルの間に直接的なコネクションを作ってしまう方法、もしくはマーカを時間的に切り替えその時系列的パターンをユーザに固有なものにするといった方法などが考えられる。

### 6.5 最大同時タッチ数について

今回の実装では VAIO Tap 20 をマルチタッチディスプレイとして使用した。このコンピュータの最大同時認識可能タッチ数は 10 である。Suzuri の最大同時タッチ数はこれに完全に負うため、最大で 10 本の指まで操作が可能である。このことから、同時に利用する人数は 2~4 人であることが望ましい。これ以上の人数で利用するにはタッチ方式を CCV<sup>\*18</sup>や reacTable<sup>\*19</sup>のような光学式のものに切り替え最大同時タッチ認識数を増やすなどの方法が考えられる。

<sup>\*17</sup> 画面端のタッチも検出可能にするための遊びを 4 辺それぞれ 15cm 程度取るとした仮定の上で、Kinect センサの仕様 (<http://msdn.microsoft.com/en-us/library/jj131033.aspx>) を基に計算した値

<sup>\*18</sup> <http://ccv.nuigroup.com/>

<sup>\*19</sup> <http://www.reactable.com/>

## 6.6 スマートフォンをタッチディスプレイ上に置くこと について

今回の実装で使用した VAIO Tap 20 のディスプレイは静電式のタッチディスプレイである。実験に使用したスマートフォンのひとつの iPhone を画面の上に置いて時、iPhone 裏面の樹脂部分がタッチのポイントとして誤認識されてしまい、前節で述べた同時タッチ数 10 のうち数個を奪ってしまうため、同時タッチ数が体感的にかなり減る現象が発生した。そこで今回の実装ではスマートフォンはテーブルの上ではなく脇に置くような使用方法を提案している。

## 6.7 手のシルエットのオクルージョンについて

深度の閾値画像から得られるのは手の輪郭、すなわちシルエットのみである。手のプロブが孤立している場合はそれをトラッキングすればいいだけであるが、オクルージョンが発生してしまった場合には特別な処理が必要である。今回の実装ではオクルージョン処理は未実装であるため、ユーザは互いの手をかぶらせたり互いの手に触れたりしてはいけな。しかしながらマルチユーザテーブルトップインタフェースとしての実用性を考えた場合、この制限は不自然である。このことからオクルージョン処理は重要な課題であるが、Papadourakis らの研究 [13] や Iason らの研究 [14] を応用することで解決可能であると考えられる。

## 7. まとめと展望

本稿では既存のテーブルトップインタフェースの課題を示し、それらを自然で現実的な方法で解決するマルチタッチマルチユーザを認識可能な Suzuri というテーブルトップインタフェースを提案した。Suzuri は個人の識別にスマートフォンを用い、手をトラッキングすることによって個人と手そしてタッチ操作を結びつけた自然な状態でのマルチユーザテーブルトップインタフェースを実現した。

Suzuri は特に、公共空間という不特定多数のユーザがいる環境でも利用可能である点で類似研究とは一線を画している。マルチユーザ認識可能な共同作業空間の実現は、インタフェースの観念を転換させ、より人間とコンピュータが自然にインタラクションを起こすことができる真のユビキタス社会への前進であると考えられる。

今後はアプリケーションを構築するための環境を整える(モジュール間の汎用的な情報伝達手法の整備、API の整備など) ことでアプリケーションにバリエーションを増やしていくと同時に、ハードウェアの制約や手のシルエットのオクルージョンの問題を解決していく必要がある。

## 参考文献

[1] Paul Dietz and Darren Leigh. Diamondtouch: a multi-user touch technology. In *Proceedings of the 14th annual*

*ACM symposium on User interface software and technology*, UIST '01, pp. 219–226, New York, NY, USA, 2001. ACM.

[2] Dominik Schmidt, Ming Ki Chong, and Hans Gellersen. Handsdown: hand-contour-based user identification for interactive surfaces. In *Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries*, NordiCHI '10, pp. 432–441, New York, NY, USA, 2010. ACM.

[3] Dominik Schmidt, Ming Ki Chong, and Hans Gellersen. Idlenses: dynamic personal areas on shared surfaces. In *ACM International Conference on Interactive Tabletops and Surfaces*, ITS '10, pp. 131–134, New York, NY, USA, 2010. ACM.

[4] Stephan Richter, Christian Holz, and Patrick Baudisch. Bootstrapper: recognizing tabletop users by their shoes. In *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems*, CHI '12, pp. 1249–1252, New York, NY, USA, 2012. ACM.

[5] Dominik Schmidt, Corina Sas, and Hans Gellersen. Personal clipboards for individual copy-and-paste on shared multi-user surfaces. CHI '13, 2013.

[6] Jangho Lee and Jee-In Kim. u-table: a tabletop interface for multiple users. In *Proceedings of the 6th international conference on Computational Science and Its Applications - Volume Part I*, ICCSA'06, pp. 983–992, Berlin, Heidelberg, 2006. Springer-Verlag.

[7] Andrew D. Wilson and Raman Sarin. Bluetable: connecting wireless mobile devices on interactive surfaces using vision-based handshaking. In *Proceedings of Graphics Interface 2007*, GI '07, pp. 119–125, New York, NY, USA, 2007. ACM.

[8] Dominik Schmidt, Fadi Chehimi, Enrico Rukzio, and Hans Gellersen. Phonetouch: a technique for direct phone interaction on surfaces. In *Proceedings of the 23rd annual ACM symposium on User interface software and technology*, UIST '10, pp. 13–16, New York, NY, USA, 2010. ACM.

[9] Dominik Schmidt, Julian Seifert, Enrico Rukzio, and Hans Gellersen. A cross-device interaction style for mobiles and surfaces. In *Proceedings of the Designing Interactive Systems Conference*, DIS '12, pp. 318–327, New York, NY, USA, 2012. ACM.

[10] K.C. Dohse, T. Dohse, J.D. Still, and D.J. Parkhurst. Enhancing multi-user interaction with multi-touch tabletop displays using hand tracking. In *Advances in Computer-Human Interaction, 2008 First International Conference on*, pp. 297–302, feb. 2008.

[11] Mark Weiser. The computer for the 21st century. *SIG-MOBILE Mob. Comput. Commun. Rev.*, Vol. 3, No. 3, pp. 3–11, July 1999.

[12] Matthew Wright Center and Matthew Wright. Opensound control: State of the art 2003. In *Proceedings of the 2003 Conference on New Interfaces for Musical Expression (NIME-03)*, pp. 153–159, 2003.

[13] Vasilis Papadourakis and Antonis Argyros. Multiple objects tracking in the presence of long-term occlusions. *Comput. Vis. Image Underst.*, Vol. 114, No. 7, pp. 835–846, July 2010.

[14] Nikolaos Kyriazis Iason Oikonomidis and Antonis Argyros. Efficient model-based 3d tracking of hand articulations using kinect. In *Proceedings of the British Machine Vision Conference*, pp. 101.1–101.11. BMVA Press, 2011. <http://dx.doi.org/10.5244/C.25.101>.