

平成 29 年度卒業論文

動きを基準とした AR の検討

宮崎大学 工学部 情報システム工学科

前田 祐希

指導教員 椋木雅之

目次

1. はじめに	1
2. ARにおける位置合わせ	2
2.1. 位置を基準とした AR	2
2.2. 従来の AR での動き	3
2.3. 動きを基準とした AR	5
3. 動きを基準とした AR システムの試作	6
3.1. 試作システムの概要	6
3.2. システムの構成要素	7
3.3. システム構成	8
3.4. 検出する動きの定義	9
3.5. 動きの検出	12
3.6. 動きに応じた仮想物体合成	14
4. 実行結果	15
5. 評価実験	17
5.1. 実験の概要	17
5.2. 評価方法	17
5.3. 実験結果	18
5.4. 考察	19
6. おわりに	20

1. はじめに

近年、拡張現実感（Augmented Reality：AR）技術を使ったアプリケーションが次々と開発されており、その技術は私たちの目に見える形で社会に貢献している。ARとは、現実空間に仮想空間の物体や情報を合成することで現実の感覚を増強することである[1]。AR分野では、現実空間と仮想空間を違和感なく合成することが重要である。そのために、現実空間と仮想空間の間で、幾何学的整合性、光学的整合性、時間的整合性の3つの整合性を考える必要がある。

幾何学的整合性は現実世界と仮想世界の3次元的な位置合わせ、光学的整合性は、現実世界の物体（現実物体）と仮想世界の物体（仮想物体）の陰影や画質の整合性、時間的整合性は、現実世界の動きに対して仮想物体の描画遅れがないことを意味する。

従来のARは、現実の物体の位置を基準に位置合わせをして幾何学的・時間的整合性をとっていた。この際、現実物体の動きは整合性を乱す要因として扱われていた。

これに対して、本研究では動きを基準に位置合わせを行い、動きに整合したARの実現について検討する。

2. AR における位置合わせ

2.1. 位置を基準とした AR

AR における最も基本的な問題は、現実世界と仮想世界の幾何学的位置合わせである。幾何学的位置合わせを行うソフトウェアツールに、ARToolKit[2]があり、多くの AR システムや AR 研究に利用されている (図 1)。ARToolKit はマーカ位置を基準に現実空間と仮想空間を合成している。これを利用する多くの AR システムは位置を基準とした AR であった。

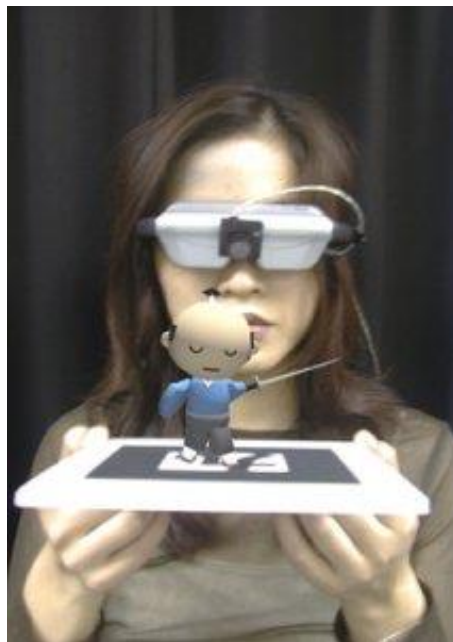


図 1 ARToolKit による AR システムの例

(出典)ARToolKit, <https://www.hitl.washington.edu/artoolkit/>,
2018 年 2 月 12 日アクセス

2.2. 従来の AR での動き

従来の AR でも動きを扱うものはあった。それらは主に、動きに追従するタイプと動き (ジェスチャ) を指令 (コマンド) 入力として使うタイプに分けられる。

動きに追従する例として、梶山ら[3]のゲームエクスペリエンス拡張システムがある (図 2)。この研究では、オフラインゲームを行っているユーザの上半身の骨格の位置・向きを検出し、ユーザの向きと位置に合わせた映像を生成し表示する。検出と表示を繰り返し行うことで、ユーザの動きに追従できるようにしている。

このようなシステムでは、各瞬間での現実物体の位置を検出し、検出位置に整合した仮想物体を時間遅れなく合成することで、現実物体の動きに追従していた。しかし、各瞬間の位置情報を利用するのみで、動きの解析は行っていない。その意味で従来の位置を基準とした AR の一種といえる。

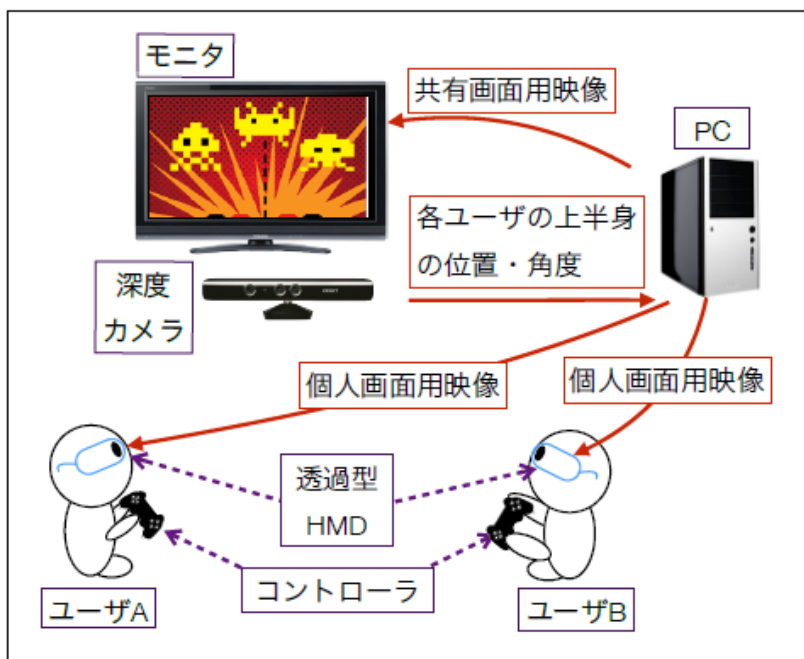


図 2 ゲームエクスペリエンス拡張システム全体図
(出典)[3]より引用

一方、ジェスチャをコマンド入力として使う例として、森田ら[4]の指動作認識を利用したユーザインターフェイスがある（図 3）。この研究では、CMOS センサをつけたスマートグラスを装着し、前方で指を動かすことによって AR 情報の操作を行っている。指には、赤外線 LED を装着しており、赤外線 LED の位置を観測することで、操作を実現している。

このようなシステムでは、動きの種類を解析し、種類に応じて現実世界や仮想世界の物体や情報を操作していた。ここでの動きは、ユーザインターフェイスの入力手段であり、動きの種類を検出することが主な目的であった。入力結果として実行される操作は、動きそのものとは無関係であり、動きの方向、速さ、タイミングに整合して、現実空間と仮想空間を合成することもなかった。

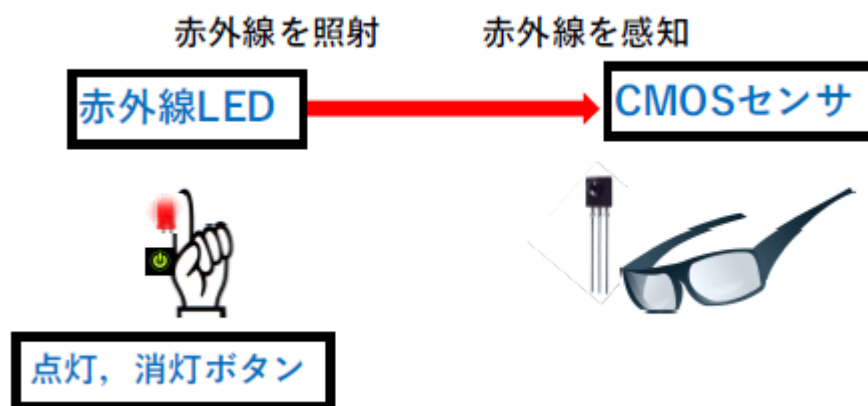


図 3 指動作認識を利用したユーザインターフェイス全体図
(出典)[4]より引用

2.3. 動きを基準とした AR

本研究では、現実世界の動きを拡張する AR を実現するために動きを基準とした AR について検討する。ここで、動きを基準とした AR を、

- ・動き（ジェスチャ）を認識して、仮想空間の反応を生成
- ・動きの方向・速さ・タイミングに整合して仮想空間を合成

するものと定義する。動きを基準とした AR により、例えば、人の動作を強調する視覚効果を付加した映像生成や、特定の動きの方向・速さに反応するインタラクティブなゲーム等が実現できる。

動きを基準とした AR の例として、meleap 社の HADO[5]が挙げられる（図 4）。HADO はヘッドマウントディスプレイとアームセンサを装着し、手から操り出した「エネルギーボール」を相手プレイヤーにぶつけ得点を競い合う AR スポーツである。本研究では、このような AR システムを試作し、動きを基準とした AR の構成要素、システム構成、要素技術等について検討する。



図 4 HADO の例(Player vs Player)
(出典)[5]より引用

3. 動きを基準とした AR システムの試作

3.1. 試作システムの概要

本研究では、右手を後ろから前に突き出す動作に対して、突き出したタイミングに合わせて右手位置から動きの方向に仮想物体を合成する AR システムを試作する。この試作システムは、定義した動きを検出し、動きの方向、タイミングに整合して仮想物体を合成するという、動きを基準とした AR の要素を含んでいる。なお、試作システムのプロトタイプは、2017年11月19日に実施された宮崎大学アドベンチャー工学部で展示し、一般来場者に体験してもらった（図5）。

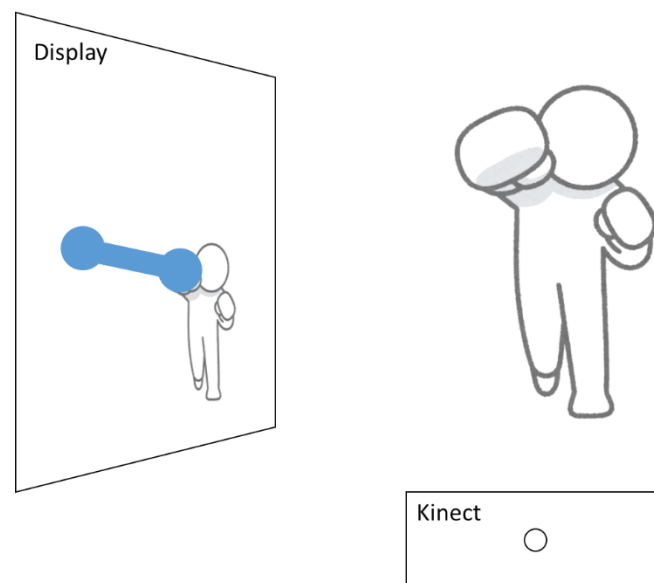


図 5 体験してもらったプロトタイプ

3.2. システムの構成要素

動きを基準とした AR の実現には、以下の要素が必要である。

- ・ 検出する動きの定義
- ・ 動きの検出
- ・ 動きに応じた仮想物体合成

本研究で扱う「動き」は、位置の系列または位置の系列から得られる位置の差分である。

動きの定義では、何らかの座標系を設定し、その座標系での位置の系列を扱う必要がある。実世界における対象の位置や向きに関わらず動きを定義する場合、動きを対象に対して相対的に定義する必要がある。従って、この座標系を対象に固定して設定するのが合理的である。この座標系を対象物座標系と呼ぶ。

動きの検出では、何らかのセンサを使って対象の位置の系列を測定し、定義した動きと照合・判定する必要がある。位置の測定は、センサに依存した座標系（センサ座標系）上で行われる。得られた測定結果を動きの定義と照合するためには、センサ座標系と対象物座標系を対応付ける座標変換が必要である。

仮想物体の合成表示には、ヘッドマウントディスプレイ、携帯端末、固定のディスプレイなどが用いられる。いずれの場合でも、表示装置の視点に合わせて座標系（表示座標系）を設定し、表示座標系上での対象の動きに整合して、仮想物体を描画する必要がある。即ち、センサ座標系から表示座標系への座標変換が必要である。

このように、動きを基準とした AR の実現には、上記のそれぞれの要素を実装し、要素間を適切な座標変換で対応付けて、整合性を保つことが重要である。

3.3. システム構成

試作システムでは、動きの検出に Kinect v2[6] と OpenNI2[7]+Nite2[8]を、合成表示に固定ディスプレイを用いる (図 6)。

Kinect は、Microsoft 社が開発した 3 次元スキャナであり、RGB 値と Depth 値が取得できる RGBD カメラの一種である。ライブラリである OpenNI2+Nite2 と組み合わせることで、人体の姿勢を容易に取得できる。人体の姿勢は、頭、首、左右の手、左右の肩などに相当する 15 個のノードの 3 次元座標とノード間の接続関係によって表現された骨格モデルとして得られる。この 3 次元座標は Kinect に固定された座標系で表現されている。

仮想物体は、Kinect で撮影した現実空間の映像の上に合成して表現する。この場合、視点の位置は Kinect を設置した位置になる。

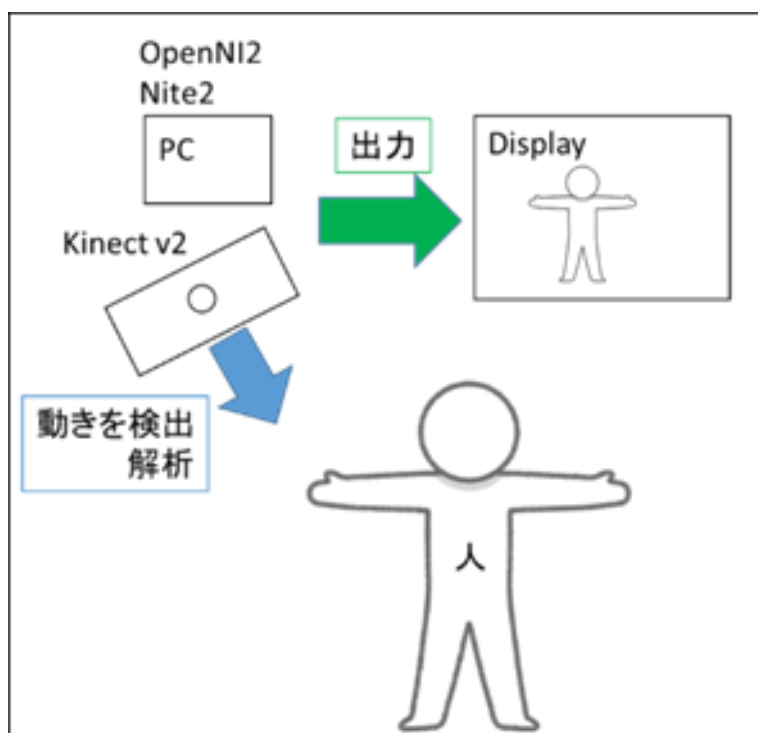


図 6 システム全体図

3.4. 検出する動きの定義

今回検出する動きは、右手を後ろから前に突き出す動作である。この動作を、右手を引いた姿勢（図 7）、右手を突き出した姿勢（図 8）の二つの姿勢に分ける。この二つの姿勢を合わせることで手を後ろから前に突き出す動作として認識する。

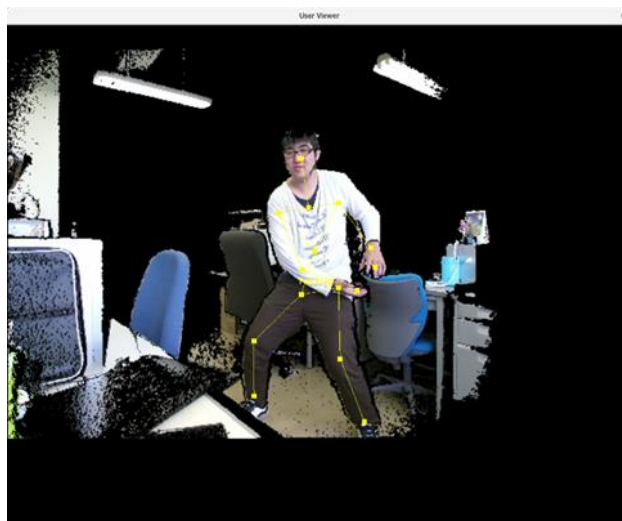


図 7 右手を引いた姿勢

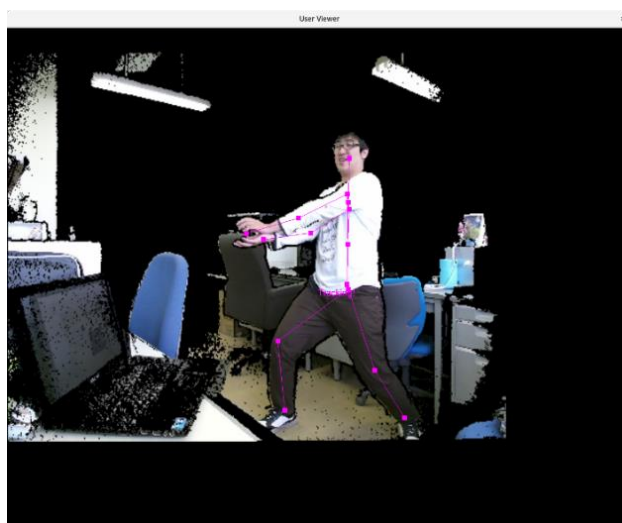


図 8 右手を突き出した姿勢

動作の認識には Kinect の胴体、右肩、右腰、右手、左手の各ノード (図 9) を使う。また、右手と右ひじの距離を L として、判定に用いる。

センサ座標系 (図 10 右) は、Kinect の中心を原点として、Kinect を正面から見たときに右側の Xbox マークがある方向を X 軸の正の方向、Kinect を水平に置いたときの垂直方向を Y 軸の正の方向、Kinect の視線方向を Z 軸の正の方向となっている。

対象物座標系 (図 10 左) は、胴体座標を原点 O として、左手側を X 軸の正の方向、センサ座標系と同じ方向を Y 軸の正の方向、カメラがある方向を Z 軸の正の方向とする。二つの座標系の対応軸関係は図 10 となっている。

右手を引いた状態を、右手が右腰と右肩の高さの間で、胴体の右側にあると定義する。この際、右手を引くときに左手がついてくるため、左手が胴体の右側にあるという補則を設ける。右手を引いた状態を検出できたら、その右手の座標を H とおく。

右手を突き出した状態は、座標 H から右手の座標の距離が、 L の半分の距離を超えたときと定義する。

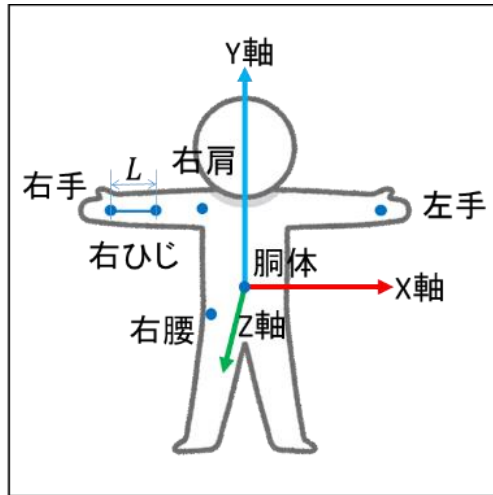


図 9 各ノード位置と軸

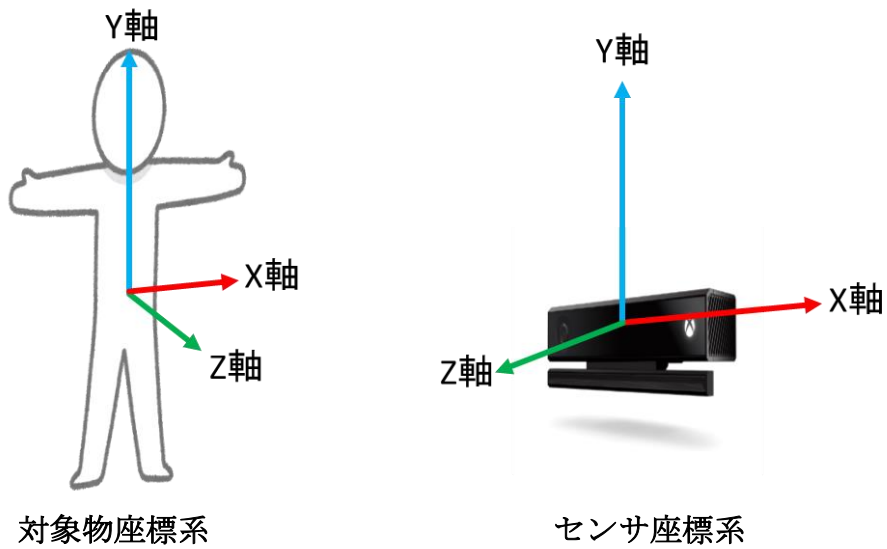


図 10 座標系の軸関係

3.5. 動きの検出

まず、Kinect で検出した各ノードの 3 次元座標を対象物座標系に変換する。
座標変換には、平行移動と回転を合わせたものを使う。

平行移動の式

$$\vec{X}' = \vec{X} + \vec{t}$$

回転の式

$$\vec{X}' = R\vec{X}$$

ここで、 \vec{t} は、移動量 \vec{t} 、 R は、回転行列 R と呼ぶ。

3.4 節で示したとおり、センサ座標系の Y 軸方向と対象物座標系の Y 軸方向は同じ垂直方向のため、回転行列 R は、Y 軸周りの回転となり、

$$R = \begin{pmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{pmatrix}$$

と、表せる。平行移動の式と回転の式を合わせると変換行列 M は次のように表現できる。

$$M = \begin{pmatrix} \cos \theta & 0 & \sin \theta & t_x \\ 0 & 1 & 0 & t_y \\ -\sin \theta & 0 & \cos \theta & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

変換前の座標を A 、変換後の座標を B とおくと、座標変換の式は次のようになる。

$$\mathbf{B} = \mathbf{M}\mathbf{A} \quad (1)$$

変換行列 M を求めるには、変換前の座標と変換後の座標がそれぞれ 2 点必要である。試作システムでは、2 つの点を、胴体と左腰としている。変換前の座標として、Kinect で取得した胴体と左腰の座標を、それぞれ点 $C(C_x, C_y, C_z), D(D_x, D_y, D_z)$ とおく (図 11)。変換後の座標 C', D' はそれぞれ

$C'(0,0,0), D'(10, D'_y, -1)$ とする。 t_y については、胴体の座標のみで求める。これらの座標を式(1)に代入すると次の式になる。

$$\begin{aligned}
 t_y &= -C_y \\
 C_x \cos \theta + C_z \sin \theta + t_x &= 0 \\
 C_z \cos \theta - C_x \sin \theta + t_z &= 0 \\
 D_x \cos \theta + D_z \sin \theta + t_x &= 10 \\
 D_z \cos \theta - D_x \sin \theta + t_z &= -1
 \end{aligned}$$

この方程式を解いて変換行列を求める。変換行列を求めた後は、各ノードの座標を座標変換の式(1)に代入して、対象物座標系に変換する。

変換した対象物座標系での右手ノードの動きが、3.4節で定義した動きに合致した場合、右手を突き出す動きをしたと判定する。

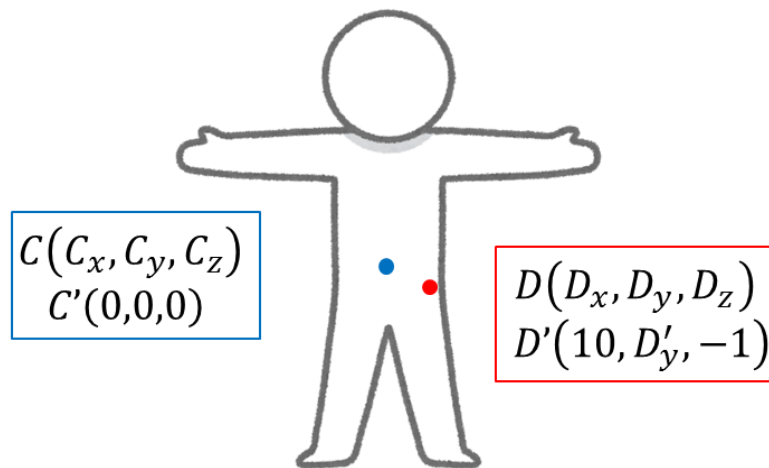


図 11 変換行列を求めるための 2 点

3.6. 動きに応じた仮想物体合成

右手を突き出したと判定したタイミングに合わせて、動きの方向に仮想物体を合成する。試作システムでは、人物の姿勢測定と現実空間の撮影はいずれも Kinect を使用した。このため、センサ座標系と表示座標系が一致している。仮想物体の合成では、センサ座標系で測定した右手の位置をそのまま用いる。また、動きの方向もセンサ座標系で取得することにより、そのまま表示座標系での合成に用いる。

図 12 のように突き出す前の右手の位置 (図 12 の H) から突き出した右手の位置 (図 12 の I) へと移動するときにはいろいろな動き方が考えられるが、今回の試作システムでは H から I への移動は、直線のみだと考え、動きの方向を、H と I の差分ベクトルとして取得する。

実際の合成では、突き出した右手の位置から動きの方向に長方形と球体を合わせた仮想物体を、差分ベクトルの方向に合成する。これにより、右手の動きを延長するような効果を実現する。

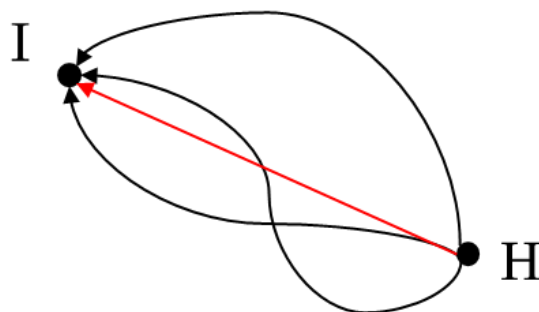
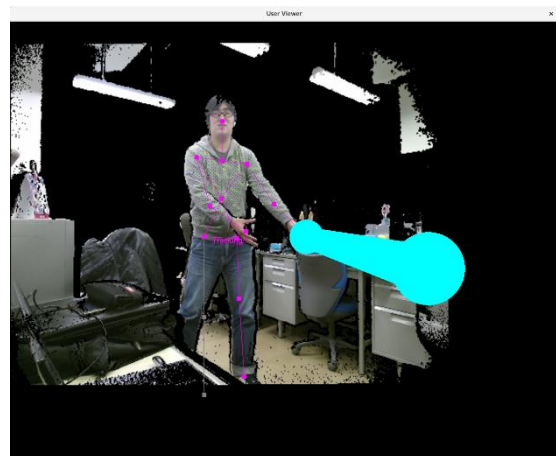
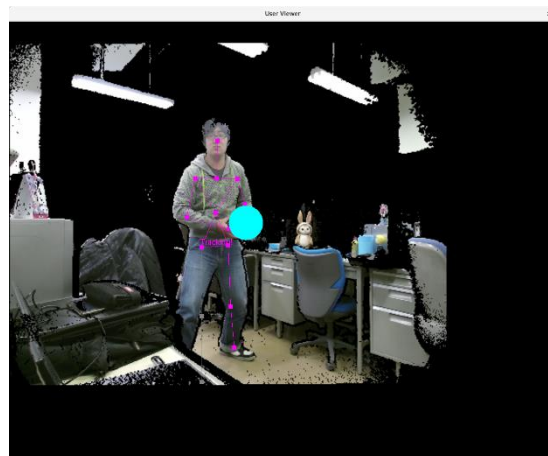
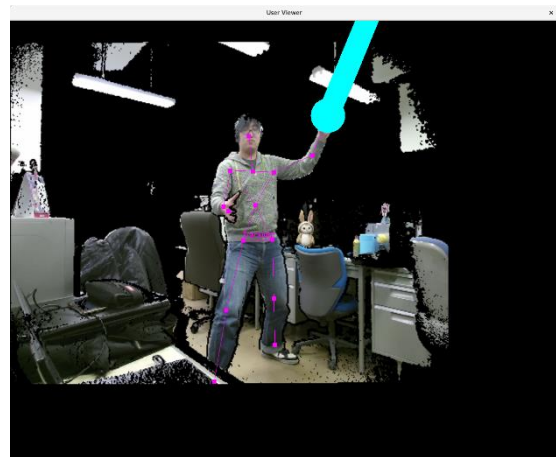
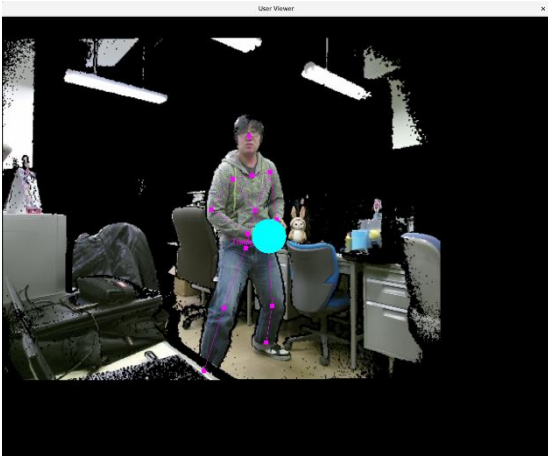
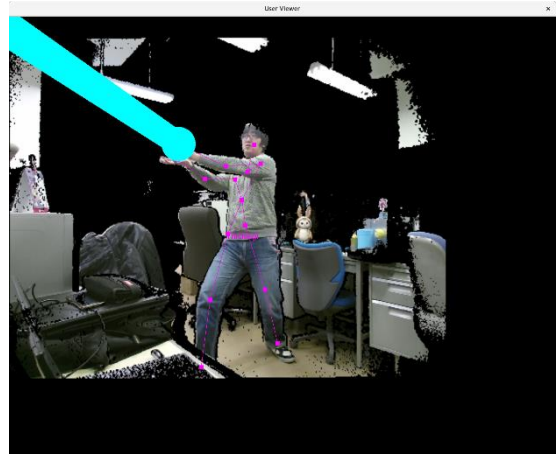
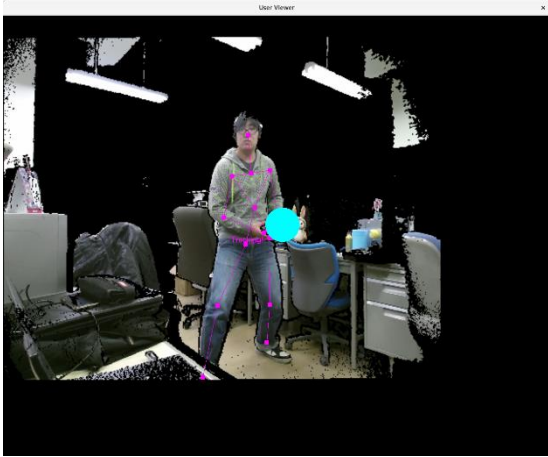


図 12 H から I への移動

4. 実行結果

試作システムの実行結果を図 13 に示す。図 13 は、右手を引いた姿勢から 3 方向に右手を突き出しているものである。それぞれの方向に仮想物体が合成できているのがわかる。右手を突き出したときにタイミングを合わせて仮想物体を表示している。また、今回の試作システムでは速さについては考えないものとしているため、一定の時間で仮想物体が伸びている。



引いた姿勢

突き出した姿勢

図 13 実行結果

5. 評価実験

試作システムが、動きが拡張されたと感じるかどうかの評価と改善点を抽出するため、アンケートによる調査を行った。

5.1. 実験の概要

7人の被験者に試作システムを使用してもらい、右手を突き出す動作を行ってもらった。動作を行うに当たって、次のような指示を出した。

- 1.かめはめ波を打ってもらいます。
- 2.右手と左手を体の右腰の辺りに持ってきてください。
- 3.仮想物体が画面上で大きくなるので、変化が終わるまで待ってください。
- 4.右手を突き出してください。

このとき、右手を突き出す方向は、被験者の自由とした。

5.2. 評価方法

被験者に試作システムを使用してもらい、その後、試作システムが動きを基準としたARといえるかアンケート調査を行い評価した。アンケート項目は3つ設定し、4段階で評価してもらった。また、「良かった点」、「改善して欲しい点」の2点を、自由記述として書いてもらった。

5.3. 実験結果

表 1 にアンケート結果の平均を示す。

表 1 アンケート結果

評価項目（最高点：4.0、最低点：1.0）	1.0	2.0	3.0	4.0	平均
動きの認識が正しいか	0人	0人	3人	4人	3.6
動きに整合した仮想空間が合成されたか	0人	3人	4人	0人	2.6
動きが拡張されたと感じるか	0人	4人	3人	0人	2.4

評価項目ごとにアンケートを見ると、ばらばらに分かれているのではなく、まとまった結果になっているのがわかる。このことから被験者全員が似たような評価をしていることがわかる。「動きの認識が正しいか」については、高い評価を受けている。これは、試作システムのプロトタイプを一般来場者に体験してもらったときに条件が厳しく、仮想物体の表示が出来なかったことが多々あったため、今回はそれをふまえて、条件を改良していた効果だと考えられる。

「動きに整合した仮想空間が合成されたか」、「動きが拡張されたと感じるか」については、いずれも中程度の評価となった。

「良かった点」については、動きに対して AR がきちんと追従していて良かった、対応できる動き（方向）が多いなどといった意見が挙げられた。

「改善して欲しい点」については、仮想物体の動きが実際の動きの延長として視覚的に分かりにくい、右手の動きの速さに応じた仮想物体の表示が欲しい、初めてやると姿勢がとりにくいなどといった意見が挙げられた。

5.4. 考察

動きの認識が上手くいったことには高い評価が得られたが、他の二項目についてはあまり評価が良くなかった。アンケートの 3 項目は、被験者の意見がばらばらになるのではなく、似たような結果を答えている。つまり、被験者内で試作システムの使いやすさや、動きが拡張できるかの感じ取り方は、変わらないということがわかる。

「動きの認識が正しいか」及び自由記述の「良かった点」から、動きの検出が上手くいき、それに対応する仮想物体を生成することができたと考えられる。

一方、「動きに整合した仮想空間が合成されたか」、「動きが拡張されたと感じるか」の部分が中程度になった点は改良の余地がある。自由記述の「改善して欲しい点」における、仮想物体の動きが実際の動きの延長として視覚的に分かりにくい、右手の動きの速さに応じた仮想物体の表示が欲しいといった意見も、同様の評価といえる。この原因としては、今回の試作システムでは、動きの方向とタイミングについては考えているが、動きの速さについては考えていないためであり、そこが試作システムを使う上で違和感となったからだと考えられる。

「改善して欲しい点」で初めてやると姿勢がとりにくいという意見が出た原因としては、本システムでは、動きを検出するための条件が多く、口頭での説明ですべての条件を伝え切れなかったことが考えられる。また、体を動かすためのスペースが小さく体を十分に動かせなかったことも原因だと考えられる。

6. おわりに

本研究では動きを基準とした位置合わせを行い、動きに整合した AR の実現について検討した。定義した動きを検出し、動きの方向、タイミングに整合して仮想物体を合成するという、動きを基準とした AR の試作システムを作成した。動きの方向、タイミングに合わせた仮想物体の表示には成功したものの、被験者には仮想物体の動く速さが現実の右手の速さに合っていないため物足りなく感じ、動きが拡張されたと感じることができない者もいた。

今後の課題として、動きの速さに応じた仮想物体の生成が挙げられる。仮想物体がただ伸びるだけでは、動きが拡張されたと感じることが難しく、速さについて考慮する必要がある。

謝辞

本論文を作成するにあたり、丁寧で熱心なご指導を頂いた指導教員の椋木雅之教授に感謝致します。また、研究の相談や助言を快く受けて頂いた椋木研究室の皆様ならびにアンケートに協力して頂いた情報システム工学科の 4 年の皆様に感謝します。本研究では、OpenNI2、Nite2 のライブラリを使用させていただきました。両ライブラリの製作者の皆様には感謝します。

参考文献

- [1] 神原誠之：拡張現実感（AR）基礎 1：拡張現実感（Augmented Reality：AR）概論,情報処理学会,情報処理 Vol.51 No.4, pp.367-372, 2010.
- [2] H.Kato et al.：Marker tracking and hmd calibration for a video-based augmented reality conferencing system,Proc. Int. WS on AR, pp.85-94, 1999.
- [3] 梶山隼 他：ユーザの姿勢検出と透過性 HMD を用いたゲームエクスペリエンス拡張システム,情報処理学会研究報告, Vol.2014-UBI-41 No.12, pp.1-7, 2014.
- [4] 森田健太郎 他：指動作認識を利用したスマートグラス上のユーザインターフェイス操作,DICOMO 2016 シンポジウム論文集, pp.866-871, 2016.
- [5] “波動”を打って闘う AR スポーツ「HADO」,
<http://jp.techcrunch.com/2017/11/06/ar-game-hado-developer-meleap-fundraised-300-mil-yen/> (2018年2月3日アクセス).
- [6] Kinect v2, <https://developer.microsoft.com/en-us/windows/kinect> (2018年2月3日アクセス).
- [7] OpenNI2, <https://structure.io/openni> (2018年2月2日アクセス).
- [8] Nite2, <http://openni.ru/files/nite/index.html> (2018年2月2日アクセス).