

# 自己位置推定可能な複数のデバイスを用いた 操作を可能とするインサイドアウト方式の XR用コントローラの提案

服部 圭介<sup>1,a)</sup> 平井 辰典<sup>1,b)</sup>

**概要:** HMDのコントローラは、3D空間内で円滑な操作をするために、現実の手の動きに連動した様々な操作方式が提案されてきた。多くの手法は、赤外線カメラなどのセンサを用いてコントローラの位置を認識することで、3D空間内のオブジェクトを直接手で触れるような直感的な操作である。しかし、コントローラがセンサに認識されなければ、現実の手に連動した直感的な操作ができない。そこで本研究では、複数台の自己位置推定可能なデバイスを用いた操作を可能とするインサイドアウト方式のXR用コントローラを提案する。具体的には、インサイドアウト方式を用いて複数のデバイスを自己位置推定し、それぞれのデバイス間で位置情報を共有する。そして、複数のデバイスの中のものから、XRコンテンツ用のディスプレイとコントローラを、最低各一つずつ扱う。これにより、これまでのHMDのコントローラの欠点であった、センサの死角内ではコントローラを操作できない点をカバーする。

## 1. はじめに

多くのエンジニアや投資機関が、数年以内にVR/AR産業は、多額の収益に達する可能性があるとして予測している。VR/ARを中心に投資を行っている「Digi-Capital」は、AR産業が5年以内に850億ドル～900億ドルの収益に達すると予測している。そして、中でも注目されているのがHMD(ヘッドマウントディスプレイ)型のデバイスである。代表的なHMDとして、VRHMDのOculusシリーズやHTC Vive、ARHMDのHololensやMagic Leapなどが挙げられる。HMDは、頭部にデバイスを装着することで、ハンズフリーの状態になるため、ユーザは両手にコントローラを持ったり、ハンドジェスチャ認識を用いて操作できる。多くのHMDに使われるコントローラは、既存のボタンやレバーのみを用いて操作するコントローラではなく、センサを用いて座標取得を可能とするため、手の動きと連動して操作することができるものである。

センサを用いた座標取得の方式は、大きく「アウトサイドイン方式」と「インサイドアウト方式」の二つに分かれている。アウトサイドイン方式は、赤外線カメラなどの外部の機器を用いて座標取得を行う方式で、多くの場合は、

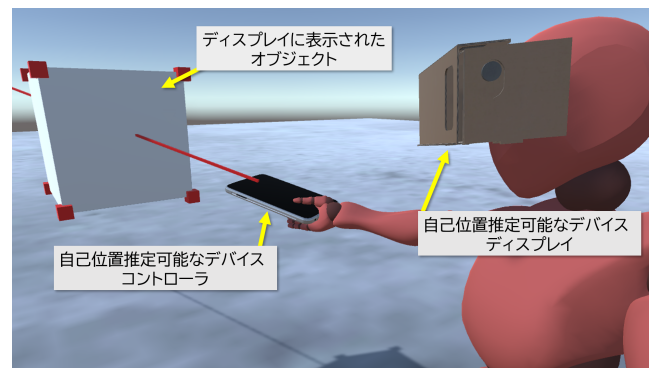


図1 スマートフォンをコントローラとして使用した例

HMDを装着したユーザの周囲にセンサを設置し、ユーザやコントローラの位置を計測している。本稿では、このような単独のデバイスで座標取得を行えず、外部の機器を要する座標取得の方法を座標推定と呼ぶことにする。センサ機器を設置した範囲内で、広範囲にコントローラを動かすことができるのが特徴である。しかし、センサ機器を設置した範囲外にユーザやコントローラが移動してしまうと座標推定が行われなため、ユーザの移動できる範囲が限られてしまっている。一方で、インサイドアウト方式は、外部の機器を使わず、HMDに取り付けられているセンサを用いてユーザやコントローラの位置を計測している。本稿では、このような単独のデバイスで行う座標取得の方法に関しては自己位置推定と呼ぶことにする。外部の機器を

<sup>1</sup> 駒澤大学  
Komazawa University  
a) 3719101k@komazawa-u.ac.jp  
b) thirai@komazawa-u.ac.jp

使わないことから、ユーザは、センサ機器の設置をせすとも手軽に、そして移動範囲に囚われず移動することができる。しかし、コントローラを使用するには、HMDに搭載されているセンサからの認識が必要である。そのためコントローラが、HMDに搭載されているセンサの死角に入るとコントローラの操作ができなくなってしまう。

そこで本稿では、アウトサイドイン方式の限定的な移動範囲でしかコントローラの操作できない点と、インサイドアウト方式のHMDに搭載されたセンサの死角にコントローラが入ってしまうと操作ができないという点の二点を解決したシステムを提案する。具体的には、図1のように、コントローラにカメラを取り付けることで、画像認識によってインサイドアウト方式の自己位置推定を行い、取得した位置情報をHMDと共有させるシステムを提案する。

## 2. 関連研究

XR用デバイスの座標取得の精度は年々増加しており、XR用コントローラも、様々な方式で提案されてきている。本稿におけるXRとは、VRとARを総称したものである。1章で記述したように、XR用デバイスの座標取得の方式は、アウトサイドイン方式とインサイドアウト方式の二つに分かれている。アウトサイドイン方式を使った座標取得方法は、HMDとは別に外部の機器やセンサを用いて定点的な座標推定を行う方式である。この方式では、HMDやコントローラにマークを取り付けたり、外部の機器から発せられる赤外線を受け取ったりして、高精度な座標推定を行っている。インサイドアウト方式で行うHMDの座標取得方法は、HMDに取り付けられているカメラやセンサを用いて周囲情報を取得し、自己位置推定を行っている。また、コントローラの座標取得方法に関しても同様で、HMDに搭載されているカメラやセンサを用いて座標推定を行っているものが多い。これらのことから、アウトサイドイン方式採用のHMDのコントローラの座標推定と、インサイドアウト方式採用のHMDのコントローラの座標取得はどちらも、アウトサイドイン方式でコントローラの座標推定をしていることが分かる。

HMDに搭載されているカメラを用いたコントローラの座標推定は、様々な方式が提案されてきた。代表的なコントローラの座標推定に、手をコントローラとして認識を行うハンドトラッキングがある[1]。ハンドトラッキングは、カメラを用いて手を検出し、直感的な操作を行える。カメラを用いて検出した手のデータを、あらかじめ用意されたハンドジェスチャのパターンと照合することで、手の形や手の位置を認識する方式や、カメラに映し出された手の外郭線を抽出し、外郭線の特徴量もしくはエッジ情報を用いて指先の座標推定を行う方式などがある[2]。また、カメラと画像マークや光源を取り付けたコントローラを使った座標推定も主流である。この手法は、赤外線などの光源をコ

ントローラに取り付けたり、コントローラに特定の画像や模様を貼り付け、画像トラッキングを用いたりしてコントローラの座標推定を行う方式である[3][4][5]。これは、検出したマークや光源をカメラでトラッキングし、パターン認識を用いて座標推定が行っている。これらの手法は、コントローラの座標や角度に連動された操作が可能のため、手の動きに連動した直感的な操作を可能としている。

しかし、ハンドトラッキングやパターン認識を用いた方式の座標取得方法は、HMDに搭載されているカメラの画角にコントローラを取めなければならない。そのため、HMDに搭載されているカメラの死角に、コントローラを移動させてしまうと座標推定ができない。したがって、死角を減らすために、HMDに取り付けるセンサやカメラの数を増やすか、コントローラの操作範囲を限定させることで問題を解決している。しかし、HMDに取り付けたセンサやカメラの数だけコストが掛かってしまう。

そこで、センサやカメラの搭載数を増やさず、またコントローラの操作範囲を限定させることなく、コントローラで操作をさせる手法がこれまでも研究されてきた。Ramosらは、GyroWand[6]というコントローラに慣性計測装置を取り付けることで、HMDにセンサやカメラを取り付けずとも操作可能なコントローラを提案した。これは、慣性計測装置によってコントローラの角度を取得し、装着しているHMDにコントローラの角度の情報を送ることで操作を可能としている。そして、この手法では、HMDや体の一部から発せられているレイキャストの角度を操作してオブジェクトを指すことができる。しかし、慣性計測装置による座標取得方式では、位置情報にズレが蓄積してしまうため、コントローラの正確な自己位置推定はできない。そのため、コントローラの座標に合わせた操作は困難である。Roらは、AR Pointer[7]という慣性計測装置を搭載したデバイスとジェスチャ認識を用いたHMD用のコントローラを提案した。これは、慣性計測装置を用いてコントローラの角度を取得すると共に、身体ジェスチャ認識が可能な外部センサであるKinectを用いて手の座標を取得することでコントローラの位置情報を取得している。そして、Kinectを用いて得たコントローラの位置情報と、慣性計測装置を用いた角度情報をHMDに送り、コントローラの位置情報から発せられているレイキャストを操作することができる。しかし、この手法はアウトサイドイン方式のHMDのように、事前にKinectを設置しコントローラの座標を取得しているため、コントローラの操作範囲に制限ができてしまう。Mohrらは、TrackCap[8]という慣性計測装置と、画像トラッキングの二つを用いたHMD用のコントローラとして扱う手法を提案した。これは、慣性計測装置を用いた自己位置推定に加え、HMDデバイスの上部に、事前に登録された画像を印刷したバイザーを取り付け、スマートフォンのフロントカメラでバイザーに印刷した画像を認識する

ことで、コントローラの自己位置推定を行っている。画像トラッキングによって取得したコントローラの位置と角度の情報を HMD に送ることで、コントローラの先から発されるレイキャストを操作している。しかし、この手法ではバイザーに印刷されている画像がスマートフォンのカメラの死角に入り、画像を読み込めなければコントローラの操作の精度が下がってしまう。これらの手法は、コントローラにセンサやカメラを取り付けることで、コントローラの操作範囲を広げることができたが、まだ限定的な操作範囲であったり、操作の精度は足りていない。

そこで本稿では、自己位置推定可能な複数のデバイスをインサイドアウト方式を用いて XR 用コントローラとして扱うシステムを提案する。具体的には、画像認識を用いて自己位置推定ができる AR デバイスを二台用意し、片方を XR コンテンツのディスプレイ用デバイスとして使い、もう片方をコントローラ用デバイスとして扱う。画像認識を用いて共通のアンカーを AR 空間内に設置し、双方のデバイスをネットワーク通信させ、座標の同期を行い距離を測り、ディスプレイ用デバイスにコントローラの座標を同期させる。これにより、操作範囲を限定することなく、操作の精度を高めたコントロールシステムを目指す。

### 3. 提案システム

本稿で提案する XR 用コントローラは、画像認識によって自己位置推定を行う二台のデバイスをインサイドアウト方式を用いて座標取得し、ネットワーク通信をする。本稿における画像認識方法は、Google が提供している SDK である ARCore における水平検出システムである「Plane Finding」を使用している。次に、自己位置推定によって取得した座標を双方のデバイス間で同期し、デバイス間の距離を取得する。そして、片方のデバイスをディスプレイとして使い、もう一方のデバイスを XR 用コントローラとして扱うというシステムとなっている。また、本稿における各デバイスの座標の同期方法は、Google が提供している SDK の ARCore の機能である「Cloud Anchor」を使用している。Cloud Anchor は、画像認識を用いて AR 空間内で座標の起点を作ることによって、空間内の座標を同期する事が出来る。この機能を使用し、双方のデバイス間の距離を測ったり、デバイスの角度を検出したりしている。以下に提案システムの詳細を述べる。

#### 3.1 システムの詳細

本稿では、二台のスマートフォンを用いており、一台を HMD として見立て、もう一台をコントローラとして使った。一般的なインサイドアウト方式を用いた HMD と同様な自己位置推定機能を揃えるために、HMD のデバイスとして ARCore 対応のスマートフォンを採用した。本システムにおける座標の同期の手順を説明する。まず、コント

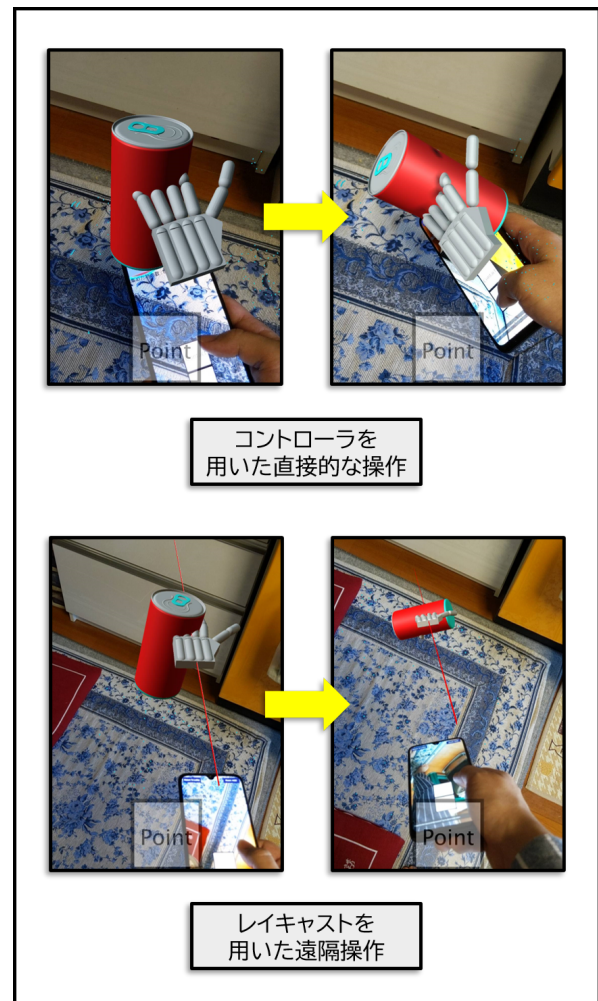


図 2 本システムを用いた操作 上:直接的な操作でオブジェクトを掴んでいる。下:レイキャストを用いてオブジェクトを操作している。

ローラとして使うデバイスのアプリケーション内に用意された通信待機画面からルームを作成する。これは、座標同期のためのバーチャル空間を作り、任意の空間を共有させるために必要な手順である。そして、デバイスに付属しているカメラを用いて周囲の状態を検出し、デバイスの自己位置推定をする。その後ユーザは、空間を共有するために任意の座標を中心としたアンカーを登録する。次にユーザは、ディスプレイとして使うデバイス上の通信待機画面で、作成したルームに参加する。前工程と同じようにディスプレイとして使うデバイスでも自己位置推定を行う。事前に登録されたアンカーの座標にカメラを向けることで双方のデバイスの座標の同期を行うことができる。

本システムの操作方法としては、デバイスを最低一台は片手で保持し、コントローラとして扱う。また、コントローラとして扱うデバイスは、一般的なテレビに使用されるリモコンのように保持し操作を行う。そして、ディスプレイとして使用するデバイスの画面を通して CG オブジェクトを確認する。ディスプレイとして使用するデバイスの画面上には、コントローラとして扱うデバイスの座標を視

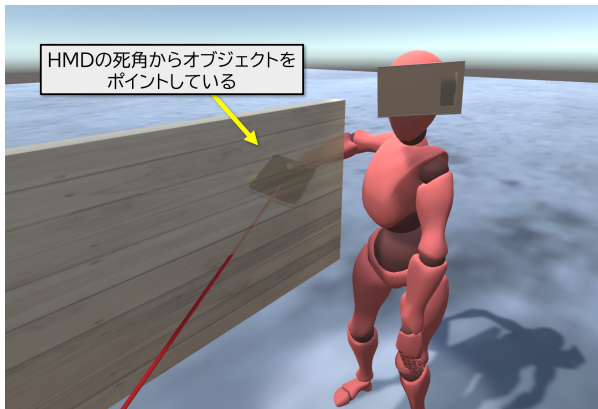


図 3 本システムの特徴的な操作方法

覚的にわかりやすくするため、同期した座標に図 2 (上) のような手を模した CG のオブジェクトを表示させ、コントローラが向いている方向にレイキャストを模した CG の赤い線を伸ばしている。また、双方のデバイスの座標の同期は一秒間に 30 回行われており、表示される CG のオブジェクトはスムーズにコントローラに追従しているように見える。

さらに、実際に同期したコントローラの座標とユーザの操作位置とのギャップが生じることを想定して、座標修正用の機能も備えてある。アプリケーション上では、「Fix ボタン」として用意されており、座標同期時に生まれた誤差をユーザが適当な位置に修正することができる。「Fix ボタン」を押している間は、ディスプレイ用デバイスに表示されている CG のオブジェクトは AR 空間内に固定され、コントローラに追従されなくなる。そして、このボタンを離れた際のコントローラの座標と CG のオブジェクトとの距離を保存し差を作る。その上で、CG のオブジェクトを差の数値だけずらし表示させることで、同期したコントローラの座標とユーザの操作位置とのギャップを無くすることができる。

加えて図 2 (下) のように、多くの既存 XR コンテンツのコントローラの機能にもある、遠隔的に操作する機能も備わっている。この機能は、ユーザが「遠隔モード切替ボタン」を押すことで使うことができる。コントローラ用デバイスの座標からコントローラが向いている方向へ常にレイキャストが発せられており、レイキャストの先に AR 空間内のオブジェクトがある場合、レイキャストとオブジェクトのぶつかった座標を検知することができる。

多くの既存 XR コンテンツのコントローラは、二つでひと組となっており、両手で持つことが想定される作りとなっている。本稿で実装したシステムでは、一台のディスプレイ用の AR デバイスに対して、一つのコントローラとなっているが、コントローラ用のスマートフォンの数を二台にすることもできる。

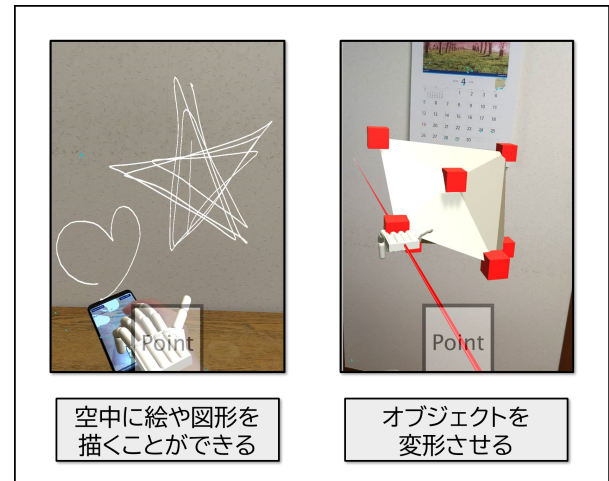


図 4 本システムを応用したコンテンツの例

### 3.2 応用例

本提案システムでしか実現困難な操作例や、本提案システムを用いることが可能なくつかの XR コンテンツの例を紹介する。

まず、アウトサイドイン方式が用いて座標推定されたコントローラは、センサ機器の死角にコントローラが移動した場合操作を行えない。本提案システムでは、コントローラデバイスの認識をインサイドアウト方式を用いているため死角が存在しないので、限らない範囲で操作を行える。図 3 のように、XR デバイスとコントローラデバイス間に壁があったとしても操作が可能な点が本システムの大きなメリットである。

本提案システムを用いることが可能なくつかの XR コンテンツの例の一つに、図 4 (左) にある、既存コンテンツにもある空間上で行う 3D ペイントアプリがある。3D ペイントアプリは、一般的なペイントツールのように平面的に絵を描くのではなく、HMD やコントローラの自己位置推定を利用してコントローラをペンに見立て、三次元的な絵を描くことができるアプリである。本提案システムであっても既存の 3D ペイントアプリと同様の操作ができる。

また、図 4 (右) にある、XR 空間内で行う 3D モデリングツールの実装も行っている。多くの 3D モデリングツールは、アート作成や設計、試作品の制作向けに開発されている。一般的なモデリングツールはマウスとキーボードのショートカットキーを用いて操作するのに対し、XR 空間内で行う 3D モデリングツールは、座標取得されたコントローラの位置から操作できるので、より直感的なモデリングが可能となっている。

## 4. システムの評価

本提案手法の XR 用コントローラの有効性を測るため評価を行った。この評価における有効性とは、以下の一般的な XR コンテンツにおける操作を可能であるかと定義する。

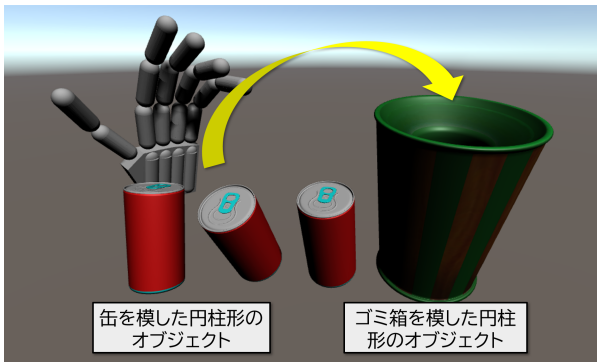


図 5 本評価実験で用いたオブジェクト。缶に模したオブジェクト (左) をゴミ箱に模したオブジェクト (右) に入れるタスク。

- 掴む：オブジェクトを掴み、移動させることができる。
- 回転させる：オブジェクトを任意の角度に回転させることができる。
- 遠隔操作：遠くに配置されたオブジェクトを、レイキャスティングを用いて選択することができる。

上記の3つの操作を必要とした簡単なタスクを用意しXR用コントローラとしての有効性を評価する。本評価実験は、4名にそれぞれ本システムの説明を行った後、実際にテストプレイとして数分間使用してもらい、以下の評価をしてもらった。本評価実験は、インサイドアウト方式で自己位置推定可能なスマートフォンである「Google Pixel3」をコントローラとして使用し、「Xiaomi Mi A3」をデバイスとして二台を使用した。

#### 4.1 評価条件

被験者には、上述の3つの操作を必要とする、特定のオブジェクトを特定の場所まで移動させるタスクを課した。これは、15個のランダムな場所に配置された円柱形のオブジェクトをコントローラを用いて、特定のオブジェクトの場所まで移動させるタスクである。この評価内容をわかりやすくするため、ゴミ拾いゲームとして説明し、図5のようなテキストをオブジェクトに貼り付けた。このタスクには、円柱形のオブジェクトを掴み、特定のオブジェクトまで移動させ、回転させつつ、再配置する、という要素が組み込まれている。また、遠くにある円柱形のオブジェクトはレイキャスティングを用いて掴むことを必要とする。このタスクにより、前述で定義した有効性を評価する。

#### 4.2 評価結果

被験者4人に対して1人ずつ評価した所、被験者全員が、4.1節で記したタスクを達成することができた。被験者4名が全ての15個の円柱形のオブジェクトを特定のオブジェクトへ移動させるまでの平均時間は、118.75秒だった。被験者個別のタスク完了までにかかった時間は、被験者Aは95秒、被験者Bは140秒、被験者Cは100秒、被験者Dは140秒であった。

#### 4.3 考察

4.2節の評価の結果から、本提案手法が、本稿で定義したXR用コントローラとしての有効性があることがわかった。また、被験者の意見にディスプレイ用のデバイスから見たコントローラの位置と実際の位置がズレているため操作が困難であることが挙げられた。加えて、オブジェクトをコントローラで掴んでいる感覚が無いため、バイブレーションの機能を使い、掴んでいる感覚をわかりやすくして欲しいという意見も得た。また、今後の評価では、一般のXRコンテンツに使われている操作方法である、ハンドジェスチャ認識や赤外線センサを用いたコントローラなどとの比較検証をしたいと考えている。加えて、2章で紹介した既存研究であるTrackCap[8]が、本稿の提案手法と近いことから、TrackCapの論文で行われた実験とも比較を行い、操作性について評価をしたいと考えている。

#### 5. まとめ

本稿では、自己位置推定可能なデバイスを用いた操作を可能とするインサイドアウト方式のXR用コントローラを提案した。これまでのHMD用のコントローラの操作方法では、HMDのデバイス付属の外部の機器であるセンサやカメラを用いてコントローラを認識し操作を可能とするものや、HMDに搭載されているセンサやカメラを用いてコントローラを認識し操作を可能とするものがあつた。しかし、外部の機器を設置するコストやコントローラの操作範囲が限定的になってしまっているといった問題点があつた。そこで、自己位置推定可能なデバイスを複数台使用し、最低一台をコントローラとして使用した。それにより、包括的に自己位置推定を行うことができ、より包括的な状況でのコントローラの操作を可能にした。この手法では、既存手法にある死角がなく、HMDとコントローラに物理的な障害物があつたとしても操作が可能である。

また、本提案手法の有効性を検証するために、評価を行った。この評価実験は、XRコントローラの一般的な機能を定義し、それらの機能を達成出来るかを確かめるための評価内容となっている。評価実験を行った結果、本提案手法のコントローラがXR用のコントローラとして有効性があることがわかった。

また、2章で紹介した既存研究であるTrackCap[8]の論文中には、事前に登録した画像をバイザーに印刷し、スマートフォンのインカメラで認識することで、自己位置推定を行っている。この研究の実験の一つにTrackCapのシステムと、当時使われていた画像認識による自己位置推定を実現するSDKであるTangoの二つのシステムを用いたコントローラの操作実験を行ったと記されている。この実験では、激しいコントローラの動きをした際に、Tangoを用いた画像認識のみでは自己位置推定にズレが生じてしまうと記されている。そして、激しいコントローラの動きによつ

て生じたズレを、自動で再定位するためにHMD上部に画像トラッキング用の画像を取り付けたと記されている。しかし、本手法には、激しいコントローラの動きの後に自己位置推定が不可能になっても、ユーザは任意の位置にコントローラの座標を、再定位のための画像を用意せずとも、手で修正できる機能が備わっている。そして、本手法で使ったSDKであるARCoreは、Tangoの後進のシステムであり、激しいコントローラの動きによって生じたズレをある程度補正する機能が備わっている。また、ARCoreには、Deep Diveという既存の画像認識より精度の高い画像認識機能がリリースされることがGoogleから発表された。また、iPad Proに搭載されたLiDARスキャナも同様で、精度の高い画像認識を使用することができる。つまり、これらの最新の技術を用いた画像認識を使用すれば、自己位置推定のみでも激しいコントローラの動きに対応できる可能性が高いため操作性を十分発揮できることが予想される。

本研究の今後の課題として、本手法と既存のHMDのコントローラ操作手法の比較を行うことが挙げられる。本手法は、HMD用のコントローラを想定しているため、既存の手法と同程度かそれ以上の機能を有しているかを検証したい。これは、VR・AR・MR問わずこの手法を用いることで操作が可能であることが証明されれば、コントローラの座標推定のための多くのセンサやカメラにコストをかけたHMDを用意せずとも、コントローラにセンサを取り付けるだけで座標取得できるため、比較的低コストでHMDコンテンツを使用することができるからである。

## 参考文献

- [1] 土屋太二, 高橋和彦: ハンドジェスチャによる拡張現実感システムの構築, 同志社大学理工学研究報告, Vol. 50, No. 3, pp. 107-113 (オンライン), DOI: info:doi/10.14988/pa.2017.0000011846 (2009).
- [2] 絵美玉城, 崇味八木, 純一暦本: インタラクティブシステムのための高精度な3次元ハンドジェスチャ認識手法 (特集 インタラクティブの基盤技術、デザインおよび応用) - (画像情報), 情報処理学会論文誌 論文誌ジャーナル, Vol. 51, No. 2, pp. 229-239 (オンライン), (<https://ci.nii.ac.jp/naid/40019545338/>) (2010).
- [3] Takegawa, Y., Tsukamoto, M. and Terada, T.: Design and Implementation of a Piano Practice Support System using a Real-Time Fingering Recognition Technique, *ICMC* (2011).
- [4] Piekarski, W. and Thomas, B. H.: Using ARToolKit for 3D hand position tracking in mobile outdoor environments, *The First IEEE International Workshop Augmented Reality Toolkit*, pp. 2 pp.- (2002).
- [5] Song, H., Benko, H., Guimbretiere, F., Izadi, S., Cao, X. and Hinckley, K.: Grips and gestures on a multi-touch pen, *Conference on Human Factors in Computing Systems - Proceedings*, pp. 1323-1332 (online), DOI: 10.1145/1978942.1979138 (2011).
- [6] Hincapié-Ramos, J. D., Özacar, K., Irani, P. and Kitamura, Y.: GyroWand: IMU-based Raycasting for Augmented Reality Head-Mounted Displays, (online), DOI: 10.1145/2788940.2788947 (2015).
- [7] Ro, H., Byun, J., Park, Y., Lee, N. and Han, T.: AR pointer: Advanced ray-casting interface using laser pointer metaphor for object manipulation in 3D augmented reality environment, *Applied Sciences (Switzerland)*, Vol. 9, No. 15 (online), DOI: 10.3390/app9153078 (2019).
- [8] Mohr-Ziak, P., Tatzgern, M., Langlotz, T., Lang, A., Schmalstieg, D. and Kalkofen, D.: TrackCap: Enabling Smartphones for 3D Interaction on Mobile Head-Mounted Displays, *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (2019).