

# VISTouch：複数の携帯端末間における動的な立体連携

安本 匡佑<sup>1,a)</sup> 寺岡 文博<sup>2,b)</sup>

受付日 2017年4月22日, 採録日 2017年11月7日

**概要：**本研究では、複数の携帯端末を任意に組み合わせて同時に使用し、それらの動的かつ立体的な連携を簡易に実現するシステム（VISTouch）を構築した。ここでは、容易にデバイスの位置を移動させることができる携帯端末の特性に着目し、これらを立体的に接触させることで3次元空間の操作を可能にした。たとえば、水平に携帯端末を配置するとそこには空間を真上から見た映像が表示される。そのディスプレイ上に別の携帯端末を接触させると、両者の相対的位置関係を動的に取得し、接触した携帯端末にはその位置、方向、角度から見える立体的な映像がリアルタイムに表示される。評価実験では、4種類のケースについて接触認識率を調べ、VISTouchを実用的に利用するために必要な仕様を確認した。

**キーワード：**インタフェース、情報視覚化、マルチデバイス、立体ディスプレイ

## VISTouch: Dynamic 3D Connection between Multiple Mobile Devices

MASASUKE YASUMOTO<sup>1,a)</sup> TAKEHIRO TERAOKA<sup>2,b)</sup>

Received: April 22, 2017, Accepted: November 7, 2017

**Abstract:** We constructed and enhanced a novel system called “VISTouch” that enables multiple devices to be used in combination dynamically and spatially. We developed a mobile device feature that would enable users to easily move the position of a device connected to another device and keep track of it. When a mobile device is placed horizontally, it displays an image that can be viewed from above. When another mobile device is placed on this display, the relative positional relation between them is dynamically acquired. Then, the device displays the image viewed from the contact position, direction, and angle in real time. This system is evaluated by calculating the contact recognition rate for the four cases; as a result, we confirmed the specifications required to use VISTouch practically.

**Keywords:** interface, visualization, multiple devices, 3D display

### 1. はじめに

近年、ゲームをはじめとするエンタテインメント分野では、立体情報や3DCGがあたり前のように利用されている。それ以外の分野に関しても、情報の視覚化において立体情報を扱う機会は多々存在している。たとえば、CTやMRIを用いた医療情報や機械の動作シミュレーション、建築物のCG、地図情報など様々な分野における用途があげ

られる。しかしながら、これらの情報に関する提示手段を考えると、2Dの文字情報を扱い始めて何十年も経過しているにもかかわらず、基本的に四角い平面のディスプレイに依存しているといえる。そのため、情報としては本来2次元しか提示できないため、現実の3次元空間における人間の認識とは異なる。従来の提示手段では、人間は仮想の3次元空間上での移動や回転をその映像の変化からしか判断することができず、実空間で利用している頭部・身体の傾きや眼球運動などから得られる身体感覚情報を用いることができない。それゆえ仮想的に3Dを2Dのディスプレイ上で表現するためには複雑な操作が求められ、ユーザにとって理解しやすいとはいえない。

現在、VRやMRなどの分野ではヘッドトラッキングが可能なHMDが一般的になってきたが、NVIDIAのGeForce

<sup>1</sup> 神奈川工科大学  
Kanagawa Institute of Technology, Atsugi, Kanagawa 243-0292, Japan

<sup>2</sup> 東京工科大学  
Tokyo University of Technology, Hachioji, Tokyo 192-0982, Japan

a) yasumoto@ic.kanagawa-it.ac.jp

b) teraokatkh@stf.teu.ac.jp



図 1 VISTouch を使用した様子

Fig. 1 Depiction of VISTouch system in use.

GTX 960 以上の高速な GPU を備えた PC を必要とすることが多い。また、HMD は頭に装着する必要があるため、重量があるため頭部に負荷がかかることに加え、大勢で使用することも多いため不衛生な面がある。そのほかにも、マルチプロジェクションやマルチディスプレイを使用する CAVE システムのようなものもあるが、高価で設備が大がかりになり可搬性に欠ける。そして、どれもスマートフォンやタブレットなどの携帯端末と比較すると普及していない。

このような背景を受け、本研究では一般的に普及している携帯端末を複数使用し、それらを安価で容易な方法で連携させることで 3 次元空間の視覚化と操作を可能にするシステム (VISTouch) を構築した。VISTouch は、携帯端末どうしの相対的位置関係を動的かつ立体的に取得することで、図 1 のように不定形な立体ディスプレイを実現し、従来とは異なるアプローチで上述した問題の解決を図っている。

以下に本稿の構成を記す。2 章では、本システムの稼働に必要なタッチケースについて述べる。3 章と 4 章では、本システムの詳細ならびに応用について述べる。5 章で評価実験について述べ、6 章で立体情報の視覚化に関する関連研究について概説したうえで提案手法の立ち位置を確認する。そして最後に、7 章でまとめを述べる。

## 2. VISTouch ケース

VISTouch が端末以外に必要なハードウェアは、導電体突起物を有するタッチケースのみである。以下では、ケースの仕組みならびに接触パターンについて説明する。

### 2.1 タッチケースの仕組み

携帯端末で広く用いられているタッチパネルは静電容量方式であり、人間の指のように導電性のもので接触することで反応する。さらに、複数の接触点を導電体で接続することで手指が触れなくても静電容量方式のマルチタッチセンサを反応させることが可能である [10]。VISTouch の

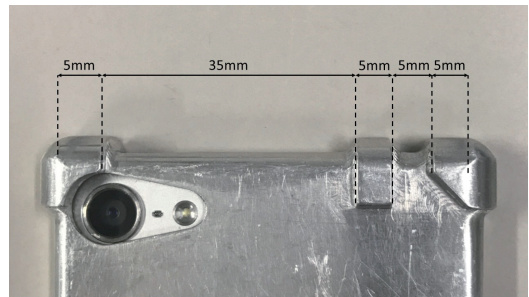


図 2 突起物の寸法

Fig. 2 Dimensions of protuberances.



図 3 iPod Touch に装着したタッチケース (type A)

Fig. 3 VISTouch case type A for iPod Touch.

タッチケースは、これらの性質に基づいて設計されており、指の役割をする複数の導電素材の突起物を有し、それらを導電素材で接続している。そのため、ユーザが触れていなくてもタッチケースを装着した接触デバイス (e.g., iPod Touch) が被接触デバイス (e.g., iPad) のディスプレイに接触していれば反応する。突起物は曲面になっており、どのような角度で接触しても認識できるようになっている。

中島ら [10] によれば多くのマルチタッチデバイスでは 5mm 幅の電極が内部で使用されていることから、タッチケースの導電体突起物の幅は 5mm 以上を確保している (図 2)。開発当初 [17], [18] は、3D プリンタで出力した紫外線硬化樹脂に導電塗料を塗布したタイプ (type A) (図 3) だったが、導電塗料の剥離、熱による樹脂の変形、これらに起因する接触不良が問題になっていた。これらに対応するため、現在はマシニングセンタでアルミ合金 (a6061) を切削したタイプ (type B) (図 4) を使用し、高剛性かつ高精度な接触の認識を実現している。

図 4 のように、タッチケース type B は 1 辺に 3 点の突起物があるタイプと 4 点の突起物があるタイプがある。接触デバイスと被接触デバイスを 1 対 1 で連携させるときは前者のタイプを使用するが、同一寸法のスマートフォンを複数連携させるときは後者も使用する。これは、タッチケースの寸法や突起物の幅によっては、接触点間の距離や比率の組合せが近似されて誤認識が発生するからである。

さらにタッチケース type B には、面接触用の突起を背面に加えている (図 5)。開発当初は、1 辺 3 点のタッチケースを使用した場合、辺接触と面接触で接触面を共有してい



図 4 アルミ合金のタッチケース (type B) (左が 1 辺 4 点のタイプ, 右が 1 辺 3 点のタイプ)

Fig. 4 VISTouch case type B (left: 4-protuberance type, right: 3-protuberance type).

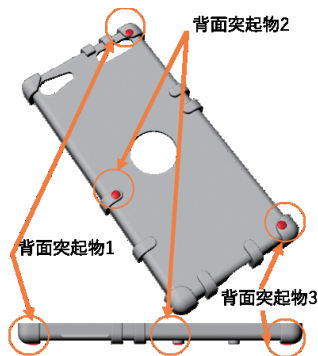


図 5 背面に加えた面接触用の突起物

Fig. 5 Protuberances for surface contact added to back side.

たため、面接触した際に最大 8 点が接触することになり、判別が煩雑になっていた。そこで、タッチケースの背面に突起を加え、通常の導電体突起物よりも出っ張らせて面接触の接触点を 3 点に減らすことで判別処理を単純化し、複数デバイスの接触を可能にした。

## 2.2 接触パターン

タッチケースが有する突起物の配置の関係上、被接触デバイスとの接触パターンは図 6 のように 5 種類ある。P1 から P4 の 4 種類は辺接触で、P5 は面接触の場合である。P1 と P2, P3 と P4 は接触する辺の物理的な長さが異なるため、それぞれを判別できる。P1 と P2 は、同じ長さの短辺で接触しているため物理的距離は同じだが、接触点間の距離の比率が異なるため、P1 と P2 を判別できる。P3 と P4 に関しても同様である。これらの比率は、接触デバイスの側面部にあるスイッチやコネクタなどを避けて、辺ごとに異なる。そして突起物間の物理距離もすべて異なるようにすることで、辺の判別精度を上げている。

VISTouch では、この物理距離を被接触デバイスのピクセル密度と点間のピクセル数から計算している。たとえば、1 辺 3 点の突起物がある iPod Touch の type B ケースの場合では、突起物間の距離は上辺が 35 mm と 5 mm (図 2)、もう一方の短辺である下辺は 28 mm と 12 mm となっている。突起物中央で認識すると仮定した場合、突起物の幅が

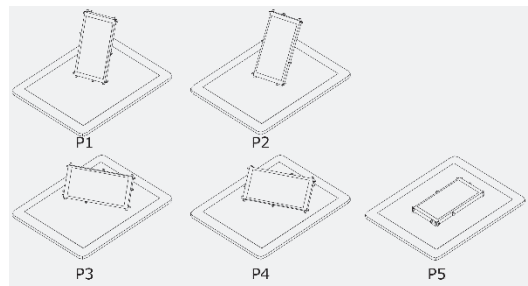


図 6 5 種類の接触パターン. P1 から P4 は辺接触. P5 は面接触. P1P2 と P3P4 は長さが異なり, P1 と P2 (P3 と P4) は比率が異なる

Fig. 6 Five connection patterns (P1–P2 and P3–P4 have different distances between connection points; all patterns have different rates between connection points).

5 mm のため被接触デバイスが認識する突起物間の距離は上辺が 40 mm と 10 mm (i.e., 比率は 4:1), 下辺が 33 mm と 17 mm (i.e., 比率は 33:17) となる。被接触デバイスが iPad Pro 12.9 インチのとき、ピクセル密度は 264 ppi のため、上辺が 416 ピクセルと 104 ピクセル、下辺が 343 ピクセルと 177 ピクセルとなる。使用するデバイスの数や形状にもよるが、5%程度の誤差を許容し、その範囲内であればその辺での接触であると判定する。

接触パターンを正確に判別するには、被接触デバイスと接触する際にタッチケースの突起物 3 点ないしは 4 点が安定した平面を形成する必要がある。従来、このような接点に関する接続機構として、導電ゴムのような柔軟性のあるものを利用することで接触の安定を確保する、あるいは Gorbet らのようにスチールウール製の接点を使用して直線上に配置された接点の接続を安定させる [4] など、様々な工夫が提案されている。本研究では、マシニングセンタに Roland MDX-5000R<sup>\*1</sup>, 被切削材にアルミ合金 (a6061) を使用することで、タッチケースの工作精度の高精度化と高剛性化を実現し、突起物による安定した接続を図っている。このマシニングセンタとコーナーカッタを組み合わせ加工することで、接触面の曲面は十分に滑らかであり、その周囲も極小のエンドミルでフィレット加工を施している。そのため、Gorbet らのスチールウール製接点よりも高精度でありながら表面を傷つける可能性が低い。また、ディスクスタイラスペンのような可変機構を取り入れて接触面の導電性ディスクをつねにディスプレイと平行にする機構も考えられる。しかし、可動部が存在すると故障のリスクが高くなり、制作コストがかさむ可能性がある。これらの理由から、本研究では同種の安定化機構を取り入れず、工作精度の高精度化とケースの高剛性化を実現させることで安定した静電容量方式の接触認識を可能にしている。

さらにその他の工夫として、図 7 のタッチケース (type

<sup>\*1</sup> 機械分解能とソフトウェア分解能がともに 0.001 mm の精度かつ加工精度が 0.01 mm である。





図 7 タッチケース (type C) (左) と type C を装着した Android 端末 (右)

Fig. 7 VISTouch case type C (left), VISTouch case for Android device (right).

C) のように 1 辺だけの接触に用途を絞り込むことで, type B よりも認識精度の向上と処理の高速化を実現している. この type C は 1 辺のみの構造だが, 各突起物が導電体で接続されているため, 直接ケースに手指が触れなくても認識可能である. 以上のことから, type B と type C のタッチケースを使い分けることで, 3 台以上の連携や 5 種類の接触パターンを利用した幅広い応用を目的とする場合と, 用途を限定する代わりに質の高い応用を目的とする場合の両方に対応することが可能である.

### 3. VISTouch システム

VISTouch は, 複数の携帯端末における 3 軸 (X 軸, Y 軸, Z 軸) の位置情報と角度情報を含んだ相対的位置関係を動的に取得する. 以下では, システムの詳細を述べる.

#### 3.1 構成

VISTouch のシステムは, 起動時に各デバイスが装着しているケースのタイプを登録し, 他のデバイスとの連携準備を行う. デバイスは, 1 台のサーバデバイスとそれ以外多数のフロートデバイスがある. これらは起動時に選択でき, デバイスのハードウェア固有のものではない. ネットワークはクライアントサーバモデルであり, サーバデバイスはサーバ, フロートデバイスはクライアントとなる. どのデバイスも被接触デバイスあるいは接触デバイスになることが可能であり, 大元の親デバイスがサーバデバイスである必要はない. 処理の簡略化のため大元の親デバイスのみを固定し, それ以外は子や孫などの子孫になりうることにする.

デバイスの親子関係は, デバイスどうしの接触時に決定する. 接触されたデバイス (被接触デバイス) が相対的な親となり, 接触したデバイス (接触デバイス) が相対的な子となる\*2. ただし, 被接触デバイスがすでに他のデバイスと接触していた場合は, その他のデバイスが相対的に親となり, 被接触デバイスが子, 接触デバイスが孫という関係になる. 相対的な親デバイスは相対的な子を複数持つこ

\*2 大元の親と相対的な親は必ずしも一致しない.

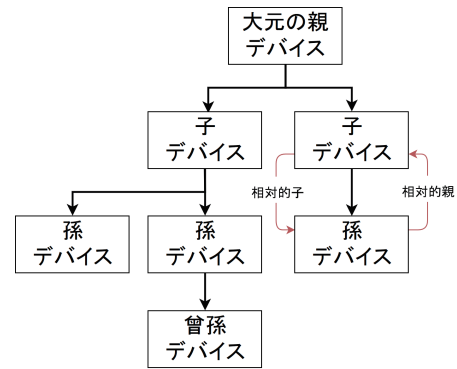


図 8 連結されたデバイス群の関係

Fig. 8 Relations of connected devices.

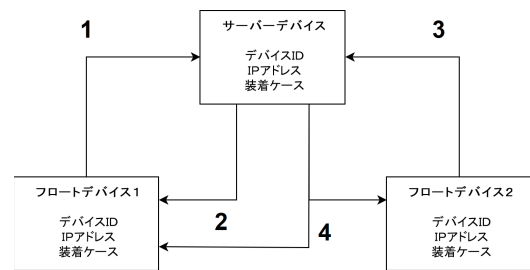


図 9 起動時のフロー

Fig. 9 Startup process flow.

とが可能だが, 相対的な子デバイスは相対的な親を複数持つことはできない. デバイス群の相対的な親は, 大元の親デバイスと必ずしも一致しないため, この大元の親デバイスに一度も接触していないと大元の親に対する位置関係が不明になる. そのため連結されたデバイス群は, 大元の親の子孫になるように接触させて使用する (図 8).

#### 3.2 起動処理

システム起動時は, 最初に各デバイスのタッチケースの情報を登録し, それぞれの役割を各デバイスに伝える必要がある. たとえば, 1 台のサーバデバイスと複数のフロートデバイスがある場合 (図 9), まずサーバデバイスを起動し, このデバイスの IP アドレスを既知のものとする. そして, フロートデバイス 1 を起動させ, 装着しているケース (装着ケース) の情報をサーバデバイスに無線で接続して送信する. サーバデバイスはデバイス ID を割り振り, すべてのフロートデバイスに対してサーバデバイスの装着ケース, デバイス ID, フロートデバイス 1 の装着ケース, デバイス ID などを送信する. フロートデバイス 2 以下を起動する際も同様の処理が行われる.

次に, 各デバイスのピクセル密度 (ppi) の値を端末から取得する. 各タッチケースの突起物の配置, 物理距離 (ミリメートル) は既知の情報であるため, これらの情報に基づいて接触時の座標値から突起物間の物理距離を取得する.

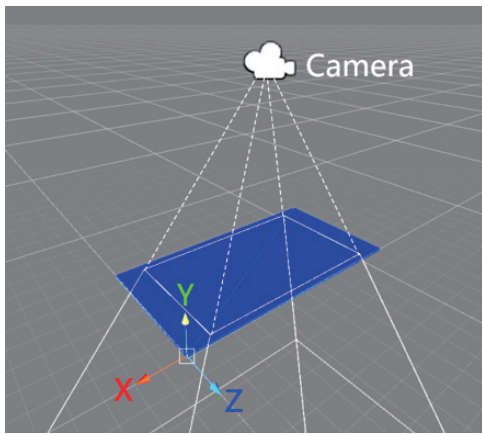


図 10 モデル化された iPod Touch  
Fig. 10 Modeled iPod Touch.

### 3.3 モデル化

タッチケースとそれに対応した端末の形状、接触点の曲面の径、ディスプレイの位置などはあらかじめモデル化されている。たとえば、図 10 の青色で描画された部分は、iPodTouch (タッチケース装着) を表している。ローカル原点は、2 辺の突起物曲面を構成する円弧の中心軸の交点であり、左上か右下のどちらかをデバイスの接触時に決定する。面で接触、左辺あるいは上辺で接触した場合は左上が、それ以外の場合は右下がローカル原点となる\*3。

映像のレンダリングは、ディスプレイの法線方向約 10cm 上部に配置したカメラからディスプレイ方向に対して行う。その際、接触デバイス (ここでは iPod Touch) の画面がちょうど収まるように視野角を 30 度前後に調整している。ただし、カメラとディスプレイの距離は調整可能であり、視野角はその距離に合わせて自動的に変化する。このようにモデル化された情報は、システムが実装されている Unity 上に登録されている。辺接触した際は接触軸の位置情報をもとに計算が行われるため、コンテンツの立体情報を、複数の連結したデバイスで正確に表示することが可能である。

デバイスの角度に関しては、接触の有無にかかわらず、自己の慣性センサからつねに取得し続ける。ただし、VISTouch では地磁気センサを使用しないため、重力方向を軸とする角度は不明である。被接触デバイスは先祖に大元の親デバイスを持つため、被接触デバイスまでの相対的位置関係はすでに決定済みである (図 11)。接触デバイスも被接触デバイスに対する相対的位置情報を取得しており、接触軸と、被接触デバイスのローカル座標系における位置は既知である。さらに、接触デバイスの加速度センサから、接触軸に対するグローバル座標系における回転角度が計算できる。これらの情報を Unity 上で構築した親子関係を取り入れたモデルにあてはめることで、被接触デバイスの傾きにかかわらず、被接触デバイスのローカル座標系にお

\*3 図 10 のローカル原点は右下の原点である。

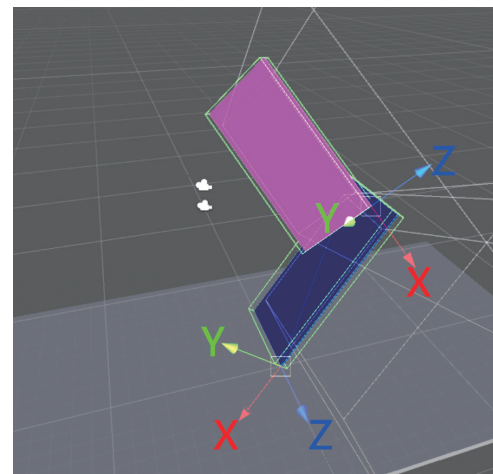


図 11 各デバイスのモデル。ローカル座標系 3 軸が集中している 2 カ所がそれぞれのローカル座標原点を指す (右上は紫色のデバイスモデルのローカル座標軸と原点、左下が青いデバイスモデルのローカル座標軸と原点、グレーのデバイスが大元の親デバイス)

Fig. 11 Model of each float device. Two points where the local coordinate system 3 axes are placed indicate each local coordinate origin (Upper is the local coordinate axis of purple device model, and lower left is one of blue devices. Gray device is original parent device).

る接触デバイスの回転角度を一意に決定する。

### 3.4 接触処理

図 12\*4のように、デバイス B がデバイス A に対して接触すると、デバイス A は接触された複数の座標をマルチタッチ認識によって取得する。Microsoft の Surface Hub などの高価なものを除き、一般的な携帯端末のマルチタッチ認識は 10 点までしかサポートしていない。このような理由から、VISTouch は最大 10 点までサポートしており、1 台のデバイスに接触できるのは最大で 3 台までとなる。そのため、認識パターンは次の (1)~(5) となる。なお、タッチケース側面の突起物は 3 点以上あるため、指などによる 1 点あるいは 2 点の接触は UI 操作として認識だけでなく、ピンチ操作や 2 本指ドラッグなどのタッチパネルで多用されるジェスチャの認識も可能である。

- (1) 2 点以下の場合は「指によるタッチ操作」
- (2) 3 点あるいは 4 点の場合は「デバイス 1 台による辺接触あるいは面接触」
- (3) 5 点の場合は存在しないため「接触不良によるエラー」
- (4) 6 点, 7 点, 8 点の場合は「デバイス 2 台による辺接触と面接触の組合せ」
  - 辺 (3 点) + 辺 (3 点)
  - 辺 (3 点) + 面 (3 点)
  - 面 (3 点) + 面 (3 点)

\*4 メインフローの接触処理についてのみのシステムフローであり、全体の処理を描画しているわけではない。

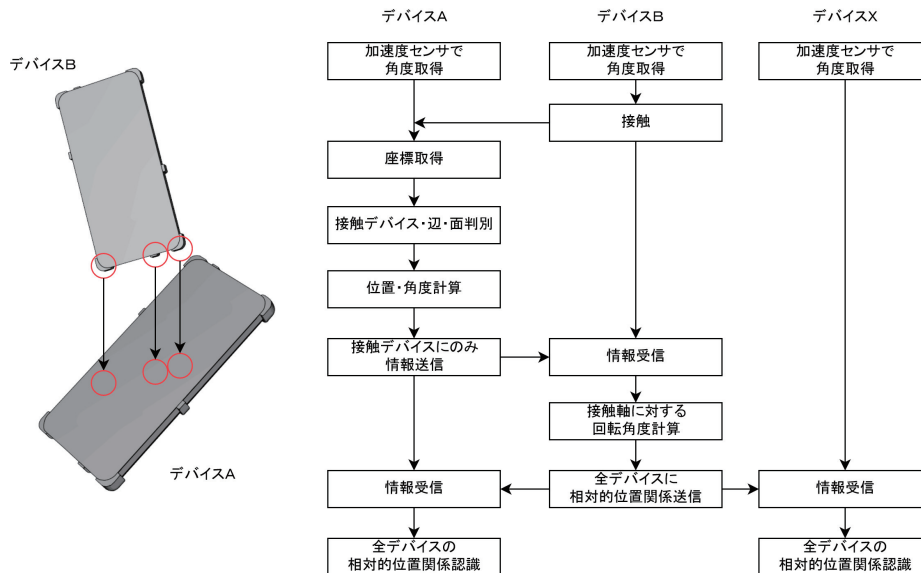


図 12 接触処理の流れ

Fig. 12 Contact process flow.

- 辺 (4点) + 辺 (3点)
  - 辺 (4点) + 面 (3点)
  - 辺 (4点) + 辺 (4点)
- (5) 9点, 10点の場合は「デバイス3台による辺接触あるいは面接触の組合せ」

- 辺 (3点) + 辺 (3点) + 辺 (3点)
- 辺 (3点) + 辺 (3点) + 面 (3点)
- 辺 (3点) + 面 (3点) + 面 (3点)
- 面 (3点) + 面 (3点) + 面 (3点)
- 辺 (4点) + 辺 (3点) + 辺 (3点)
- 辺 (4点) + 辺 (3点) + 面 (3点)
- 辺 (4点) + 面 (3点) + 面 (3点)

デバイス3台の同時接触の場合、1辺3点のケースが1台と1辺4点のケースが2台のときは11点、1辺4点のケースが3台のときは12点となるが、検出できないため使用側が混乱する可能性がある。たとえば、8点の場合は4辺×2をサポートしているが、もう1台追加すると認識できなくなる。1辺が4点でも面の場合は接触点が3点のため、この場合は3台まで接続できる。ゆえに、原理上は可能だが、組合せによって認識可否が分かれるため、ユーザビリティの観点から1台への同時接続は2台までに制限するなど運用上の工夫が求められる。

### 3.4.1 単一デバイスでの接触

タッチケースが辺接触している場合は、突起物3点あるいは4点は必ず一直線上に並ぶため、1点から他の点までのベクトルがなす角を計算する。これらが十分に小さいときは一直線上に並んでいると判断し、直線上の点間の距離から、接触している辺を判別する。

面接触の場合は、背面の突起物3点で構成される三角形の形状をもとに判別を行う。被接触デバイスが、他デバイ

スの接触で得られた3つの座標値において各点間の距離(座標のピクセル値とppiから計算)と、それらがなす角を計算する。三角形の形状はタッチケースごとに異なるため、Unity上に登録されているタッチケースの情報と比較し、差分が十分小さいときは合同と判断する。そして、この三角形の位置と方向から接触デバイスの位置と角度を計算する。面接触の際は、被接触デバイスと接触デバイスは平行のため、接触デバイスの加速度センサから得られた角度情報を使うことはなく、この時点で接触デバイスの被接触デバイスに対する相対的位置関係が確定する。

以上の判定から、パターン(2)による接触を判別する。判定できない場合は、接触不良あるいはデバイス以外(指など)が接触している可能性があるため、エラーとして判別処理を終了する。

### 3.4.2 複数デバイスでの接触

複数デバイスによる接触の場合は、どの点がどのデバイスの接触によるものを判別するプロセスが追加されるが、それ以外は単一デバイスによる接触と同様である。まず、複数の点群を3点か4点のグループに分け、総あたりですべての組合せを試す。点間の距離が、登録されているタッチケースの点間の最大値よりも長いあるいは最小値よりも短い場合は不適切と判断し、次の組合せを試す。

組合せが不適切ではない場合は、単一デバイスによる接触と同様に辺接触と面接触の判別を行い、エラーが出なければ他のグループに対しても同様に判別を行う。複数デバイスの場合は、すでに接触しているタッチケースの別の辺が接触することはない。そのため、2つ目以降の判別には同一タッチケースでの接触有無を判別に加えている。すべてのグループでエラーが生じず、組合せが唯一の場合はこれで確定する。





図 13 携帯端末の直列連携

Fig. 13 VISTouch Multi-Cascade Connection.

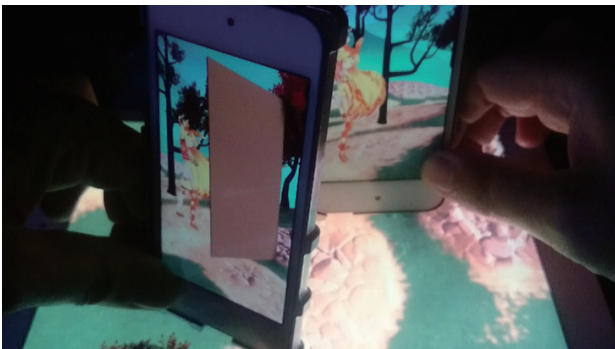


図 14 携帯端末の並列連携

Fig. 14 VISTouch Multi-Parallel Connection.

たとえば、3台のデバイスを連携させる場合、直列(図 13)と並列(図 14)の連携が可能となり、これらは設定をいっさい変えることなく接触の仕方だけで自動的に切り替えられる。前者の直列連携によって2台の連携だけでは確認できない空中での様々な切断面を視覚化でき、後者の並列連携によって複数人で同時に使用できる。このように、3台以上の連携方式を使い分けることにより、VISTouchの応用に関して幅広い用途が期待できる。

### 3.5 デバイス間の通信処理

各デバイスは、他のデバイスと無線で通信を行っている。屋内だけでなく屋外での使用も想定し、開発初期はBluetoothを使用していた。しかし、iOSと他のOSの通信に制限があったため、現在ではWi-Fiを使用している。多くのデバイスにテザリング機能があるため、無線の親機がなくても屋外での使用が可能である。主な通信は次の3種類である。

#### 通信 1：被接触デバイスから接触デバイスへの個別通信

被接触デバイスが接触デバイスとどの辺あるいは面での接触なのかを特定した後に、接触デバイスに以下の情報を送信する。

- 被接触デバイスのローカル座標系での3軸の位置情報
- 被接触デバイスのディスプレイの法線方向の回転角度
- 接触デバイスのどの辺あるいは面で接触しているのか

- 被接触デバイスの加速度センサから取得した角度情報
- 被接触デバイスの立場(大元の親、子や孫など)

#### 通信 2：全デバイスへの全体通信

相対的位置関係が確定した際に、接触デバイスが全デバイスに送信する。これは自己の位置・角度情報を全デバイスに登録することを目的とし、主に以下の情報が含まれる。

- 被接触デバイスがどのデバイスなのか
- 被接触デバイスのローカル座標系における自己のデバイスの3軸の位置・角度情報

#### 通信 3：その他の汎用通信

VISTouchにおける相対的位置関係以外に関する通信。主に以下が該当し、その他の情報も任意に追加することが可能である。

- 起動時に自己のデバイスIDと装着ケースを全デバイスに知らせる
- シーン遷移(e.g., いっせいにタイトル画面に戻る)
- UI操作
- 原点の位置・角度情報の更新
- 共通オブジェクトの位置・角度情報(e.g., 「Phantom Window」のボール)

前プロセスで被接触デバイスが接触デバイスを特定した後、接触デバイスに対して通信1を行う。接触デバイスは、この通信を受信して初めて他のデバイスとの接触を認識する。接触デバイスは、被接触デバイスに接触している自分の辺を認識しているため、加速度センサからその軸に対する回転角度を取得する。これにより、被接触デバイスに対する接触デバイスの3軸(X軸、Y軸、Z軸)の位置、角度が確定する。Unity上でモデル化されたデータにこれらの値を入れることで、ディスプレイの正確な位置ならびにその法線方向上に存在するレンダリング用のカメラの位置が確定する。なお、接触が行われていない場合は位置情報は確定していないため、前回取得した位置情報は維持されながら、加速度センサから取得した角度情報のみ更新する。

次に、全デバイスに対して通信2を行う。被接触デバイスとそのローカル座標系での位置・角度情報を送信し、全デバイスでそれらの情報を更新する。これにより、接触デバイスの大元の親デバイスに対する相対的位置関係を、すべてのデバイスが取得した状態となる。この段階で被接触デバイスがすでに他のデバイスに接触していたとしても、大元の親デバイスから被接触デバイスまでの判別処理は済んでおり、位置や角度は相対的な親のローカル座標系における値のため、全デバイスの相対的位置関係が明らかになっている。



図 15 Google Maps と Google Street View の連携

Fig. 15 Cooperation of Google Maps and Street View.

## 4. VISTouch の応用

VISTouch の特長は、インタラクティブな立体ディスプレイであり、その形状や数を用途に応じて変更が可能な点にある。以下では、この特長を活かした立体情報の視覚化とゲームへの応用事例をあげ、その効果について言及する。

### 4.1 立体情報の視覚化

従来、3D 空間を 2D のディスプレイで操作すると、移動や方向に関する情報は映像情報からしか判断できなかったため、ユーザによっては方向を見失う傾向があった。たとえば、Google Street View を使用するとき、2D のマップと 3D の Street View が別の位置に描画されるため、Street View 側でしばらく操作を続けると地図上のどの位置でどちらの方角を見ているのか判断が難しくなることがあげられる。

図 15 は、VISTouch で既存サービスの Google Maps と Google Street View を組み合わせて使用した様子である。処理の流れを概説すると、まず携帯端末を水平に置き、インターネット上の API を利用してリアルタイムに Google Maps を表示する。このディスプレイ上に、タッチケースを装着した別の携帯端末が接触すると、接触点の位置座標（右端の突起物の位置）から Google Maps 上での緯度と経度を取得する。緯度、経度、被接触デバイスのローカル座標系での角度などの情報を接触デバイスに無線で送信し、接触デバイス側でインターネット上の Google Street View API より緯度と経度に対応する全天周映像を取得して、仮想的に接触デバイスの周囲に配置する。そして、接触デバイスの角度情報より、Google Maps 上の接触点から接触デバイスのディスプレイを向けた方向の映像をリアルタイムに取得して表示する。ユーザは、被接触デバイス側でドラッグやピンチ操作を行いながら Google Maps で任意の



図 16 頭部 MRI の立体的視覚化

Fig. 16 Visualizing Medical Information.

場所に移動し、接地した接触デバイスの画面の向きに対応した Google Street View の映像を見ることができる。それぞれのデバイスの物理的な配置を動かすことで移動や回転も可能なため、従来の地図アプリの問題を解消し、ユーザは地図情報を容易に理解することが可能になる。

図 16 は、CT や MRI などの平面連続画像をポイントクラウドに変換して立体化した情報を VISTouch で視覚化した様子である。なお、被接触デバイス側の UI で基準の高さを調整することが可能となっている。ディスプレイ上のスライダーを操作することで、その高さに応じた断面が接触デバイスで表示される。これにより、撮影時に保存された水平の断面だけでなく、任意の位置と角度における断面をリアルタイムに表示させ、立体化した医療情報を俯瞰することができる。

### 4.2 ゲームへの応用

ゲームに応用した場合、プレイヤーは仮想空間の視点を変えながら探索し、オブジェクトを触って操作することが可能である。たとえば、図 17 のように接触デバイスに表示されているボールをドラッグすると、その方向にボールを弾くことができる。この「Phantom Window」の本質は、複数のデバイスを組み合わせることで実現するインタラクションにある。接触連携した複数デバイスは映像の位置や角度がぴったりと合い、現実空間で見えているそれらの物理的位置関係がそのままゲームの中に反映されている。たとえば、被接触デバイスを水平に置きそのディスプレイ上で接触デバイスを傾けると、接触デバイス上のボールがその傾きに応じて転がる。接触デバイスのディスプレイの下までボールが転がると、そこにつながっている被接触デバイスのディスプレイの同じ場所からボールが転がってくる。つまり、これら 2 つのデバイスにおける物理的位置関係が、ディスプレイ間を転がるボールの軌跡に反映されている。また、接触デバイス上でボールを指で弾くと、ボー





図 17 VISTouch を用いたボールゲーム「Phantom Window」  
Fig. 17 Ball game for VISTouch “Phantom Window”.

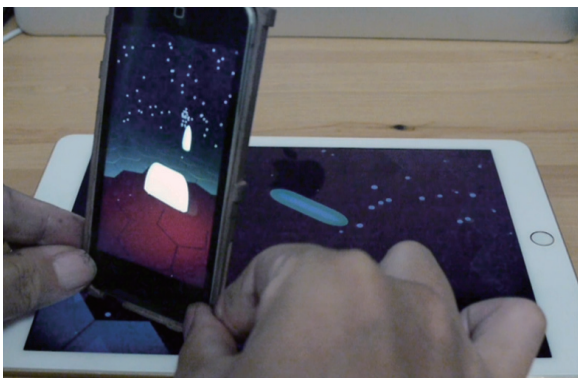


図 18 アクション性の高いゲーム「Shadow Dancer」  
Fig. 18 Action game for VISTouch “Shadow Dancer”.

ルはそのデバイスの角度で弾いた方向に飛んでいく。この場合は、接触デバイスがジャンプ台のように機能している。この位置や角度は VISTouch によるものであり、それを正確に反映したインタラクションをプレイは楽しむことができる。

図 18 の「Shadow Dancer」は、接触デバイス自体をゲームのプレイヤーが操作するキャラクターと見立てたゲームである。接触デバイスを動かすことで敵の攻撃を避けたり、接触デバイスを傾けて攻撃を掻い潜ったりすることができるアクション性の高いゲームとなっている。3 台以上にも対応しており、より大きなデバイスを被接触デバイスに用いれば複数人で同時に対戦したり、協力して遊んだりすることが可能である。なお、被接触デバイスのディスプレイには、プレイヤーが放つ銃弾や、それがあたってときのダメージエフェクト、接触デバイスでは見えない別方向からの攻撃などが表示されるため、プレイヤーの周りにいるオーディエンスにも具体的なプレイの様子が分かる。

### 4.3 応用の効果

本稿では、被接触デバイスと接触デバイスとして iPad と iPod Touch をそれぞれ用いて例示しているため、VISTouch が 1 人用向けと判断されてしまうきらいがあるが、実際には多くの人が同時に自由な視点で体験を共有しながら情報にアクセスすることが想定されている。誰が何をどこからどういった角度で見ているのかが、他者から理解される必要があり、そのためには「ミリ単位の移動計測」と「相対的位置関係の視覚化」が重要となる。

デバイス単体で外部にセンサなどを設置しなくてもミリ単位の移動を計測できるという意味においては、VISTouch 以外には Google Tango [7] などが該当する。Google Tango は単一デバイスで動作し、慣性センサと深度映像で、空中でのポジショナルトラッキングを実現している。しかし、継続使用するとセンサは徐々にずれていき、さらに人混みなどの変化の激しい環境では映像もずれていく傾向がある。そのため、時間が経過した後に物理的にまったく同じ場所に戻ったとしても、完全に 1mm もずれていないということはない。一方、VISTouch は空中での操作に制限があるが、接触さえしていればポジショナルトラッキングが可能である。デバイスどうしの接触で認識するため、どれほど時間が経過しても物理的にまったく同じ位置に戻ったのであれば、寸分の違いなく同位置・同角度で認識が可能である。

相対的位置関係の視覚化は、デバイス进行操作しているユーザだけでなく、周囲にいる他者もその人がどこからどういう角度で何を見ているのかが分かる手がかりになる。物理的に見えているその角度と同じ角度の映像がディスプレイに映し出されることも加えると、VISTouch を用いたデバイス操作は、ゲームコントローラのアナログスティックやマウスのような相対的な操作ではなく、まったくの直接的な操作といえる。ここでいう相対的な操作とは、マウスを動かすとその方向と移動量だけカーソルが移動するようなものを指し、これは絶対座標と角度で操作しているわけではない。一方で、スマートフォンやタブレットなどのタッチ端末は絶対座標での直接的な操作に該当する<sup>\*5</sup>。一部のペンタブレットなどであればペンの傾きも認識が可能だが、指による接触では位置のみとなる。一方、VISTouch は位置と角度の絶対指定に基づいたより高度な直接的な操作を実現している。ゆえに、位置と角度の相対的位置関係が見えていることも含めて、直感的な操作であるといえる。被接触デバイスと接触デバイスに映し出されている各映像や、両デバイスの現実の相対的位置関係を指標とすることで、ユーザ個人は操作の微調整が容易となり、周りの他者はその具体的な様子を理解することができる。

以上より、VISTouch を用いることで精密な移動計測を

<sup>\*5</sup> 厳密にはタッチ端末のローカル座標系における絶対座標指定である。

実現するだけでなく、相対的位置関係が見えるため直感的な操作が可能となるといえる。

## 5. 評価実験

VISTouch を利用するには、タッチケースの安定した認識が必要となる。そこで、最新のタッチケース (type C) で形状、特に突起物の幅や曲面の径を変えた 4 種類を用意し、それぞれの認識率について評価を行った。以下では、実験の詳細ならびに評価の結果について述べる。

### 5.1 実験環境

iPhone7 用の 4 種類の type C ケース (図 19) について認識率を調べる。各ケースには突起物が 3 カ所存在し、ケースごとに突起物の仕様 (直径・幅) が異なっている。また、突起物の端には 0.5 mm 追加して、半径 0.5 mm のフィレット加工 (図 20 は直径 15 mm・幅 6 mm のモデル) を施している。被接触デバイスには iPad Pro 12.9 インチ、接触デバイスには iPhone7 を用いる。計測は 3 点とも認識した場合を成功とし、それ以外の場合は失敗として数える。計測時間は 10 秒とし、およそ 300 回前後のうち何回成功したかを計測するソフトウェアを用いて認識率を求める。

### 5.2 実験の手順

VISTouch は、中島らの論文にあるような静電容量方式の中でも、マルチタッチパネルに用いられる相互容量方式

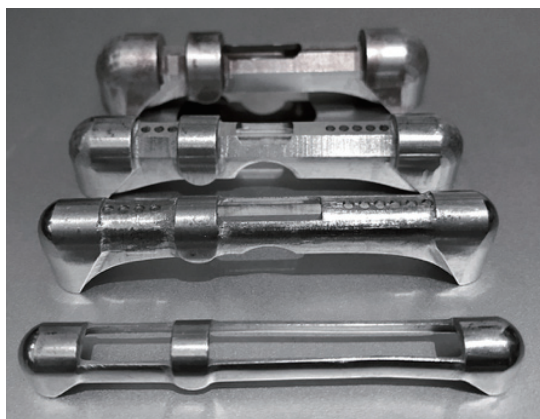


図 19 実験に用いた 4 種類のタッチケース type C

Fig. 19 Four touch cases (type C) used for evaluation experiments.

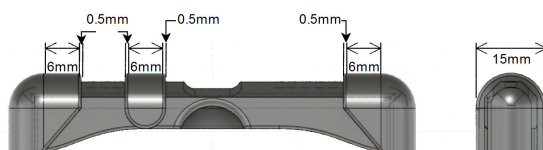


図 20 突起物の仕様が直径 15 mm・幅 6 mm の type C モデル

Fig. 20 Model of type C with 15 mm diameter and 6 mm width of protuberances.

を対象としたブリッジマーカシステムを応用している。これは、ディスプレイ面に対して縦横に平行な短冊状の透明電極が格子状に並べられた構造を内部に持つ。この電極に平行 (0 度あるいは 90 度) になるように突起物を配置した場合は、電極間の静電容量の差が生じにくくなるため認識率が著しく低下し、斜めに配置した場合は認識率が上がることが報告されている [10]。そこで、実験 1 と実験 2 ではディスプレイ面の縦横に対してケースを 0 度あるいは 45 度に配置し、実験 3 ではディスプレイ平面に対して 0 度から 90 度の間で回転させて、次のように認識率を求めた。

まず、各ケースを単体で被接触デバイスのディスプレイ上に配置し、10 秒間の認識率を計測した (実験 1)。次に、各ケースを接触デバイスに装着した状態で、実験 1 と同様に認識率を計測した (実験 2)。この際、接触デバイスを手に持ち、ケースには直接手が触れないようにした。そして最後に、ケースを装着した状態で接触デバイスを手で保持し、ケースに直接手が触れない状態で 0 度と 90 度の間を 10 秒間で 10 回程度、回転させながら往復運動させ、認識率を計測した (実験 3)。なお、いずれの実験もそれぞれの試行を 20 回ずつ行った。

### 5.3 結果と考察

表 1 は実験 1 から実験 3 までの結果をまとめたものである。実験 1 では、ケース単体の場合、直径 8.6 mm のケース以外は 0 度 (あるいは 90 度) に配置しなければ人の手が触れてなくても接触を確実に認識した。特に直径が 12 mm と 15 mm の場合は、接触した状態で揺らしても 100% の認識率があった。しかしながら、0 度の場合にはどのケースもまったく反応しなかった。

実験 2 では、ケースをデバイスに装着した場合、直径 8.6 mm 以外のケースは 0 度であっても 100% 認識した。ケースに手が直接触れていなくても接触デバイスがケースと密着しているため、間接的に手が被接触デバイスに接触していると認識されたためである。なお、8.6 mm のケースは、この実験のときもまったく認識しなかった。このケースは軽量化を優先して製作したため、突起物や径が小さいだけでなく、それぞれをつなぐフレーム部分が薄かったことが主な原因としてあげられる。ゆえに全体的な剛性が低く、歪みによって突起物の 3 点が平面を形成していなかったため、接触が不十分だったことが考えられる。

実験 3 では、突起物の径が大きくなるほど認識率が上がっていた。直径 15 mm のケースは、20 試行中 13 試行は 100% 認識しており、認識率の平均は 99.5% であった。手動による試行のため、動かしている間にデバイスが浮いてしまい認識できない場合があったが、ほぼ 100% 認識できているといえる。逆に、8.6 mm のケース (平均 26.3%) は動かしている間に偶然うまく接触した場合のみ認識し、9.8 mm のケース (平均 75.5%) と比較しても極端に認識率は低かつ



表 1 評価の結果

Table 1 Results of evaluations.

突起物の仕様 (直径・幅)	実験 1 (ケース単体)		実験 2 (ケース装着)		実験 3 (ケース装着)
	0 度	45 度	0 度	45 度	回転
8.6 mm · 4 mm	0.0%	0.0%	0.0%	0.0%	26.3%
9.8 mm · 5 mm	0.0%	100.0%	100.0%	100.0%	75.5%
12 mm · 6 mm	0.0%	100.0%	100.0%	100.0%	90.3%
15 mm · 6 mm	0.0%	100.0%	100.0%	100.0%	99.5%

た. なお, 各ケースの認識率 (全 20 試行) については, 各組合せに対して F 検定で等分散・不等分散を調べた後に両側 t 検定を行った. 多重比較のため Bonferroni 法を用いて検定したところ, どの場合も統計的に有意 ( $p < 0.0083$ ) であることを確認した.

これらの結果から, 突起物の直径と幅がそれぞれ 9.8 mm と 5 mm 以上ならば, 直接手でケースに触れることなく認識することが明らかになった. また, 突起物の幅だけでなく直径も大きくなると認識率が上がることが分かった. 直径・幅が 15 mm · 6 mm のケースは, ほぼ 100% 認識していたが, iPhone 7 本体の厚み (7.1 mm) をふまえると, ケース装着時は厚みが倍以上になり, かさ張ってしまう. 認識が失敗した際は相対的位置関係の更新が行われただけであるため, 認識率 90% 以上の直径・幅が 12 mm · 6 mm のケースであれば, ユーザは実際に使用するうえでほとんど気にはならないといえる.

接触時の認識可否は, 被接触デバイス側のハードウェアやソフトウェアに依存するため, 別のデバイスでは認識率が変わってしまうことが考えられる. ゆえに, 今後は 15 mm のケースと同等の認識率を実現するより小さい径の限界値を調べるとともに, どのデバイスにも対応できるように余裕を持った設計を目指していきたい.

## 6. 関連研究

これまでに提案されてきた立体情報の視覚化手段としては, 複数のプロジェクタを用いる RoomAlive [5], モバイルプロジェクタとセンサなどにより方向を認識し, その方向に映像を提示する HideOut [15] や参式電子弓 [16] がある. また, 複数の携帯端末を組み合わせるマルチデバイス, マルチディスプレイの分野では, 特にゲームにおいて活発で任天堂の 3DS や Wii U がある. しかし, これらは複数のディスプレイの使い分けであり, 各々の位置関係を効果的に利用しているとはいえない. マルチディスプレイの配置に関する研究では, 特徴的なものとして Pinch [12] やその派生の研究 [11], [13] があげられる. これは複数の隣接した iOS デバイスをピンチ操作することで 1 つの画面としてつなげることができるが, 平面的にしか接続できない. また, リアルタイムかつ連続的に相対的位置を変化させることには完全には対応しておらず, 端末間の相対的位置関係

が変わった場合は接続し直さなければならない.

カメラからの映像認識により複数デバイスの連携を行う研究 [1], [2] では, 代表的なものとして THAW インタフェース [6] がある. しかし, 動的な位置関係の変化に対応できるものの, 立体的に角度を変化させることには対応していない. 視点トラッキングすることで限定的に立体配置に対応した研究 [8] もあるが, 本質的に立体情報を扱いきれしておらず, 複数人での使用にも対応できない. 動的な移動と立体的配置を実現する研究も代表的なものがいくつかあげられる [3], [9], [14]. なかでも, Deskrama は光学式モーションキャプチャシステムを使用してディスプレイの位置を検出し, 立位ディスプレイとして映像を映すことができる [9]. また, PaperLens もプロジェクタとカメラを使用し, 床面に平行なディスプレイ面上の空間に別ディスプレイを浮かせ, 立位ディスプレイを可能としている [14]. しかし, これらはカメラやプロジェクタを毎回セットアップしてキャリブレーションを必要とするため, 持ち運びが難しく可搬性が低い. そして, 明るい場所や屋外ではプロジェクタの映像が不鮮明になるため使用環境に制約がある. プロジェクタの解像度が, 同程度の価格の場合は PC ディスプレイ, タブレット, スマートフォンと比べて一般的に低いことを考慮すると, PaperLens の立位ディスプレイでは投影する映像の一部のみを使用するため解像度はさらに低くなる. そのため, 詳細かつ繊細な情報が必要な医療情報や小さな文字などに関しては正確な判別が困難である. また, 上方からプロジェクタで投影することから, たとえば投影光に対して立位ディスプレイ面が並行に近づいた場合, ディスプレイの位置や角度, 色再現性について制約がかかる. さらに, 下に位置するディスプレイは水平にして利用する必要があることに加え, 手に持ったディスプレイに映像を映す場合はプロジェクタで上方から投影するためそれまで投影されていた部分が影になることから, 視覚的な制約も有している.

Google Tango [7] は, スマートフォン単体で空間認識とポジショナルトラッキングを実現している. しかし, 慣性センサと併用している関係で, 端末を急激に動かすとトラッキングロストしてしまい, 復帰までに時間を要する. 移動体の中での使用も難しいため, 環境が制限されている. 空間情報を学習させたとしても, 類似した空間情報 (たと



表 2 VISTouch と関連研究の特徴

Table 2 Characteristics of our VISTouch and related studies.

		VISTouch	Pinch	PaperLens	Deskrama	Google Tango	Apple ARKit
使用環境	暗闇	○	○	○	○	×	×
	移動体	△	○	×	○	×	×
	磁場が乱れた場所	○	○	○	○	×	×
	明るい場所・屋外	○	○	×	×	○	○
	空中	△	×	○	△	○	○
複数対応	複数デバイスを複数人で共有	○	○	-	×	○	×
	単一デバイスを複数人で共有	○	○	○	○	○	○
対応機種	iOS	○	○	×	×	×	○
	Android	○	×	×	×	△ (2機種)	×
	Windows Phone	○	×	×	×	×	×
	PC/Mac/Linux	○	×	○	○	×	×
可搬性	○	○	×	×	○	○	



図 21 接触デバイス（スマートフォン）を空中で使用した様子  
 Fig. 21 Depiction of using a contact device (i.e., smartphone) in air.

例えば、似たような部屋など)があった場合は区別するのが困難である。また、深度センサを備えた Google Tango 対応端末でなければ動作しないため、対応する端末が限定され、広く普及しているとはいえない\*6。また、Apple ARKit は、深度センサ（深度カメラ）ではなく、端末のカメラと慣性センサを組み合わせているため、Google Tango と比べると精度が落ちる。さらに、映像情報として特徴のない状況（たとえば、真っ白で何もない部屋）では正確なトラッキングができない。カメラを利用する関係で、周りに十分な明るさがないと使用できないなどの制約を有している。

本研究の VISTouch は、必要とするハードウェアはタッチケースのみであり、OS や特殊なハードウェア機能に依存しないため、一般的に普及しているスマートフォン端末であればほとんど利用が可能である。解像度と色再現性は端末に依存するが、プロジェクタと比べると一般的に普及しているスマートフォン端末はどれも ppi は非常に高く、高解像度といえる。たとえば、iPhone 7 Plus では 1920 × 1080 の解像度で 326 ppi であり、4k 解像度のスマートフォンでは ppi は 800 を超えている。ディスプレイは暗闇でも屋外でも視認性に問題はない。さらに、外部にセンサやプロジェクタを設置する必要がないため可搬性に優れており、これらの点で Paper Lens よりも有用性が高いといえる。また、ディスプレイが大きな被接触デバイス (e.g., Surface Hub) を用いることで、Paper Lens と同様に広い領域を複

\*6 2017年8月現在で2機種のみ。

数人で共有して見ることが可能であり、複数台を連携させ複数人の各デバイスに別々の情報を映し出すことも共有することも可能であり、スケーラビリティに富む。一方で、VISTouch は、Google Tango や Apple ARKit のように空中でポジショナルトラッキングができない。しかし、タッチケースの代わりに指3本を被接触デバイス (e.g., タブレット) の画面に接触させることで、接触デバイス (e.g., スマートフォン) をもう片方の手に持って空中で使用することが可能である。この場合、角度は接触デバイス自体の情報を用いることで、指に触れた XZ 平面上 (被接触デバイスのローカル座標系) の位置に応じた空間を認識できる (図 21)。被接触デバイスのローカル座標系の XZ 平面から Y 軸方向の上下への移動に関しては、端末どうし (接触デバイスと被接触デバイス) を接触させる必要があるため物理的に2台では不可能であるが、ソフトウェア的に XZ 平面 (被接触デバイスのローカル座標系) の高さを変更しながら使用する、被接触デバイス自体を傾けて使用する、もしくは3台以上のデバイスを用いることで対応が可能である。

以上の点について、VISTouch と関連研究の特徴をまとめると表 2 になる。このように、VISTouch は空中での使用には物理的な制約があるが、対応端末ならびに使用環境に関して汎用性が高く、複数の端末を連携させて3次元空間の視覚化・操作を安価で容易に可能にする技術は他に類を見ない。

## 7. まとめ

VISTouch は、複数の携帯端末を任意に組み合わせて同時に使用し、それらの動的かつ立体的な連携を実現するシステムである。導電体突起物を有するタッチケースを装着するだけで、一般的に普及している携帯端末を連携することができ、マルチ OS とマルチユーザの環境に対応している。評価実験では、4種類のケースの比較を行い本システムの認識率について調べ、実用に適したケースの仕様を確認した。以上より、本稿の貢献は、従来研究とはまったく異なるアプローチで、簡易に複数端末の連携を実現すると

ともに実用性の高い応用を可能にしている点であるといえる。

参考文献

[1] Baur, D., Boring, S. and Feiner, S.: Virtual Projection: Exploring Optical Projection As a Metaphor for Multi-device Interaction, *Proc. SIGCHI Conference on Human Factors in Computing Systems, CHI '12*, New York, NY, USA, ACM, pp.1693–1702 (2012).

[2] Boring, S., Baur, D., Butz, A., Gustafson, S. and Baudisch, P.: Touch Projector: Mobile Interaction Through Video, *Proc. SIGCHI Conference on Human Factors in Computing Systems, CHI '10*, New York, NY, USA, ACM, pp.2287–2296 (2010).

[3] Dalsgaard, P. and Halskov, K.: Tangible 3D Tabletops: Combining Tangible Tabletop Interaction and 3D Projection, *Proc. 7th Nordic Conference on Human-Computer Interaction: Making Sense Through Design, NordiCHI '12*, New York, NY, USA, ACM, pp.109–118 (2012).

[4] Gorbet, M.G., Orth, M. and Ishii, H.: Triangles: Tangible Interface for Manipulation and Exploration of Digital Information Topography, *Proc. SIGCHI Conference on Human Factors in Computing Systems, CHI '98*, pp.49–56 (1998).

[5] Jones, B., Sodhi, R., Murdock, M., Mehra, R., Benko, H., Wilson, A., Ofek, E., MacIntyre, B., Raghuvanshi, N. and Shapira, L.: RoomAlive: Magical Experiences Enabled by Scalable, Adaptive Projector-camera Units, *Proc. 27th Annual ACM Symposium on User Interface Software and Technology, UIST '14*, pp.637–644 (2014).

[6] Leigh, S.-W., Schoessler, P., Heibeck, F., Maes, P. and Ishii, H.: THAW: Tangible Interaction with See-through Augmentation for Smartphones on Computer Screens, *Proc. Adjunct Publication of the 27th Annual ACM Symposium on User Interface Software and Technology, UIST'14 Adjunct*, New York, NY, USA, ACM, pp.55–56 (2014).

[7] Marder-Eppstein, E.: Project Tango, *ACM SIGGRAPH 2016 Real-Time Live!, SIGGRAPH '16*, pp.40:25–40:25 (2016).

[8] Nacenta, M.A., Sakurai, S., Yamaguchi, T., Miki, Y., Itoh, Y., Kitamura, Y., Subramanian, S. and Gutwin, C.: E-conic: A Perspective-aware Interface for Multi-display Environments, *Proc. 20th Annual ACM Symposium on User Interface Software and Technology, UIST '07*, New York, NY, USA, ACM, pp.279–288 (2007).

[9] Nagakura, T. and Oishi, J.: Deskrama, *ACM SIGGRAPH 2006 Emerging Technologies, SIGGRAPH '06*, New York, NY, USA, ACM (2006).

[10] 中島康祐, 伊藤雄一, ジモンフェルカー, クリスチャントーレン, シェルオーバーゴード, ヤンボルヒャース: PUCs: 静電容量方式マルチタッチパネルにおけるユーザの接触を必要としないウィジェット検出手法, *情報処理学会論文誌*, Vol.56, No.1, pp.329–337 (2015).

[11] Nielsen, H.S., Olsen, M.P., Skov, M.B. and Kjeldskov, J.: JuxtaPinch: Exploring Multi-device Interaction in Collocated Photo Sharing, *Proc. 16th International Conference on Human-computer Interaction with Mobile Devices & Services, MobileHCI '14*, New York, NY, USA, ACM, pp.183–192 (2014).

[12] Ohta, T. and Tanaka, J.: Pinch: An Interface That Relates Applications on Multiple Touch-screen by 'Pinching' Gesture, *Proc. 9th International Conference on Advances in Computer Entertainment, ACE'12*, Berlin, Heidelberg, pp.320–335, Springer-Verlag (2012).

[13] Rädle, R., Jetter, H.-C., Marquardt, N., Reiterer, H. and Rogers, Y.: HuddleLamp: Spatially-Aware Mobile Displays for Ad-hoc Around-the-Table Collaboration, *Proc. 9th ACM International Conference on Interactive Tabletops and Surfaces, ITS '14*, New York, NY, USA, ACM, pp.45–54 (2014).

[14] Spindler, M., Stellmach, S. and Dachselt, R.: PaperLens: Advanced Magic Lens Interaction Above the Tabletop, *Proc. ACM International Conference on Interactive Tabletops and Surfaces, ITS '09*, pp.69–76 (2009).

[15] Willis, K.D.D., Shiratori, T. and Mahler, M.: HideOut: Mobile Projector Interaction with Tangible Objects and Surfaces, *Proc. 7th International Conference on Tangible, Embedded and Embodied Interaction, TEI '13*, pp.331–338 (2013).

[16] Yasumoto, M. and Teraoka, T.: Shadow Shooter: 360-degree All-around Virtual 3D Interactive Content, *ACM SIGGRAPH 2015 Posters, SIGGRAPH '15*, pp.29:1–29:1 (2015).

[17] Yasumoto, M. and Teraoka, T.: VISTouch, *ACM SIGGRAPH 2015 Posters, SIGGRAPH '15*, pp.41:1–41:1 (2015).

[18] Yasumoto, M. and Teraoka, T.: VISTouch: Dynamic Three-dimensional Connection Between Multiple Mobile Devices, *Proc. 6th Augmented Human International Conference, AH'15*, New York, NY, USA, ACM, pp.89–92 (2015).



安本 匡佑

2010年東京藝術大学大学院映像研究科博士後期課程修了。博士（映像メディア学）。2011年より東京工科大学メディア学部助教。2015年より神奈川工科大学情報学部情報メディア学科助教。2017年より同大学准教授。メディアアーティストとして近年では「The Light Shooter」「参式電子弓」「CUBISTA」等の作品を制作。



寺岡 文博（正会員）

2005年慶應義塾大学環境情報学部卒業。2007年同大学大学院政策・メディア研究科修士課程、2013年に後期博士課程修了。博士（学術）。2011年より慶應義塾インフォメーションテクノロジーセンター本部助教。2014年より東京工科大学メディア学部助教。自然言語処理の研究に従事。人工知能学会、言語処理学会、日本認知科学会各会員。