

# カードボード VR のための空中操作を用いた ターゲット選択手法

石井 晃 阿達 拓也 島 佳吾 中前 秀太 志築 文太郎 高橋 伸

スマートフォン (以下, 携帯端末) をヘッドマウントディスプレイとして用いることによりバーチャルリアリティ環境 (以下, カードボード VR 環境) を実現する製品がある。このようなカードボード VR 環境では安価かつ手軽に VR 体験が可能である一方で, タッチパネルを覆うことによりタッチ操作が制限されるため, ゲームのようなインタラクションを必要とするアプリケーションは実現が困難である。この問題に対し本研究においては, 携帯端末の背面空間にて手を動かすことによるターゲット選択手法を示す。提案手法は, 携帯端末の背面カメラにより検出した手の位置に応じて画面上にポインタを表示する。ユーザは親指を曲げる動作により選択を行う。提案手法は携帯端末に内蔵されたカメラのみで実装可能であるため, カードボード VR 環境におけるターゲット選択手法として容易に適用可能である。提案手法の性能を調べるため, ターゲット選択時間およびエラー率を評価する実験をカードボード VR 環境において行った。また, 提案手法を用いたゲームを制作し使用感調査を行った。

Cardboard virtual reality (VR) environment using a cardboard viewer and smartphone as a head-mounted display (HMD) opens the door to new possibilities where people can experience VR easily and inexpensively. However, such cardboard VR environment covers the touch screen of the smartphone and thus restricts the user from touching the screen. To address this problem, we present FistPointer, a target selection technique using mid-air interaction behind a smartphone for cardboard VR environment. Our technique displays a pointer on the screen corresponding to the position of the hand detected by the built-in back camera of the smartphone. The user can move the pointer by moving the hand in a thumbs-up posture and select a target by folding the thumb similar to pushing the button of a joystick. Our technique can be implemented using only the built-in camera of a smartphone, thus it is easy to apply our technique to a target selection in cardboard VR environment. To evaluate the performance of our technique, we conducted experiments in selection task. Furthermore, we developed a game using our technique and investigated the user impressions.

## 1 はじめに

Google Cardboard [6] やハコスコ [9] に代表される, スマートフォン (以下, 携帯端末) をヘッドマウントディスプレイ (HMD) のディスプレイとして用いることによりバーチャルリアリティ (VR) 環境 (以下, カードボード VR 環境) を実現する製品がある。このようなカードボード VR 環境では安価かつ手軽に VR 体験が可能である一方で, タッチ操作はタッチパネル

が覆われることにより制限されるため, 現状ではコンテンツの閲覧が主な用途となっており, ゲームのような積極的なインタラクションを必要とするアプリケーションは実現が困難である。

この問題に対し我々は, 携帯端末の背面空間にて手を動かすことによるターゲット選択手法 FistPointer を示す (図 1)。FistPointer は, 携帯端末の背面カメラにより検出した手の位置に応じて画面上にポインタを表示する。ユーザは図 1 に示す姿勢にて手を動かすことによりポインティングを行った後, 親指を曲げる操作 (クリック操作) によりターゲット選択を行う。提案手法は手の検出に携帯端末の背面カメラを用いるため, カードボード VR 環境向けのターゲット選択手法として追加の機器無しに適用可能であり, カードボード VR 環境に対する新たな入力手法となり得る。

---

Target Selection Technique using Mid-air Interaction for Cardboard VR Environment.

Akira Ishii, Takuya Adachi, Keigo Shima, Shuta Nakamae, Buntarou Shizuki, and Shin Takahashi, 筑波大学, University of Tsukuba.

コンピュータソフトウェア, Vol.35, No.2 (2018), pp.64–78.  
[研究論文] 2017 年 6 月 16 日受付。



図 1 FistPointer の概観.

今回、我々は提案手法の性能を調べるためターゲット選択時間およびエラー率を評価する実験を行った。また、アプリケーション例として提案手法を用いたシューティングゲームを制作し使用感調査を行った。本論文では、以上の結果について報告する。

## 2 関連研究

### 2.1 大型ディスプレイ向けのターゲット選択

手を空中にて動かすことによるターゲット選択手法は大画面ディスプレイ向けの手法として数多く研究されてきた[7]。例えば、Vogel および Balakrishnan [24] は、大型ディスプレイ向けに、ジェスチャによる3つのポインティング手法および2つのクリック手法を提案した。Markussen ら[17] は、大型ディスプレイ向けに、空中インタラクションを用いた3つのテキスト入力手法を提案した。Fukazawa ら[4] は、複数の大画面ディスプレイにおけるマルチモーダルインタラクションについて検討している。築谷ら[26] は、大画面ディスプレイ向けに影のメタファを用いたポインティング手法を提案している。対照的に、我々はカードボード VR 環境向けの選択手法に焦点を当てる。

### 2.2 手をカメラで認識することによる

#### 携帯端末操作手法

Baldauf らは、検出した指先の位置によりポインティングを行うインタフェースを提案し[1]、また評価を行っている[14]。Lv ら[16] は携帯端末を腕時計の

ように腕に固定して使うような状況を想定し、携帯端末に搭載されているバックカメラを用いて検出した指の方向(角度)によりアプリケーションを操作する手法を示した。Song ら[22] はカメラで検出した空中における手の形やジェスチャによる操作を提案した。この研究においては、空中におけるジェスチャを機能の呼び出し等に割り当てタッチ操作と組み合わせることで柔軟な操作を実現している。また、携帯端末に外部カメラを取り付けることによりポインティングを可能とする研究もなされてきた。Niikura ら[18] はハイスピードカメラと赤外線 LED を用いて指のトラッキングを行った。Wigdor ら[25] は、Web カメラを用いて端末の背面タッチを認識し、10 本指でのマルチタッチ入力を可能とした。

### 2.3 VR 環境向けのターゲット選択手法

近年、VR 向けのポインティング手法が数多く提案されている。中には、カメラを用いて認識した手に基づく手法も多く考案されている。例えば、Sugiura ら[23] は HMD に取り付けられた Web カメラにて認識した人差し指を用いたポインティング手法を開発した。ユーザは HMD に表示された仮想ボタンを指先にてポインティングを行い、タップのような動作で選択することができる。Kato ら[11] は、手のジェスチャによるクリックおよびポインティングを検出できるシステムを開発し、HMD に表示されたターゲットの選択において、マウスによる選択と速さを比較する実験を行った。

その他にも Lee ら[15] および Gugenheimer ら[8] は HMD の前面にタッチパネルを取り付けることによりユーザのタッチを認識し、仮想空間への操作を可能にする研究を行った。Lee らの手法は仮想空間のターゲット選択に手の動きに応じて移動するポインタを用いる。Gugenheimer らはタッチ位置に応じてポインタを表示することによって操作を可能にしている。Rupprecht ら[20] はスマートウォッチをセンサとして用いることにより、Kato ら[12] は銀ナノ粒子インクにて導電性のパターンをカードボードに印刷することにより、VR 環境における操作を実現している。Petry ら[19] は、HMD の前面に取り付けられ

た Leap Motion により、ユーザの顔の前での手の左右の動きを取得し、パノラマビデオの操作を行っている。

## 2.4 本研究の位置づけ

本研究においては、追加の機器なしにカードボード VR 環境へのインタラクション性の付与を行うためのターゲット選択手法の設計、および携帯端末の背面カメラを用いた実装法と、ターゲット選択手法の評価を行う。携帯端末において追加の機器なしにカメラで手を認識しターゲット選択に用いるという点では上記に述べた研究 ([1][14][16][22]) と同様のアプローチである。ターゲット選択手法の設計においては、これまで関連研究において挙げてきた Natural User Interface (NUI) において広く一般的に用いられている人差し指を立てる姿勢によりポインティングを行い、そしてその指をタッチパネルにおけるタップ操作のように軽く折り曲げることにより選択を行うという操作姿勢ではなく、新たに親指を立てる姿勢 (サムズアップポーズ) によりポインティングを行い、そしてその指を完全に折り曲げることにより選択を行うという操作姿勢を用いることを試みた。

## 3 提案手法：FistPointer

FistPointer は、携帯端末の背面空間にて手を動かすことによるターゲット選択手法である。ユーザは図 2(左) に示す姿勢にて手を動かすことによりポインティングを行った後、図 2(右) に示すように親指を曲げる操作 (クリック操作) によりターゲット選択を行う。FistPointer は携帯端末の背面カメラから得た RGB 画像 (今回の実装では  $320 \times 240$  ピクセル) 中の手の位置に応じて画面上にポインタを表示し、画像中の手の形状からクリック操作の判定を行う。本章にて、手の検出、ポインタ座標の決定、およびクリック操作の判定の方法を示す。

### 3.1 手の検出

まず RGB 画像の各画素が手に相当するか否かを決定する。これには各画素に対して Song ら [22] の式 (1) を用いた。



図 2 提案手法に用いられる手の姿勢。  
左：ポインティング時、右：クリック時。

$$S(u) = \begin{cases} 1 & (\min(R(u) - G(u), R(u) - B(u)) > \tau) \\ 0 & (\text{otherwise}) \end{cases} \quad (1)$$

$R(u)$ ,  $G(u)$ ,  $B(u)$  はそれぞれ座標  $u$  にある画素の RGB 値であり、 $\tau$  は閾値である。結果として  $S(u)$  は、座標  $u$  が手に相当する場合には 1、それ以外の場合には 0 となる。

次に各  $y$  座標において  $S(u) = 1$  となる画素群のうち最も幅のあるものを手とする。これにより、手の幅より小さなノイズを除去することが可能である。なお、ユーザは環境光および肌の色に応じて  $\tau$  を調整する必要がある。ただし、閾値は 1 つのみであるため調整は容易である。

### 3.2 ポインタの座標の決定

提案手法ではポインタ座標に対するクリック操作による影響を抑えるために、親指の動きに影響を受けない位置である手の右上 (人差し指の付け根に相当) をポインタとした。提案手法ではクリック操作時に親指を上下に折り曲げるため、親指の先端にポインタを表示するとクリック操作の時にポインタがぶれる。手の右上は親指の動きに影響されず、またユーザの操作の意識が最も向いている親指に最も近い位置であるため、ここにポインタを表示した。このポインタの座標を  $(x_p, y_p)$  とする。各フレームにおけるポインタ座標を求めるには、 $y_p$  を先に求め、次に  $y_p$  をもとに  $x_p$  を求めることとした。このためにまず各  $y$  座標における手の幅を求める。次に  $y$  座標  $i$  において、 $i$  を含む直前の 9 つの  $y$  座標  $(y_i, \dots, y_{i-8})$  における手の幅の回帰直線の傾きを求め (図 3)、この回帰直線の傾

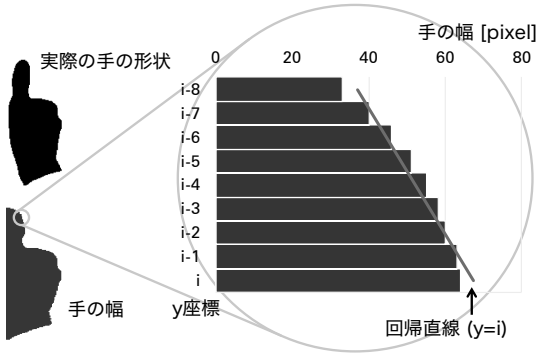
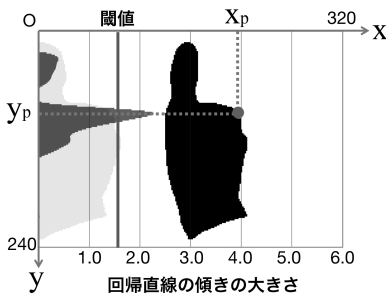
図3  $y$  座標  $i$  における回帰直線.

図4 検出した手の形状 (中央), 幅 (薄いグラフ), および回帰直線の傾きの大きさ (濃いグラフ).

き ( $\Delta x / \Delta y$ ) を指先 (肌領域における最上端) 以降のすべての  $y$  座標に対して求める. 回帰直線の傾きがなす波形および手として検出された画素を重量表示した様子を図4に示す. この図から, 親指の指先および付け根において傾きが大きくなっていることが分かる. 特に, 親指の付け根における傾きは親指の指先と比べても大きい. そこで, 回帰直線の傾きのピークを見つけ, そこを  $y_p$  とする. 次に,  $y_p$  において手と認識された画素群のうちの右端 (図4に丸で示されている座標) の  $x$  座標を  $x_p$  とする.

なお, ポインタ座標を決めるために手の幅を利用すること, およびクリック操作の判定に親指を検出する必要があることから, カメラ画像には手全体が映っていないといけない. 以上により手が移動できる範囲が限られるため, ポインタ座標の取りうる値の範囲は実際のカメラ画像のサイズよりも狭くなる. そこ

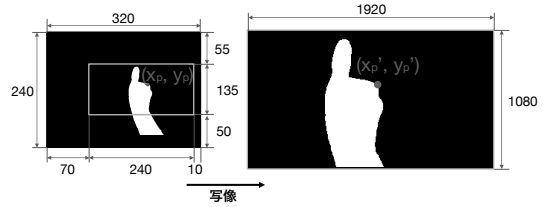


図5 カメラ画像と操作領域間のポインタの位置関係.

で, 図5の矩形領域 (切り取り矩形領域) 内のポインタの座標が操作領域に写像されるようにした. 写像後の座標 ( $x'_p, y'_p$ ) はそれぞれ式 (2), (3) にて表される (操作領域のサイズが  $1920 \times 1080$  ピクセルの場合).

$$x'_p = \begin{cases} 0 & (x_p < 70) \\ (x_p - 70) \times 8 & (70 \leq x_p \leq 310) \\ 1920 & (x_p > 310) \end{cases} \quad (2)$$

$$y'_p = \begin{cases} 0 & (y_p < 55) \\ (y_p - 55) \times 8 & (55 \leq y_p \leq 190) \\ 1080 & (y_p > 190) \end{cases} \quad (3)$$

なお, 提案手法はカメラ画像からポインタ座標を決定するため, カメラと手の距離に応じてポインタの移動量を調節することができるという特徴を持つ. つまり, ユーザは手をカメラに近づけることにより大まかな操作が素早く行え, 手をカメラから遠ざけることにより細かな操作が行える.

### 3.3 クリック操作の判定

クリック操作は, 親指座標がポインタ座標に近づいた時に行われたと判定する. まず, 手として認識された画素のうち,  $y$  座標が最も小さい画素群の中央座標を親指座標 ( $x_t, y_t$ ) とする. 手の大きさ, およびカメラまでの距離の変化に対応できるよう,  $y_p$  における手の幅を親指の高さ ( $y_t$  から  $y_p$  までの長さ) で割った値を求める. この値が閾値 (今回は実験的に求めた 1.6) を超えたとき, クリック操作が行われたと判定する.

#### 4 実験 1：ターゲットサイズ別選択性能評価

カードボード VR 環境における FistPointer のターゲット選択時間およびエラー率を評価する実験を行った。VR 環境においては HMD を頭に固定するため、ターゲット選択に両手を用いることが可能である。そこで、操作に用いる腕をもう一方の腕にて支える support 条件および支えない non-support 条件を支え条件 (図 6) として設け、性能の比較を行った。

##### 4.1 実験協力者および実験機器

研究室内の大学生および大学院生のボランティア 6 名 (男性 6 名, 年齢 21-24 歳, 平均 22.7 歳) を実験協力者とした。全ての実験協力者が日常的にスマートフォンを利用し, 全員が右利きであった。スマートフォンの利用歴は平均 75.7ヶ月であった。すべての実験協力者はスマートフォンを用いた VR アプリケーションを使用しなかった。また, すべての実験協力者は正常な裸眼もしくは矯正視力であり, 実験に支障をきたさない視力であった。3 名がメガネを着用し, 1 名がコンタクトレンズを着用していた。

実験には Android 端末 (ELUGA P-03E, 端末サイズ: 高さ 132 mm × 幅 65 mm × 厚さ 10.9 mm, 画面サイズ: 4.7 インチ, 解像度: 1920 × 1080 ピクセル, カメラレンズの F 値: 2.4, カメラレンズの焦点距離: 29 mm, OS: Android 4.2.2) をハコスコ タタミ 1 眼 [9] に装着した HMD を用いた。この HMD は手にて筐体を支える必要がある。そのため今回の実験では HMD を頭部に固定できるように, マジックテープ製のバンドおよびスポンジを HMD に取り付けた (図 7)。

##### 4.2 支え条件

FistPointer は片手にて使用可能な操作手法であるが, 今回は HMD を頭部に固定したため両手を使用することができる。操作に用いる腕をもう一方の腕にて支えることにより, 操作する腕が安定するため選択速度およびエラー率の向上が期待される。そこで, 支え条件がどの程度選択時間およびエラー率に影響を及ぼすかを調査することとした。そのために, 操作

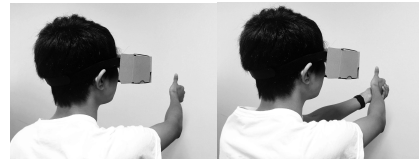


図 6 支え条件. 左: non-support 条件. 右: support 条件.



図 7 実験において使用した HMD.

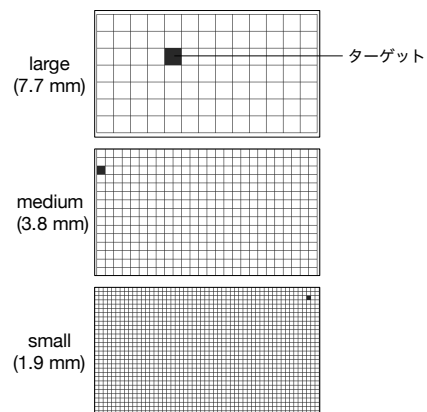


図 8 ターゲット条件.

に用いる腕をもう一方の腕にて支える support 条件 (図 6 右), および支えない non-support 条件 (図 6 左) を支え条件として設けた。

##### 4.3 ターゲット条件

ターゲットの大きさがターゲットの選択時間およびエラー率に及ぼす影響を調査するため, 図 8 に示すように 3 種類のターゲット条件を用意した。実験協力者が次に選択すべきターゲットは格子内においてハイライトされた正方形のセルである。Google が推奨している [5] 最低のターゲットの幅である 48 dp (今回使用した端末において 144 ピクセル, 7.7 mm) を最大と

し、1/2 ずつ小さくしたターゲットを 3 種類用意した。つまり、ターゲットの 1 辺は 48, 24, 12 dp (7.7, 3.8, 1.9 mm) である。以降、それぞれを **large**, **medium**, **small** 条件と呼ぶ。図 8 に示すように、各条件における格子の行数および列数はそれぞれ  $7 \times 13$ ,  $15 \times 26$ ,  $30 \times 53$  となる。

#### 4.4 実験手順

実験は蛍光灯によって十分に照らされた静かな会議室で行われた。カメラの撮影する背景は汚れ、模様、および影の無い白い壁であった。実験者はまず、実験協力者に椅子に深く腰掛けるよう指示した。次に実験協力者の頭部に HMD を違和感なく、かつ頭を左右に振ってもずれないように装着した。その後、実験協力者に FistPointer によるポインティングおよびクリックが正常に動作するかどうかを、画面中央および四隅に出現するターゲットをクリックすることにより確認するよう指示した。正常に動作しなかった場合は、閾値を調整してから再度動作確認を行うこととした。その後実験協力者には large 条件において、操作に慣れるまで繰り返し中央および四隅の選択を練習タスクとして行ってもらった。このとき、実験条件をそろえるため、実験者は実験協力者に、右手を用いて FistPointer の操作を行うように指示した。また、ターゲットをできるだけ速く、かつ正確に選択するよう指示した。練習後、実験協力者は本番タスクを開始した。

実験協力者に行ってもらったタスクは、ランダムな位置に表示されるターゲットの選択 1 回を 1 試行とし、20 試行を 1 セッションとするタスクである。セッションは、実験協力者が 1 つ目のターゲットの選択に成功した直後から開始される。実験協力者がターゲットの選択に成功すると次のターゲットが表示された。20 回選択に成功すると格子が赤色になりセッションが終了する。実験協力者には、支え条件ごとに large, medium, small 条件順に 1 セッションずつ行ってもらった。よって各実験協力者は、合計 120 試行 (= 20 試行  $\times$  2 支え条件  $\times$  3 ターゲット条件) を行った。また、手法間の順序効果を相殺するため、実験協力者を 2 グループに分け、一方を support 条件か

ら、もう一方を non-support 条件から行うこととした。疲労による影響を軽減するために、実験協力者にはセッション間に 1 分以上の休憩をとってもらった。

全てのセッション終了後に、実験協力者には使用感に関するアンケートに回答してもらった。本実験は、約 40 分の時間を要した。

#### 4.5 実験結果

1 試行に要した選択時間を図 9 に、エラー率を図 10 に示す。支え条件とターゲット条件の選択時間およびエラー率への影響を検証するため、二元配置分散分析を行った。なお検定に先だって、ターゲットの中心の座標と選択した座標の  $x$  または  $y$  の差が  $\mu \pm 3 \times \sigma$  ( $\mu$ : 平均値,  $\sigma$ : 標準偏差) の範囲を外れる選択を外れ値として除外した。これは、ターゲットの選択に成功した後、誤って再びそのターゲットを選択すること、および FistPointer において実験協力者の意図しないクリック操作判定<sup>†1</sup>があったためである。全選択数における外れ値の割合は 0.96% であった。なお、ここではターゲットの領域外を選択した場合をエラーとし、エラー率を「エラー回数/選択総回数」と定義した。

二元配置分散分析の結果、選択時間に関して支え条件には主効果が認められず、ターゲット条件に主効果が認められた ( $F_{2,10} = 37.66$ ,  $\eta^2 = 0.88$ ,  $p < 0.001$ )。両条件に交互作用は認められなかった。主効果が認められたターゲット条件に対して Tukey 法による多重比較検定を行ったところ、以下のことがわかった。1) large 条件は small 条件と比べて有意に選択時間が速い ( $p < 0.001$ )。2) medium 条件は small 条件と比べて有意に選択時間が速い ( $p < 0.001$ )。

また、二元配置分散分析の結果、エラー率に関しても支え条件には主効果が見られず、ターゲット条件に主効果が認められた ( $F_{2,10} = 59.03$ ,  $\eta^2 = 0.92$ ,  $p < 0.001$ )。両条件に交互作用は認められなかった。なお、有意差は認められなかったものの、small 条件下においては support 条件の方が約 6.5% エラー率が

<sup>†1</sup> 本手法は図 5 に示すようにカメラ画像の一部を切り取り、そして操作領域に写像している。この時にこの切り取り矩形領域内から腕が外れた場合に意図しないクリック判定が発生した。

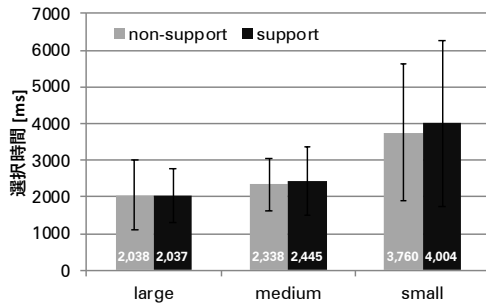


図 9 ターゲット選択時の選択時間。  
エラーバーは標準偏差を示す。

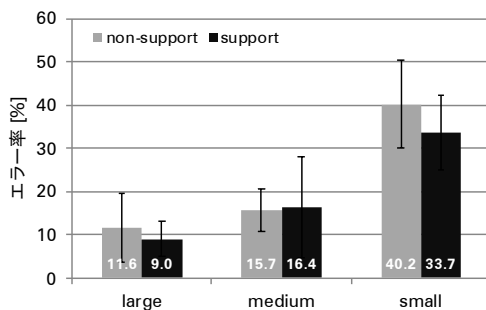


図 10 ターゲット選択時のエラー率。  
エラーバーは標準偏差を示す。

低かった。主効果が認められたターゲット条件に対して Tukey 法による多重比較検定を行ったところ、以下のことがわかった。1) large 条件は small 条件と比べて有意にエラー率が低い ( $p < 0.001$ )。2) medium 条件は small 条件と比べて有意にエラー率が低い ( $p < 0.001$ )。

#### 4.6 議論

実験結果からは支え条件による選択時間およびエラー率への影響は無かった。ただし、今回の VR 環境実験においては腕を支える位置は一箇所のみしか検討していない。そのため、腕を支える位置を複数箇所設定しそれらの条件下における選択時間およびエラー率を比較することにより、より最適な操作姿勢を特定する必要がある。

また、実験協力者のコメントから、ターゲットの領域ごとにエラー率に偏りがあることが示唆された。1 人の実験協力者は、「スクリーンの左側のターゲット

は右側のターゲットに比べて選択しにくかった」と答えており、また 2 人の実験協力者は「スクリーンの端はより選択しにくかった」と回答した。スクリーンの左側は右手から最も遠い位置にあるため、実験協力者は他のターゲットを選択する時よりも左側のターゲットを選択する時は腕を伸ばす必要があり、より選択しにくい姿勢になるため、このような結果になったと推測される。

### 5 実験 2：ターゲット領域別選択性能評価

実験 1 の結果において、large と small 条件間、および medium と small 条件間において選択時間とエラー率に有意差が見られた。また、実験 1 において、実験協力者は画面中心部のターゲットよりも周辺部のターゲットの方が選択しにくかったとコメントしている。そこで、画面内の領域 (以降、ターゲット領域) ごとのエラー率を明らかにするために、medium 条件にて新たな実験を行った。この実験により、ターゲット領域ごとのエラー率を示すとともに、カードボード VR 環境において提案手法を用いたユーザインタフェースの設計指針を示す。なお、実験 1 の結果において支え条件には有意差が見られず、かつ、カードボード VR においては片方の腕を用いてカードボードを固定する状態 (すなわち non-support 条件) にて利用することが一般的であるため、non-support 条件のみにて実験を行った。

#### 5.1 実験協力者および実験機器

実験協力者 15 名 (男性 14 名、女性 1 名、年齢 21–30 歳、平均 23.9 歳) をこの実験のために雇用した。14 名の実験協力者が日常的にスマートフォンを利用し (平均利用歴：45.6 ヶ月)、全員が右利きであった。3 名の実験協力者は VR アプリケーションを過去に使用した経験があった。すべての実験協力者は正常な裸眼視力もしくは矯正視力を有していた。9 名がメガネを着用し、3 名がコンタクトレンズを着用していた。

本実験には、実験 1 と同程度のカメラ性能を持つ Android 端末 (Xperia Z5, 端末サイズ：高さ 146 mm × 幅 72 mm × 厚さ 7.3 mm, 画面サイズ：5.2 インチ、解像度：1920 × 1080 ピクセル、カ

メラレンズの F 値: 2.0, カメラレンズの焦点距離: 28mm, OS: Android 6.0.1) をハコスコ タタミ 1 眼に装着した HMD を用いた。今回は non-support 条件下において実験を行うため, HMD に変更を加えなかった。なお, 実験 2 の実験協力者は全員, 実験 1 には参加していなかった。

## 5.2 実験計画

medium 条件における総ターゲット数は 390 個 (= 15 行 × 26 列) と多い。本実験においては, 画面領域ごとのエラー率を求めることを主目的としたため, 慣れおよび疲れによるエラー率への影響を最小限に留める必要がある。そのため, 実験 1 と同様に実験時間が 1 時間を超過しないように, 実験協力者を 5 人ずつ 3 グループ (A, B, C) に分け, 390 個のターゲットをランダムに 130 ターゲットごとに分割した。よって各ターゲットは実験全体で 5 回ずつ選択された。

## 5.3 実験手順

実験は実験 1 と同様の環境で行われた。実験者はまず, 実験協力者に椅子に深く腰掛けるよう指示した。次に実験協力者に HMD を手渡した。実験条件をそろえるため, 実験者は実験協力者に, 左手にて HMD を把持し, 右手を用いて FistPointer の操作を行うよう指示した。その後, 実験協力者に FistPointer によるポインティングおよびクリックが正常に動作するかどうかを, 画面中央および四隅に出現するターゲットをクリックすることにより確認するよう指示した。正常に動作しなかった場合は, 閾値を調整してから再度動作確認を行うこととした。動作確認後, 実験協力者には練習タスクとして, 中央および四隅の選択を少なくとも 4 セット繰り返してもらった。練習タスク終了後, 実験者は実験協力者が中央および四隅の選択を失敗することなく完了できることを確認した。このとき実験協力者が選択に失敗した場合は, 失敗することなく完了できるまで繰り返して行った。その後, 本番タスクを開始した。

タスクとして実験協力者に 1 つずつターゲットを選択してもらった。これを 1 試行と定義する。実験協

力者には 5.2 節にて示した, グループ毎に割り当てられた 130 個すべてのターゲットを選択してもらった。実験協力者がターゲットの選択に成功すると次のターゲットが表示された。なお, 順序効果を打ち消すために, ターゲットは重複なしのランダムな順序にて提示された。疲労による影響を軽減するために, 実験協力者には 20 試行終了するごとに 1 分以上の休憩をとってもらった。なお, ターゲットの総数は 130 個のため最後の 10 試行で終了となる。

すべての試行終了後に, 実験協力者には System Usability Scale (SUS) [2][3], および使用感に関するアンケートに回答してもらった。ただし, SUS は英語で記述されているものの, 本実験の実験協力者は全員日本語を第一言語としていたため, 今回は古井ら [27] によって日本語に翻訳されたものを原文に併記した。また, アンケート調査も実施した。「FistPointer は簡単に使えましたか?」「FistPointer は思ったように操作することができましたか?」という問いに対して 5 段階のリッカート尺度 (1 = 強くそう思わない, 5 = 強くそう思う) を用いて回答を求めた。本実験は, 実験説明からすべての試行およびアンケートへの回答が終了するまでに, 約 40 分の時間を要した。実験終了後, 実験協力者に 820 円支払った。

## 5.4 実験結果

すべてのターゲットの平均選択時間は 2532 ms (SD = 786) であった。今回の結果の妥当性を確かめるため, 今回の結果と実験 1 の結果を t 検定にて検定を行った。その結果, 有意差はみられなかった。一方で, エラー率に関しては興味深い結果が得られた。

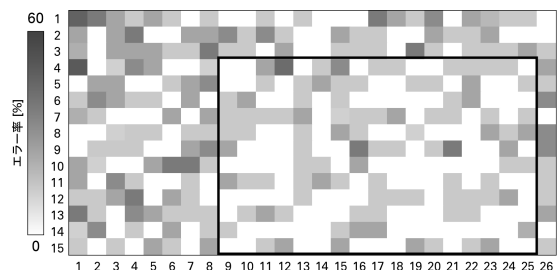


図 11 実験 2 の結果。各領域におけるエラー率。矩形で囲まれた領域は, 平均エラー率が 10% 未満の領域である (精度 90%)。



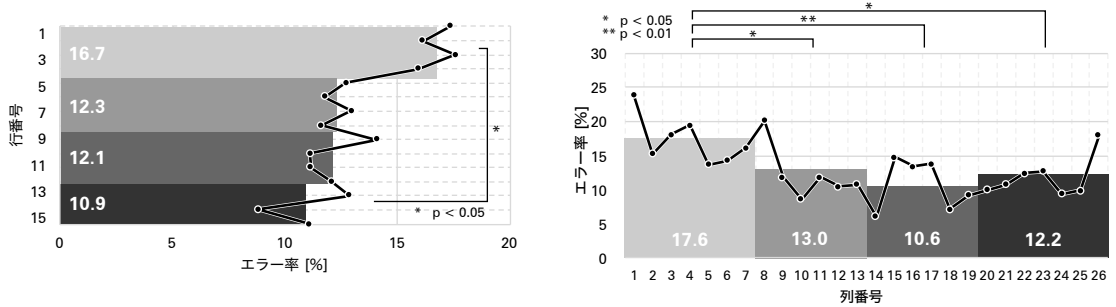


図 12 実験 2 の結果。左：各行のエラー率。上から下へ 1–15 行を示す。右：各列のエラー率。左から右へ 1–26 列を示す。線グラフは各行もしくは各列の平均エラー率を示す。棒グラフは 4 つの領域に分割された各行もしくは各列の平均エラー率を示す。

#### 5.4.1 エラー率

すべてのターゲットの平均エラー率は 13.1%(SD = 13.3) であった。今回の結果の妥当性を確かめるため、今回の結果と実験 1 の結果を t 検定にて検定を行った。その結果、有意差はみられなかった。ターゲット別エラー率をヒートマップにて図 11 に、medium 条件にある 15 行の各行ごとの平均エラー率を折れ線グラフにて図 12(左) に、各列ごとの平均エラー率を折れ線グラフにて図 12(右) に示す。また、ターゲット領域間エラー率の違いを調べるため、すべての行を 4 領域に分け (1–4 行 (最上)、5–8 行、9–12 行、13–15 行 (最下))、それぞれの領域の平均エラー率を棒グラフにて図 12(左) に示す。列ごとに 4 領域に分け (1–7 列 (最左)、8–13 列、14–19 列、20–26 列 (最右))、それぞれの領域の平均エラー率を棒グラフにて図 12(右) に示す。

ターゲット領域のエラー率への影響を検証するため、一元配置分散分析を行った。その結果、行ごとに 4 分割したターゲットにおけるエラー率に関して、ターゲット領域に主効果が認められた ( $F_{3, 386} = 3.68$ ,  $\eta^2 = 0.28$ ,  $p < 0.05$ )。Tukey 法による多重比較検定を行ったところ、最上領域 (1–4 行) のエラー率は最下領域 (13–15 行) と比べ有意に高かった ( $p < 0.05$ )。また、列ごとに 4 分割したターゲットにおけるエラー率に関して、ターゲット領域に主効果が認められた ( $F_{3, 386} = 4.87$ ,  $\eta^2 = 0.36$ ,  $p < 0.01$ )。Tukey 法による多重比較検定を行ったところ、最左領域 (1–7 列) のエラー率はその他の領域と比べ有意に高かった

(8–13 列 :  $p < 0.05$ , 14–19 列 :  $p < 0.01$ , 20–26 列 :  $p < 0.05$ )。

さらに、ターゲットを行ごとおよび列ごとに  $4 \times 4$  にて計 16 分割し、ターゲット領域を独立変数、エラー率を従属変数とする一元配置分散分析を行った。なお、分割の仕方は上記の各行および各列における 4 分割と同様である。分散分析の結果、ターゲット領域に主効果が認められた ( $F_{15, 374} = 1.99$ ,  $\eta^2 = 0.074$ ,  $p < 0.05$ )。Tukey 法による多重比較検定を行ったところ、最左上領域 (1–4 行  $\times$  1–7 列) のエラー率はやや左寄り真ん中領域 (9–12 行  $\times$  8–13 列)、および最右下領域 (13–15 行  $\times$  20–26 列) と比べ有意に高かった ( $p < 0.05$ )。

また、高精度にて操作可能な領域 (高精度な選択操作が要求されるユーザインタフェース部品を配置可能な領域) を調べるために、ユーザが平均 90%(large 条件における精度と同程度) の精度にてターゲットを選択することができる矩形領域を調査した。具体的には、各ターゲットで構成されるすべての矩形領域のエラー率を総当りで計算した。その結果、図 11 に示す矩形領域が最大面積の矩形領域であった。

#### 5.4.2 ユーザビリティ

ユーザビリティを調査するために行った SUS の平均スコアは 74.2(SD = 11.1) であった。また、アンケート調査の結果を図 13 に示す。

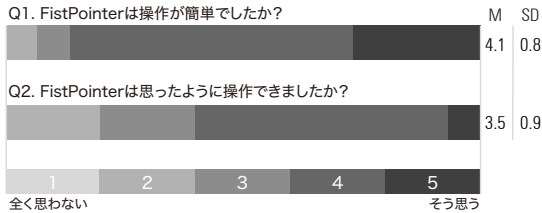


図 13 アンケート調査の結果.

## 5.5 議論

### 5.5.1 ターゲット領域ごとのエラー率

実験結果よりターゲット領域ごとにエラー率の偏りがあることがわかった。とりわけ、最左領域のエラー率は有意に高かった。アンケート回答からもこの結果を支持する結果が得られた。4 人の実験協力者が「左側に配置されたターゲットの方がより選択しにくかった」とアンケートに記述した。これは 3 つの理由に起因すると考えられる。1 つめは画面の左側 (利き手の反対側) 周辺のターゲットを選択するためには、実験協力者は腕を限界まで伸ばす必要があり、これは普段行わない姿勢であるため、操作しにくかったためであると推測される。2 つめはターゲット選択を左領域にて行おうとすればするほど、身体的な制約上、手とカメラの位置が近くなり、ターゲット選択精度が粗くなるためであると推測される。3 つめは検出された手の形が今回実装した手検出アルゴリズムにふさわしくない形として映るときがあったためであると推測される。図 14(左) に手検出アルゴリズムが想定している手の形を示す。この手検出アルゴリズムは図 14(左) のような形の手がカメラから取得されると仮定し実装されている。よって、図 14(右) に示すような手の形がカメラから取得された場合適切に手が検出されない。左側のターゲットを選択する場合腕を横へ伸ばす必要があり、このときカメラに対する手の角度が変化するため、このような現象が発生する。精度をより向上させるために、このような状況下においても強固に手を検出できるようなアルゴリズムに変更する必要がある。

また、最上領域のエラー率も高いことが分かった。これは腕を上へ上げれば上げるほど疲労が蓄積し、上部の選択時の性能に影響した可能性がある。この問題

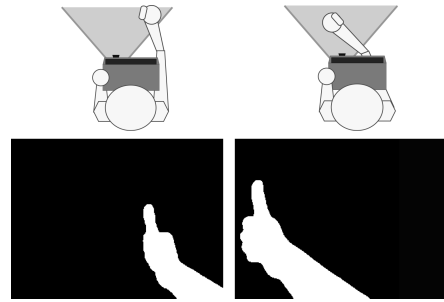


図 14 検出された手. 左: 検出アルゴリズムが正しく認識できる手の形. 右: 検出アルゴリズムが正しく認識できない手の形. なお、カメラは左側にある。

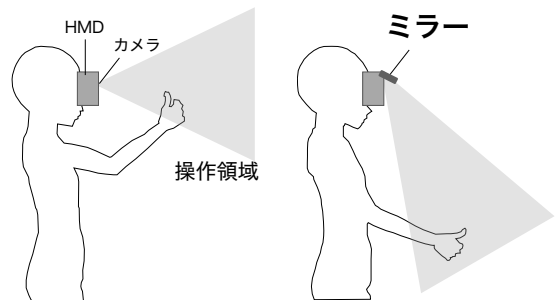


図 15 小型のミラーをカメラ部分に取り付けることにより操作領域を下部に移動することができる。

を解決するために、例えば図 5(左) における切り取り矩形領域を下方に移動する方法や、カメラの前方に小型のミラーを追加し図 15 のように操作領域を下部に移動する方法が考えられる。

### 5.5.2 設計指針

今回の実験からユーザの利き手 (操作している手) とは反対側の領域は選択がしにくいという結果が得られた。また、エラー率に関するその他のフィードバックもいくつか得られた。2 名の実験協力者は「画面の端に配置されたターゲットは選択がしにくかった」と答えた。カードボード VR の HMD には視野角を広くするためのレンズが使われており、その影響により画面の端は歪む。その結果、ターゲットを鮮明に捉えることができず選択が難しくなったと推測される。実験 2 においては実験 1 において用いたスマートフォンに比べ画面サイズが大きくなっており、より画面の端の歪みによる影響を受けたと考えられる。アンケー

ト回答からもこの結果を支持する結果が得られた。2 名の実験協力者は「画面端に配置されたターゲットは見にくかった」と回答した。

実験結果から、精度の高い選択を必要とするターゲット (重要なボタン等) は画面の右側 (操作している手側) もしくは真ん中領域に配置すると良いことがわかった。また、左側や周辺領域にターゲットを配置する場合にはターゲットのサイズを大きくした方が良い。

### 5.5.3 ユーザビリティ

5.4.2 節にて述べたように、我々のシステムはポジティブな評価を得た。これは、提案手法のジェスチャのデザインに起因すると考えられる。ポインティングのためのジェスチャ (腕を目の前に構える) はターゲットに狙いを定めるメタファに基づく。また、決定ジェスチャ (親指を折り曲げる) はボタンを押すメタファに基づく。そのため、これらのジェスチャには想起性があり、覚えやすかったと推測される。アンケート回答からもこの結果を支持する結果が得られた。12 名の実験協力者が「FistPointer のジェスチャは覚えやすく、また理解しやすかった」と回答した。

一方で、2 名の実験協力者は「選択を行う時に、ポインタが時々意図せずぶれてしまった」と回答した。これは選択時に親指を折り曲げる動作を行う際、つられて手そのものが少し動くためであると推測される。この問題に対し、選択ジェスチャの開始を検出した場合は、ポインタの座標を固定するという改善策が考えられる。この改良により選択動作がより安定し、FistPointer の精度を改善させることが期待される。

### 5.5.4 腕の疲れ

実験のアンケートより、疲れに関するフィードバックを得た。9 名の実験協力者は「もし FistPointer を長時間使用した場合、腕が疲れてしまうかもしれない」と回答した。そのため、今後はまず腕の疲れを定量的に測定するため、Consumed Endurance [10] を使用し実験を行うことを予定している。さらに、適切な操作姿勢 (例えば肘を体に付け、体で肘を支えながら操作をする) を調査し、そのような操作姿勢下においても適切に動作するようにシステムを改良する予定である。また、疲れを感じない時間を調査し設計指

針として提示する予定である。

### 5.5.5 より手法の評価に焦点を当てた実験

5.5.1 節において述べたように今回の実験結果には一部画像処理アルゴリズムに起因するターゲット選択エラーが発生しており、FistPointer の性能評価と画像処理の性能評価が同時に内包された実験結果である可能性がある。よって、ポインタの座標決定のために指にマーカを装着し、またクリック操作のために物理ボタンを用いるなど、画像処理アルゴリズムに起因する問題を最小化した上で、FistPointer の操作性の評価に焦点を当てた実験を行う必要がある。

### 5.5.6 HMD の持ち方による実験結果への影響

実験 1 においては腕の支えによる性能への影響を調査するために HMD を頭部に固定した一方で、実験 2 においてはより現実利用に即した環境においての結果を得るために HMD を頭部に固定せず操作に用いない手にて支えた。実験 1 に比べ HMD を支える手の状態が変化している。これに伴って、実験の経過とともに HMD を支える手に疲労が蓄積し、精度への影響が生じる可能性がある。今回は十分な休息をとりつつ実験を行い、また実タスクの所要時間は 15 分程度であったため HMD の持ち方による実験結果への影響は出なかった (5.4.1 節)。しかしながら、今後は HMD の持ち方による性能への影響を調べるため HMD を頭部に固定した場合と、手に持った場合を実験条件とし、より長時間に及ぶ実験を行い、その影響を調査する必要がある。

### 5.5.7 ポインタの表示位置

3.2 節にて述べたように提案手法は検出された握りこぶしの右上にポインタ位置を割り当てている。これは、検出された握りこぶしの幅を水平方向へ走査していき、最も幅の増加率が大きい箇所を求めることによって実現している。しかしながら、この増加率の大きい箇所は直前 9 箇所の握りこぶしの幅をもとにした回帰直線の傾きのピークによって求められており、カメラが捉える手の形が大きく変動した際 (例えば顔と手の距離に応じてポインタの移動量を調節した場合) に親指の付け根と人差し指の付け根の間どこが特定されるかが不安定になる可能性がある。そのため、ポインタ位置の  $x$  座標が  $y$  座標に比べて不安定

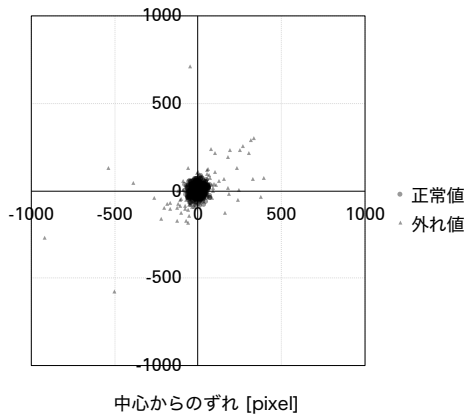


図 16 中心からのずれ.

であることが懸念される。そこで、実験 2 におけるデータを用いてターゲット選択時の  $x$  座標と  $y$  座標の誤差に違いがあるのかどうかを調査した。

図 16 に実験 2 における実験協力者が実験タスク中に行った全ての選択動作のターゲットの中心からのずれを示す。中心からのずれの平均値 [pixel] は  $(x, y, distance) = (22.2, 20.0, 33.1)$ 、標準偏差は  $(x, y, distance) = (69.7, 33.3, 5.9)$  であった。また、この  $x$  軸と  $y$  軸のそれぞれのずれについて、統計分析を行った。等分散性を確かめるために Levene の検定を行ったところ等分散性が保証されなかったため ( $p < 0.05$ )、等分散性が仮定されていない場合の対応なし  $t$  検定を行った。その結果、 $x$  軸および  $y$  軸のずれに有意差はみられなかった ( $t = 1.42$ ,  $df = 3495.8$ ,  $p = 0.155$ )。

さらに、Smirnov-Grubbs 検定を用いて外れ値の除去を行った。除去後の中心からのずれの平均値 [pixel] は  $(x, y, distance) = (15.7, 16.5, 25.1)$ 、標準偏差は  $(x, y, distance) = (12.8, 13.4, 15.1)$  であった。除去後のデータに関しても統計分析を行った。等分散性を確かめるために Levene の検定を行ったところ等分散性が保証されなかったため ( $p < 0.05$ )、等分散性が仮定されていない場合の対応なし  $t$  検定を行った。その結果、 $x$  軸および  $y$  軸のずれに有意差がみられ ( $t = -2.02$ ,  $df = 4718.8$ ,  $p < 0.05$ )、 $y$  軸のずれの方が有意に大きかった。

分析の結果より、 $y$  座標の方が  $x$  座標に比べ不安定

であることが分かった。これは、5.5.3 節で述べたようにクリック時に腕自体が垂直方向に動くことに起因すると思われる。ポイントの表示位置は実験的に最も親指の折り曲げ動作に影響を受けない位置に決定したが、今回の実験データにおいては特に  $x$  軸方向の方が誤差が大きいということにはなかった。しかしながら、今後は顔と手の距離に応じてポイントの移動量を調節する場合などを考慮したより最適な位置を模索したい。

## 6 アプリケーション

### 6.1 シューティングゲーム

FistPointer を用いたアプリケーション例としてシューティングゲームを開発した (図 1)。プレイヤーは手を動かすことによりターゲットスコップを動かすことができ、クリック操作によって弾を打つことができる (ターゲットのサイズ:  $18.0\text{ mm} \times 12.0\text{ mm}$ , large 条件に比べ面積比 3.6 倍)。FistPointer は精度の高いポインティングが行えるため、このようなシューティングゲームにも応用可能である。

作成したシューティングゲームを 9 名のボランティア (男性 8 名、女性 1 名、年齢 21–25 歳、平均 23.6 歳) に体験してもらい、その後感想を求めた。その結果以下のようなフィードバックが得られた。3 名の体験者が「精度が高くターゲットを簡単に狙うことができた」と答え、また 2 名の体験者からは「親指を曲げる操作が弾を撃つ操作と関連付いていて分かりやすかった」と答えた。一方で、3 名の体験者からは「長時間利用すると腕がつかれると思う」という意見が得られ、またその内の 1 名からは「腕を伸ばした方が細かな操作が行えるが、腕を伸ばすことによってより疲れやすくなると感じた。この問題はカメラをより広角にすることにより改善されるのでは」という意見が得られた。

提案手法は腕を目の前に構えるという動作から、ターゲット選択を想起しやすい姿勢であることが実験協力者の意見からも示唆された。また、手を空中において動かすことにより腕に疲労が溜まるため、長時間利用には向かないことが示唆された。よって、長時間の操作を必要としないミニゲームやメニュー選択に提

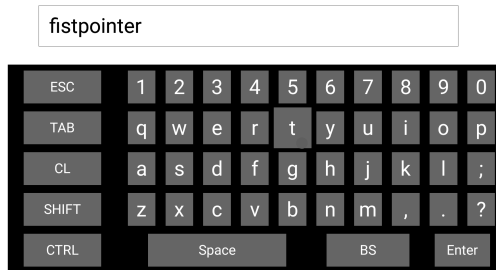


図 17 文字入力アプリケーション。

案手法は適していると考えられる。

## 6.2 文字入力

設計指針に基づいたソフトウェアキーボードを実装した(図 17)。図 11 に示すエラー率の低い領域にキーボードを配置し、エラー率の高い領域に修飾キーを大きなサイズにて配置した。キーボードのキーサイズは通常時は  $6.0\text{ mm} \times 6.5\text{ mm}$ (large 条件に比べ面積比 0.7 倍) であり、ポインタがキーに重なった場合は  $8.1\text{ mm} \times 8.6\text{ mm}$ (large 条件に比べ面積比 1.2 倍) に拡大される。

## 7 今後の課題

### 7.1 手検出アルゴリズムの改良

今回行った実験の目的は、FistPointer によるターゲット選択時間およびエラー率を評価することであった。そのため、今回の実装は実験時の環境のみを想定していた。提案手法を多様な環境で利用するためには、より強固なノイズ除去を実装する必要がある。ノイズを除去し手を検出している研究はいくつかあり[23]、今後はこれらの手法を参考にしながらノイズ除去を実装する。また、現状のアルゴリズムでは本来の肌領域よりも大きなノイズが検出された場合、そのノイズが肌領域として認識される。より高い精度にて手の位置を検出する方法として、手の特徴点を抽出する手法や機械学習を用いた手法があり、今後実装の参考にしたい[13][21][22]。しかしながら、これらの手法は計算コストが高いため、精度とのトレードオフを考慮して適切な方法を検討する必要がある。

### 7.2 決定動作に用いる指の検討

NUI においては広く人差し指を用いたターゲット選択手法が研究されている(例: Microsoft 社 HoloLens の AirTap)。よって、FistPointer において用いている親指を用いたターゲット選択手法と人差し指を用いたターゲット選択手法を実験により比較する必要がある。

### 7.3 奥行きを利用した操作方法の検討および評価

提案手法では顔と手の距離に応じてポインタの移動量を調節することができる。具体的には、ユーザは手をカメラに近づけることにより大まかな操作が素早く行え、手をカメラから遠ざけることにより細かな操作が行える。この特性が選択時間やエラー率にどのような影響を与えるのかを今後調査をする必要がある。

また、この顔と手の距離という奥行き方向を利用したユーザインタフェースの検討を行う。例えば、VR ゲームにおいては手を伸ばした状態における入力であればポインティング、手を手前に引いた状態における入力であればメニュー表示(武器の切り替え等)というインタフェースや、ファーストパーソンシューティングのようなシューティングゲームにおいては奥行き方向の動きを銃のスコップのズームに割り当てるといったインタフェースも考えられる。また、6.2 節に示した文字入力においては、手を伸ばした状態における入力はキー選択、手を手前に引いた状態における入力は変換候補の選択、のようなインタフェースも実現可能である。今後はこのような奥行きを利用した VR 環境における操作を検討していく。

## 8 おわりに

本論文では、カードボード VR 環境における携帯端末の背面空間にて手を動かすことによるターゲット選択手法である FistPointer を示した。本手法は手の位置に応じてポインタを表示し、ユーザは手を動かすことによりポインティングを行った後、親指を曲げることによりターゲット選択を行う。提案手法は外部機器を必要とせず、端末に内蔵されたカメラのみを用いて実装可能である。提案手法の性能を調べるため、ターゲット選択時間およびエラー率を評価する実

験をカードボード VR 環境において行った。その結果, large 条件 (ターゲットサイズ: 7.7 mm) におけるターゲット選択であれば 89.7%の精度にて, medium 条件 (ターゲットサイズ: 3.8 mm) であれば 84.0%の精度にて選択可能であることがわかった。また, 画面の中心領域よりも周辺領域の方が選択しにくいというフィードバックを実験協力者から得た。そこで, ターゲット領域別のエラー率を求めるための実験を行った。その結果, 画面の領域ごとにエラー率の偏りがあることが分かり, 特に画面の最左領域および最上領域のエラー率が有意に高いことが分かった。

ただし, 今回の実験においては実験結果に Fist-Pointer という手法に起因する問題と画像処理アルゴリズムに起因する問題の双方が含まれており, Fist-Pointer に関する純粋な実験結果が得られておらず限定的な結果である可能性がある。よって今後は指にマーカを装着するなどし, 画像処理アルゴリズムに起因する問題を最小化した上で FistPointer の評価を行い, 導き出される結果が同一であるかどうかを調査する予定である。また, 精度向上のためのアルゴリズムの改良や腕の疲れという面からの調査を行う予定である。

## 参考文献

- [1] Baldauf, M., Zambanini, S., Fröhlich, P. and Reichl, P.: Markerless Visual Fingertip Detection for Natural Mobile Device Interaction, in *Proc. MobileHCI '11*, pp. 539–544.
- [2] Brooke, J.: SUS: A Quick and Dirty Usability Scale, *Usability Evaluation in Industry*, Taylor & Francis, 1996.
- [3] Brooke, J.: SUS: A Retrospective, *The Journal of Usability Studies*, Vol. 8, No. 2(2013), pp. 29–40.
- [4] Fukazawa, R., Takashima, K., Shoemaker, G., Kitamura, Y., Itoh, Y. and Kishino, F.: Comparison of Multimodal Interactions in Perspective-corrected Multi-display Environment, in *Proc. 3DUI '10*, pp. 103–110.
- [5] Google: Google design guidelines: Button, <https://material.google.com/components/buttons.html#buttons-style> (2017 月 5 月 25 日閲覧).
- [6] Google: Google VR Cardboard, <https://vr.google.com/cardboard/> (2017 月 5 月 25 日閲覧).
- [7] Groenewald, C., Anslow, C., Islam, J., Rooney, C., Passmore, P. and Wong, W.: Understanding 3D Mid-Air Hand Gestures with Interactive Surfaces and Displays: A Systematic Literature Review, in *Proc. British HCI '16*, pp. 1–13.
- [8] Gugenheimer, J., Döbelstein, D., Winkler, C., Haas, G. and Rukzio, E.: FaceTouch: Enabling Touch Interaction in Display Fixed UIs for Mobile Virtual Reality, in *Proc. UIST '16*, pp. 49–60.
- [9] HACOSCO INC.: スマホ VR のハコスコ, <http://hacosco.com/> (2017 月 5 月 25 日閲覧).
- [10] Hincapié-Ramos, J. D., Guo, X., Moghadasian, P. and Irani, P.: Consumed Endurance: A Metric to Quantify Arm Fatigue of Mid-air Interactions, in *Proc. CHI '14*, pp. 1063–1072.
- [11] Kato, H. and Yanagihara, H.: PACMAN UI: Vision-based Finger Detection for Positioning and Clicking Manipulations, in *Proc. MobileHCI '13*, pp. 464–467.
- [12] Kato, K. and Miyashita, H.: Creating a Mobile Head-mounted Display with Proprietary Controllers for Interactive Virtual Reality Content, in *Proc. UIST '15 Adjunct*, pp. 35–36.
- [13] Kölsch, M. and Turk, M.: Fast 2D Hand Tracking with Flocks of Features and Multi-Cue Integration, in *Proc. CVPR '04*, pp. 158–165.
- [14] Lasinger, K., Baldauf, M. and Zambanini, S.: Evaluating Camera-based Methods for Mobile Markerless Fingertip Detection, in *Proc. Workshop on Smart Mobile Applications, 10th International Conference on Pervasive Computing*, pp. 1–6.
- [15] Lee, J., Kim, B., Suh, B. and Koh, E.: Exploring the Front Touch Interface for Virtual Reality Headsets, in *Proc. CHI EA '16*, pp. 2585–2591.
- [16] Lv, Z., Halawani, A., Lal Khan, M. S., Réhman, S. U. and Li, H.: Finger in Air: Touch-less Interaction on Smartphone, in *Proc. MUM '13*, pp. 16:1–16:4.
- [17] Markussen, A., Jakobsen, M. R. and Hornbæk, K.: Selection-Based Mid-Air Text Entry on Large Displays, in *Proc. INTERACT '13*, Kotzé, P., Marsden, G., Lindgaard, G., Wesson, J. and Winckler, M.(eds.), pp. 401–418.
- [18] Niikura, T., Watanabe, Y., Komuro, T. and Ishikawa, M.: In-Air Finger Motion Interface for Mobile Devices with Vibration Feedback, *IEEE Transactions on Electrical and Electronic Engineering*, Vol. 9, No. 4(2014), pp. 375–383.
- [19] Petry, B. and Huber, J.: Towards Effective Interaction with Omnidirectional Videos Using Immersive Virtual Reality Headsets, in *Proc. AH '15*, pp. 217–218.
- [20] Rupperecht, F. A., Ebert, A., Schneider, A. and Hamann, B.: Virtual Reality Meets Smartwatch: Intuitive, Natural, and Multi-Modal Interaction, in *Proc. CHI EA '17*, pp. 2884–2890.
- [21] Song, J., Pece, F., Sörös, G., Koelle, M. and Hilliges, O.: Joint Estimation of 3D Hand Position and Gestures from Monocular Video for Mobile Interaction, in *Proc. CHI '15*, pp. 3657–3660.
- [22] Song, J., Sörös, G., Pece, F., Fanello, S. R., Izadi, S., Keskin, C. and Hilliges, O.: In-air Ges-

tures Around Unmodified Mobile Devices, in *Proc. UIST '14*, pp.319–329.

- [23] Sugiura, A., Toyoura, M. and Mao, X.: A Natural Click Interface for AR Systems with a Single Camera, in *Proc. GI '14*, pp.67–75.
- [24] Vogel, D. and Balakrishnan, R.: Distant Freehand Pointing and Clicking on Very Large, High Resolution Displays, in *Proc. UIST '05*, pp.33–42.
- [25] Wigdor, D., Forlines, C., Baudisch, P., Barnwell, J. and Shen, C.: Lucid Touch: A See-through Mobile Device, in *Proc. UIST '07*, pp.269–278.
- [26] 築谷喬之, ガースシューメイカー, ブース S. ケログ, 高嶋和毅, 伊藤雄一, 北村喜文, 岸野文郎: 大画面壁面ディスプレイ上での影のメタファを利用したポインティング動作におけるフィッツの法則の検討, *Proc. インタラクション 2010*, pp.317–320.
- [27] 古井陽之助, 前田環貴, 松本誠義: 講義画面の自動連続キャプチャを行う講義・学習支援システムとその評価実験, *Proc. 火の国情報シンポジウム 2014*, pp.1–8.



石井 晃

1992 年生。2016 年筑波大学情報学群情報メディア創成学類卒業。同大学大学院システム情報工学研究科コンピュータサイエンス専攻博士前期

課程在学中。ヒューマンインタフェースに興味を持ち、現在はウェアラブルデバイスにおける入力手法や AR/VR に関する研究を行う。ACM, 情報処理学会各学生会員。



阿達 拓也

1993 年生。2016 年筑波大学情報学群情報メディア創成学類卒業。同大学大学院システム情報工学研究科コンピュータサイエンス専攻博士前期課

程在学中。ヒューマンインタフェースやデータ分析に興味を持ち、現在はプローブデータの分析に関する研究を行う。情報処理学会学生会員。



島 佳吾

1992 年生。2015 年筑波大学情報学群情報メディア創成学類卒業。2017 年同大学大学院システム情報工学研究科コンピュータサイエンス専攻博士

前期課程修了。ヒューマンインタフェースに関する研究を行う。



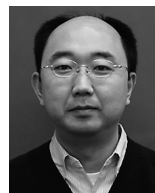
中前 秀太

1993 年生。2016 年茨城工業高等専門学校専攻科産業技術システムデザイン専攻情報工学コース修了。筑波大学大学院システム情報工学研究科コンピュータサイエンス専攻博士前期課程、高度 IT 人材育成のための実践的ソフトウェア開発専修プログラムに所属。ヒューマンインタフェースに関する研究に従事。ACM, 情報処理学会各学生会員。



志築 文太郎

1971 年生。1994 年東京工業大学理学部情報学科卒業。2000 年同大学大学院情報理工学研究科数理・計算科学専攻博士課程単位取得退学。博士(理学)。現在、筑波大学大学院システム情報系准教授。ヒューマンインタフェースに関する研究に興味を持つ。日本ソフトウェア科学会, ACM, IEEE Computer Society, 情報処理学会, 電子情報通信学会, ヒューマンインタフェース学会各会員。



高橋 伸

1991 年東京大学理学部情報科学科卒業。1993 年同大学大学院理学系研究科情報科学専攻修士課程修了。1995 年同博士課程中退。博士(理学)。1995 年東京工業大学大学院情報理工学研究科数理・計算科学専攻助手。2004 年筑波大学大学院システム情報工学研究科コンピュータサイエンス専攻講師。現在、同大学システム情報系准教授。ユーザインタフェース・ユビキタスコンピューティングに興味を持つ。ACM, 情報処理学会, 日本ソフトウェア科学会各会員。