

FistPointer :

モバイル VR 環境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†

概要. 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、モバイル VR 環境）を実現する製品がある。このようなモバイル 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操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、モバイル 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、ターゲット選択速度および精度を評価する実験をモバイル VR 環境において行った。結果、7.7 mm × 7.7 mm のターゲットを 90%の精度にて選択可能であった。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。

1 はじめに

Google Cardboard [3] やハコスコ [4] 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、モバイル VR 環境）を実現する製品がある。このようなモバイル 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、現状ではコンテンツの閲覧が主な VR アプリケーションとなっており、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である 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のみを用いるため、モバイル 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、モバイル VR 環境に対する新たな入力手法となり得る。

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速度および精度を評価する実験を行った。



図 1. FistPointer の概観。

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。本稿では、これらについて報告する。

2 関連研究

提案手法は、携帯端末の背面カメラを用いてユーザの手の動きを認識することによるターゲット選択手法である。よって、ユーザの手および指の動きをカメラにて認識し、端末操作に活用する操作手法が関連する。また、提案手法はモバイル VR 環境向けのターゲット選択手法として適用可能であるため、それらのターゲット選択手法とも関連する。

Copyright is held by the author(s).

* 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

† 筑波大学システム情報系

2.1 カメラを用いた端末操作手法

Baldauf ら [1] 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た。この研究では、クリック操作については述べられていないが、我々はこの研究とは異なる手の姿勢にてポインティングを行い、かつクリック操作も実装している。

Lv ら [8] は検出した指の方向により独自の GUI を操作する手法を示した。本研究は指の方向ではなく、手の 2 次元座標を利用する点において異なる。

Song ら [11] はカメラで検出した空中における手のジェスチャによる操作を提案した。この研究は手のジェスチャを特定の動作の起動および補助に用いているが、我々の研究では手をポインティングに用いている。

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。新倉ら [9] はハイスピードカメラと赤外線 LED を用いて指の追跡を行った。Wigdor ら [13] 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。我々の手法では、携帯端末に内蔵されたカメラのみを用いてポインティングを可能とするため、外部機器は不要である。

2.2 VR 環境向けのターゲット選択手法

HMD に表示されたターゲットの選択などを可能とするために、VR 向けのポインティング手法が提案されてきた。例えば、Web カメラを用いて認識した手によるポインティング手法がある。杉浦ら [12] は、指先で HMD に表示された仮想ボタンを指にてタッチする研究を行った。加藤ら [5] 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開発し、HMD に表示されたターゲット選択において、マウスによる選択の速さと比較する実験を行った。

Lee ら [7] 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、仮想空間へのインタラクションを可能とした。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。Petry ら [10] 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。Petry らの手法は顔と手の距離に応じて操作を切り替えているが、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。

以上の研究は、ユーザの手を認識するために追加の機器を用いているが、提案手法は端末に内蔵されたカメラを用いて手の認識を行う。そのため、携帯端末を用いたモバイル VR 環境に対して容易に適用可能である。

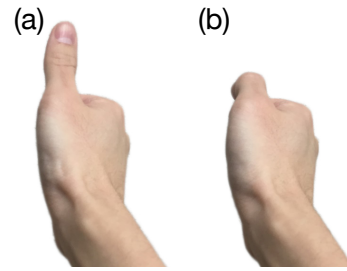


図 2. 提案手法に用いられる手の姿勢. a) 通常時. b) クリック時.

3 提案手法 : FistPointer

FistPointer は、ユーザが図 2a 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 2b 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手法である。端末の背面カメラから得た RGB 画像（今回の実装では 320×240 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。本節にて、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 [11] の式 1 を用いた。

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

$R(u)$, $G(u)$, $B(u)$ はそれぞれ座標 u にある画素の RGB 値であり、 τ 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 τ 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

提案手法ではクリック操作の際にポインタの位置に対する影響を最小限に抑えるために、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。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、まず y_p を先に求め、次に y_p をもとに x_p を求める

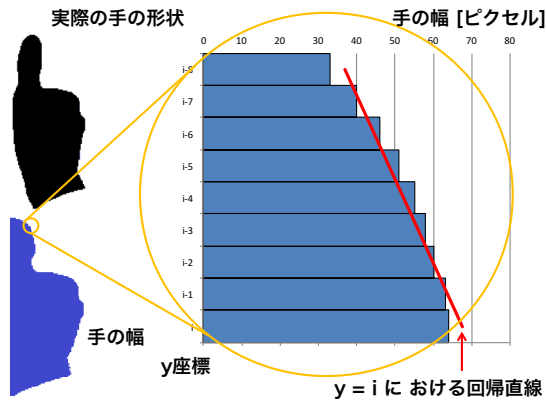


図 3. y 座標 i における回帰直線 (赤線).

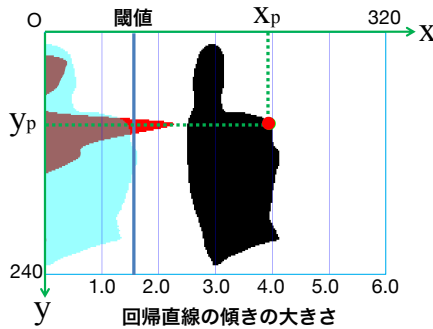


図 4. 検出した手の形状 (黒), 幅 (水色), および回帰直線の傾きの大きさ (赤)

こととした. 最初に, 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る (図 3). この回帰直線の傾きを指先以降のすべての 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畳表示した図を図 4 に示す. この図から, 親指の先および親指の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, 親指の付け根は親指の先と比べても傾きが大きい. そこで, 回帰直線の傾きのピークを見つけ, そこを y_p とした. x_p は y_p において手と認識された画素群のうちの右端の x 座標とした (図 4 に赤い丸で示されている位置).

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なければならない. これ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. そこで, 図 5 の黄色の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 (画面

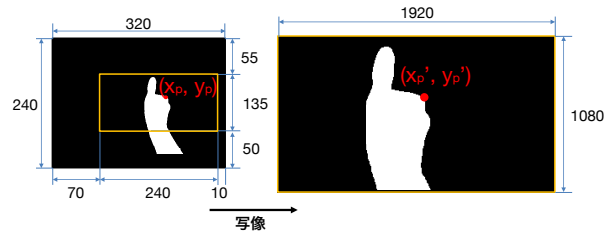


図 5. カメラプレビューと操作領域間のポインタの位置関係.

サイズが 1920×1080 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

親指座標がポインタ座標に近づいたとき, クリック操作が行われたとした. まず, 手として認識された画素のうち, 最も y 座標が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に, 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 ターゲット選択性能評価

モバイル VR 環境における FistPointer のターゲット選択速度および精度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 条件を支え条件として設け, 性能の比較を行った.

4.1 被験者および実験機器

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) を被験者とした. 全ての被験者が日常的にスマートフォンを利用



図 6. 実験において使用した HMD.

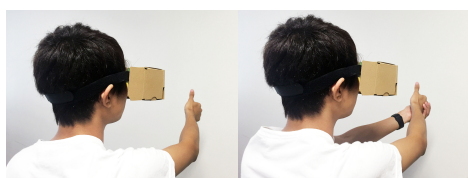


図 7. 左 : non-support 条件. 右 : support 条件.

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。また、すべての被験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たことがなかった。

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, OS: Android 4.2.2) をハコスコ タタミ 1 眼 [4] 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マジックテープ製のバンドおよびスポンジを用いて HMD を頭部に固定できるようにした (図 6)。

4.2 比較手法

FistPointer は片手にて使用可能な操作手法ではあるが、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。そこで、操作に用いる腕をもう一方の腕にて支えることにより、どの程度選択速度および精度に影響を及ぼすかを調査することとした。そのために、操作に用いる腕をもう一方の腕にて支える **support** 条件 (図 7 右), および支えない **non-support** 条件 (図 7 左) を支え条件として設けた。

4.3 ターゲットのサイズ条件

ターゲットの大きさがターゲットの選択速度および精度に及ぼす影響を調査するため、図 8 に示すように 3 種類の大きさのターゲットセットをサイズ条件として用意した。ターゲットは格子内に青で塗られた正方形のセルである。Google が推奨している [2] 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、ターゲットの幅は 48, 24, 12 dp

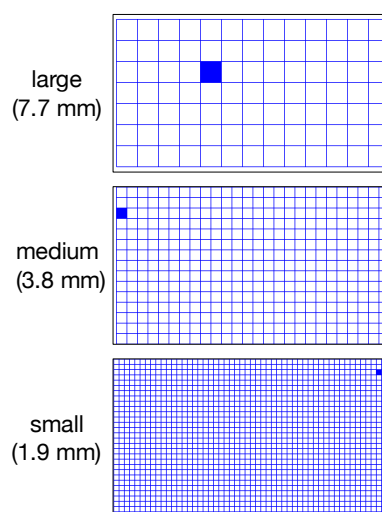


図 8. ターゲットのサイズ条件.

(7.7, 3.8, 1.9 mm) である。以降、それぞれ 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 × 13, 15 × 26, 30 × 53 となる。

4.4 実験手順

実験者はまず、被験者に椅子に深く腰掛けるよう指示した。次に被験者の頭部に 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、被験者に 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。その後被験者には 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、実験者は被験者に、右手を用いて FistPointer の操作を行うよう指示した。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、被験者はハコスコの下部に空いた穴を経由して画面をタップすることによりタスクを開始した。

被験者に行ってもらったタスクは、1 回のランダムな位置に表示されるターゲットの選択を 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、被験者が 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。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。被験者には、手法ごとに large, medium, small 条件順に 1 セッションずつ行ってもらった。よって各被験者は、合計 120 試行 (= 20 試行 × 2 支え条件 × 3 サイズ条件) を行った。また、手法間のカウンタバランスをとるため、被験者を 2 グループに分け、一方

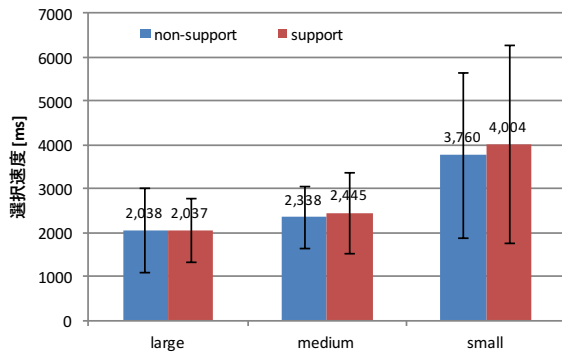


図 9. ターゲットの選択速度.

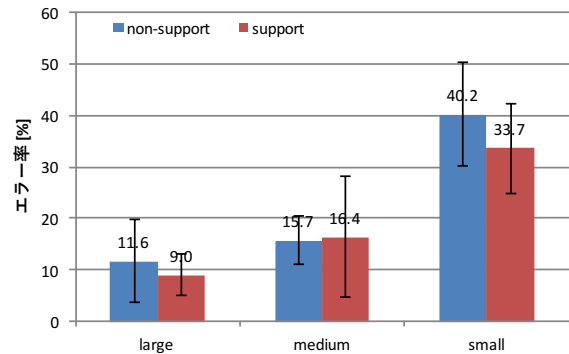


図 10. ターゲット選択時のエラー率.

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、被験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、被験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 分程度の時間を要した.

4.5 実験結果

我々は Kruskal-Wallis の検定により各条件の手法間の有意差を検定した. また、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 $\mu \pm 3 \times \sigma$ (μ : 平均値, σ : 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タッチすること、および FistPointer において被験者の意図しないクリック操作判定があったためである. 全選択回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、エラー率は「エラー回数 / 選択総回数」と定義した.

1 試行に要した時間を図 9 に、エラー率を図 10 に示す. support 条件および non-support 条件間に有意な差は見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% エラー率が低かった.

4.6 考察

実験結果からは support 条件による速度および精度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速度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.

large 条件においては 90% の精度にて選択可能であり、VR コンテンツ内のメニュー選択やシューティングゲームのカーソル操作などに応用が可能である.

またクリック操作の際、親指を曲げる動作に連動し手が微小に上下に動き、それに伴ってポインタが目的のターゲットからずれるといった現象が確認された. 実験中に、被験者の一人から「ポインタをターゲットに合わせたあとクリック操作により手自体がぶれる」という意見も得られた. この問題に対しては、クリック操作の開始を認識した時点でポインタの座標を固定するよう実装を変更するという改良が考えられる. これにより、このずれを抑えることができるため、よりポインティング精度が高まると考えられる.

5 アプリケーション: シューティングゲーム

FistPointer の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 11). プレイヤーは手を動かすことによりターゲットスコー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. 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にも応用可能である.

作成したシューティングゲームを 4 名の大学院生ボランティア (男性 4 名, 年齢 23–25 歳, 平均 23.5 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 名の被験者からは「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 名の被験者からは「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被験者からの意見により示唆された. また、手を空中



図 11. **FistPointer** を用いたシューティングゲーム。

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案手法は適していると考えられる。

6 議論

本研究における実験の目的は、**FistPointer** によるターゲット選択速度および精度を評価することであった。また、今回の実装は実験時の環境のみを想定していたため、多様な環境で利用するためには、より頑強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 [11, 12]、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、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もある [6]。ただしこ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 まとめと今後の課題

本稿では、モバイル 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。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、ターゲット選択速度および精度を評価する実験をモバイル 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）程度のサイズのターゲット選択であれば 90% の精度にて選択可能であり、VR コンテンツ内のメニュー選択などに応用が可能であることが分かった。また、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 条件を支え条件として設け性能の比較を行った。しかしながら、支え条件による性能への影響は確認されなかった。

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。その結果、様々なフィードバックを得ることができた。

今後は、実環境利用に耐えうるようなノイズ除去を実装する。また、今回のモバイル VR 環境実験においては支え条件の支える位置は一箇所のみしか検討していない。そのため、今後は複数箇所を検討しより最適な操作姿勢を特定する必要がある。

参考文献

- [1] Baldauf, Zambanini, Fröhlich, and Reichl. Markerless Visual Fingertip Detection for Natural Mobile Device Interaction. In *Proc. MobileHCI '11*, pp. 539–544. ACM, 2011.
- [2] Google. Google design guidelines: Button. <https://material.google.com/components/buttons.html#buttons-style> (2016 月 9 月 4 日閲覧) .
- [3] Google. Google VR Cardboard. <https://vr.google.com/cardboard/> (2016 月 9 月 4 日閲覧) .
- [4] HACOSCO INC. スマホ VR のハコスコ. <http://hacosco.com/> (2016 月 9 月 4 日閲覧) .
- [5] Kato and Yanagihara. PACMAN UI: Vision-based Finger Detection for Positioning and Clicking Manipulations. In *Proc. MobileHCI '13*, pp. 464–467. ACM, 2013.
- [6] Kölsch and Turk. Fast 2D Hand Tracking with Flocks of Features and Multi-Cue Integration. In *Proc. CVPRW '04*. IEEE Computer Society, 2004.
- [7] Lee, Kim, Suh, and Koh. Exploring the Front Touch Interface for Virtual Reality Headsets. In *Proc. CHI EA '16*, pp. 2585–2591. ACM, 2016.
- [8] Lv, Halawani, Lal Khan, Réhman, and Li. Finger in Air: Touch-less Interaction on Smartphone. In *Proc. MUM '13*, pp. 16:1–16:4. ACM, 2013.
- [9] Niikura, Watanabe, Komuro, and Ishikawa. In-Air finger motion interface for mobile devices with vibration feedback. *IEEJ Transactions on Electrical and Electronic Engineering*, 9(4):375–383, 2014.
- [10] Petry and Huber. Towards Effective Interaction with Omnidirectional Videos Using Immersive Virtual Reality Headsets. In *Proc. AH '15*, pp. 217–218. ACM, 2015.
- [11] Song, Sörös, Pece, Fanello, Izadi, Keskin, and Hilliges. In-air Gestures Around Unmodified Mobile Devices. In *Proc. UIST '14*, pp. 319–329. ACM, 2014.
- [12] Sugiura, Toyoura, and Mao. A Natural Click Interface for AR Systems with a Single Camera. In *Proc. GI '14*, pp. 67–75. Canadian Information Processing Society, 2014.
- [13] Wigdor, Forlines, Baudisch, Barnwell, and Shen. Lucid Touch: A See-through Mobile Device. In *Proc. UIST '07*, pp. 269–278. ACM, 2007.