

2025 年度（令和 7 年度）

創造工学セミナー II Final Report

# 全天球カメラを用いた 人物位置推定手法に関する研究

研究メンバー

S522012 勝山 凱文

S522051 向野 雅人

指導教員

金丸 隆志 教授

所属研究室

知能機械研究室

## 目次

1.	緒言 (向野)	3
1.1	研究背景	3
1.2	先行研究	4
2.	研究目的と用いる手法 (向野)	6
2.1	研究目的	6
2.2	用いる手法	7
2.3	人物位置推定手法	8
3.	研究環境 (勝山)	10
3.1	ソフトウェア	10
3.1.1	骨格認識モデル MediaPipe	10
3.1.2	多人数認識モデル Lightweight OpenPose	11
3.1.3	グラフ描画ライブラリ Matplotlib	12
3.2	ハードウェア	13
3.2.1	システム動作用 GPU 非搭載パソコン	13
3.2.2	システム動作用 GPU 搭載パソコン	14
3.2.3	全天球カメラ RICOH THETA V	15
4.	実験	17
4.1	手動計測による予備実験 (向野)	17
4.1.1	実験概要	17
4.1.2	人物距離推定の精度検証	18
4.1.3	人物位置推定の精度検証	19
4.1.4	カメラを設置する高さによる精度の検証	23
4.2	MediaPipe による実験 (勝山)	26
4.2.1	実験概要	26
4.2.2	実験環境	27
4.2.3	実験方法	29
4.2.4	実験結果	30
4.3	Lightweight OpenPose による実験 (勝山)	33
4.3.1	実験概要	33
4.3.2	実験環境	34
4.3.3	実験方法	34
4.3.4	認識対象者が 1 人の場合の実験結果	36
4.3.5	認識対象者が 2 人の場合の実験結果	38
4.4	考察 (勝山)	43

5. 結言 .....	46
5.1 結論 (勝山) .....	46
5.2 課題 (向野) .....	47
5.3 応用 (向野) .....	48
参考文献 .....	49
謝辞.....	51

## 1. 緒言（向野）

### 1.1 研究背景

近年、空港や美術館、オフィスビル、商業施設などの屋内環境で自律移動ロボットの活用が進んでいる。例えば、東京スカイツリーでは、自律走行型の案内ロボットを用いて、団体客を目的地まで誘導する実証実験が行われており、人手不足の解消や業務の効率化を目的とした取り組みが報告されている(図 1)[1]。身近な所では、レストランの配膳ロボット[2]や、オフィスビルなどで活用されている警備ロボット[3]などがある(図 2)。これらのロボットは、人口減少に伴う労働力不足の解消や、業務の省人化・効率化への貢献が期待されている。これにともない、今後も、国内では自律移動ロボット市場は拡大が見込まれており、2024年に2億2,576万米ドルと評価され、2033年には10億7,934万米ドルの市場評価額に達すると予測されている[4]。

これらのロボットが普及していく中で、ロボットが人と適切な距離を保つことが重要であり、近すぎて圧迫感を与える、遠すぎて本来の役割を遂行できないといった問題を防ぐ制御が求められる。

そこで、本研究では人との適切な距離を保ちながら誘導や案内をするロボットへの適用を目的として、リアルタイムな人物位置推定を実現することを目指す。



図 1 東京スカイツリーでのロボット案内誘導イメージ[1]



図 2 配膳ロボット(左)[2]と警備ロボット(右)[3]

## 1.2 先行研究

本研究では、人との適切な距離を保ちながら誘導や案内をするロボットへの応用を目指し、リアルタイムな人物位置推定を行うことを目的としている。そこで本節では、人物位置推定手法に関連する先行研究を挙げる。

1 つ目は、「動的環境下において複数人の同行者を誘導する自律移動ロボット」に関する研究である[5]。この研究では、人工ポテンシャル法を応用した移動制御手法を提案している。人工ポテンシャル法とは、障害物に斥力、目標地点に引力を設定し、それらの合成ベクトルを基にロボットの移動方向を決定する手法である。動的障害物と静的障害物を区別し、動的障害物にはより強い斥力を与えるなど、リアルタイムな安全な回避行動を実現している。また、同行者の位置を維持するために引力ポテンシャルを適用することで、障害物回避と同行者誘導の両立を達成している。

この研究では、人物の脚部と胸部の位置情報を、2つのレーザーレンジファインダー(LRF)でそれぞれ取得することで、人物位置を推定する手法が用いられている(図3)。これは Carballo 氏らによって提案された人物検出手法である(図4)[6]。しかし、この手法では身長が低い人物の検出には課題が残る。

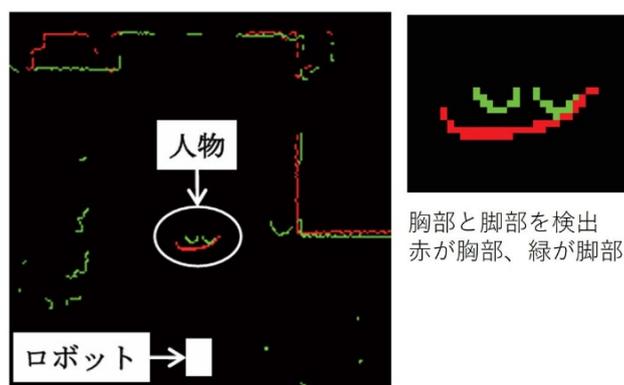


図3 2つのLRFを用いた人物位置推定[5]

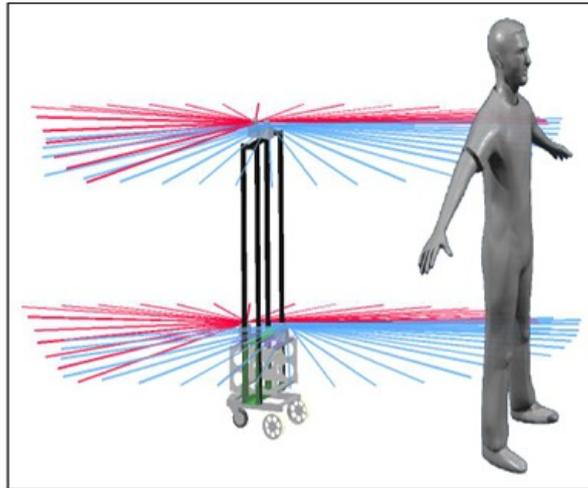


図 4 胸部と脚部の位置情報を取得する 2つの LRF のイメージ[6]

2 つ目は、「単眼全天球カメラを用いた自己位置推定と人の位置情報のマッピング」に関する研究である[7]。この研究では、全天球カメラを用いて効率的に施設内の混雑状況を 3 次元的に可視化する手法を提案している。具体的には、移動するカメラで撮影した画像から 3 次元の環境モデルを生成し、物体検出モデルである YOLOX で人物検出をすることで、3 次元環境内の人の分布を可視化している。

この研究では、Masuda 氏らが提案した、全天球カメラを用いた物体の位置推定手法が人物に応用されている[8]。この手法は、画像の正距円筒図法変換を用いて人の足元座標をピクセル単位で取得し、視線方向や方位角を計算することで、カメラの設置高さに依存せず人物の位置を特定することができる。全天球カメラを用いた人物位置推定手法については次章以降で詳しく説明する。

これらの先行研究を踏まえ、本研究では、移動ロボットへの搭載を想定し、全天球カメラから取得される映像を用いて、リアルタイムに人物位置を推定する手法について検証する。

## 2. 研究目的と用いる手法（向野）

### 2.1 研究目的

人と空間を共有する自律移動ロボットにおいて、周囲の人物位置をリアルタイムに把握することが必要である。特に、ロボットと人の距離を適切に制御するためには、人物の方向および距離を安定して推定できることが求められる。

本研究では、参考文献[7]と[8]で利用されていた全天球カメラに着目し、移動ロボットに搭載した全天球カメラから取得される映像のみを用いて、リアルタイムに人物位置を推定することを目指す。

## 2.2 用いる手法

本研究では、移動ロボットに搭載した全天球カメラを用い、周囲に存在する人物の位置推定をリアルタイムに行う。人物位置推定手法には Masuda 氏らが提案した手法[8]を用いる。全天球カメラは周囲 360 度の全方位を一度に撮影できるため、ロボットの進行方向に依存せず、複数方向に存在する人物を同時に観測できるという利点がある。この特性により、混雑した屋内環境においても安定した人物検出および位置推定が可能となる。全天球カメラについては 3 章にて詳しく説明する。

本研究では、全天球カメラで撮影した画像を正距円筒図法で表現し、その画像の中から人物検出モデルにより人物を検出する。検出された人物のうち、地面と接している足元の位置に注目し、人物がカメラから見てどの方向にあり、どの程度離れているかを計算することで、実空間での人物の位置を推定する。これらの手法については、次節以降で詳しく説明する。

### 2.3 人物位置推定手法

本節では、全天球画像内の人物の位置から、実空間内の人物の位置を推定する手法を解説する。正距円筒図法で出力された全天球画像内で人物の足元に相当する画素位置が $(a, b)$ であるとする(図 5)。この画素位置を基に、全天球画像の鉛直下方向とカメラから人物の足元へ方向とがなす角度 $\theta_1$ は(1)式で、カメラ座標系のZ軸を基準とした人物の足元の方位角 $\theta_2$ は(2)式で求められる(図 6)。

$$\theta_1 = \frac{\pi a}{H} \quad (1)$$

$$\theta_2 = 2\pi \frac{b}{W} \quad (2)$$

ここで、 $H$ は画像の高さ、 $W$ は画像の幅を表す。これにより、全天球カメラの位置から見た人物の方向が特定される。

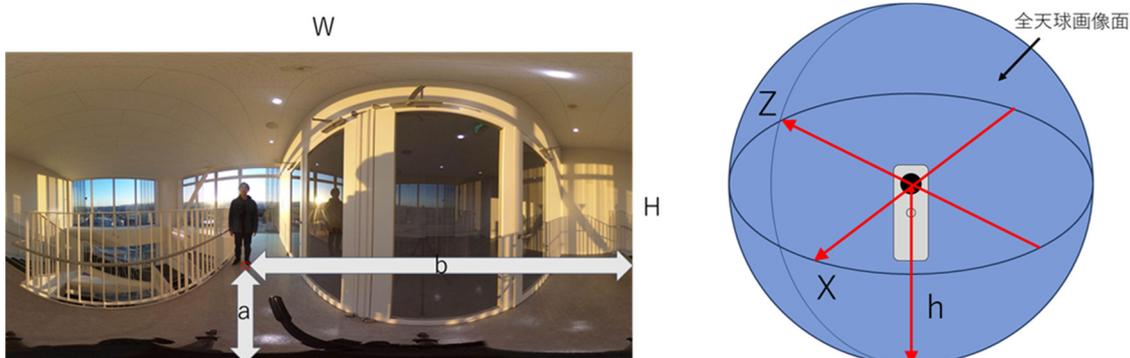


図 5 全天球画像とカメラ座標系

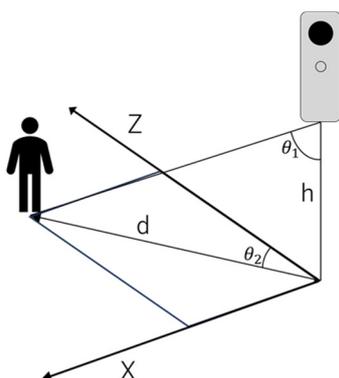


図 6 カメラと人物の位置関係

次に、実空間における人物の位置を計算する。カメラから地面までの高さを $h$ 、カメラから人物までの距離を $d$ とした場合、図 6 より $d$ は(3)式で求められる。

$$d = h \tan \theta_1 \quad (3)$$

$d$ を用いて、実空間における人物の位置( $X, Z$ )はそれぞれ(4)式、(5)式で算出される。

$$X = d \sin \theta_2 \quad (4)$$

$$Z = d \cos \theta_2 \quad (5)$$

これらの計算により、実空間における人物の位置が推定できる。

### 3. 研究環境（勝山）

本章では研究で用いるソフトウェアとハードウェアについて説明する。

#### 3.1 ソフトウェア

本節では、本研究のソフトウェアで用いる機械学習ライブラリとグラフ描画ライブラリについて説明する。

##### 3.1.1 骨格認識モデル MediaPipe

ここでは骨格認識モデルである MediaPipe について説明する。

MediaPipe とは Google が開発したオープンソースの機械学習ライブラリ [9] である。映像に対して顔や手、足、指先などの骨格点の検出を行うツールであり、リアルタイムでの動作検出に優れている。本研究では MediaPipe のポーズ推定機能で検出される 33 点のランドマーク (図 7) の中から左のつま先を示す「left foot index」を足元位置として検証を進める。ポーズ推定機能を用いるため、kazuhito00 氏が公開しているポーズ推定用のプログラム [10] から left foot index の座標を求め、2.3 節の  $(a, b)$  を得る。それにより、人物の位置  $(X, Z)$  を求めるのである。

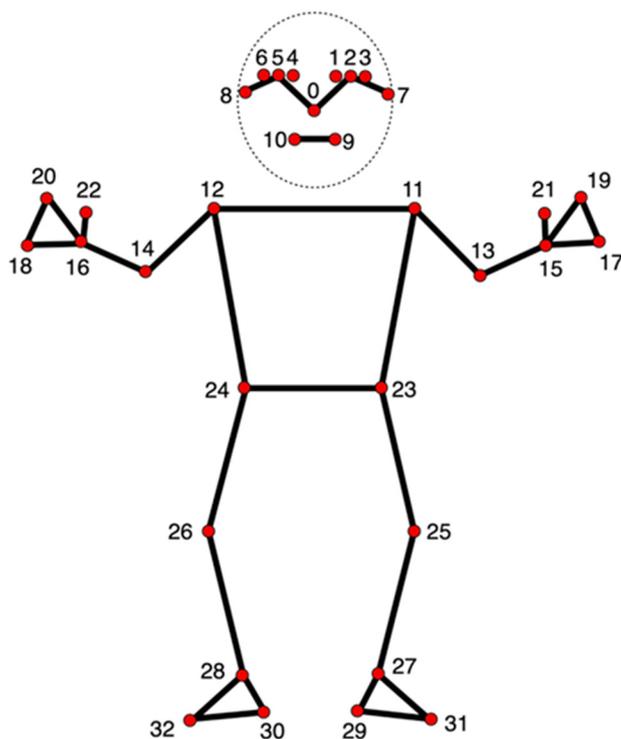


図 7 33 点のランドマーク

### 3.1.2 多人数認識モデル Lightweight OpenPose

前節で解説した MediaPipe は、画像中の 1 人の人物しか認識できないという問題がある。そのため、画像中の複数の人物を認識できるモデルが必要となる。そのため、本研究では多人数認識モデルである Lightweight OpenPose も用いることにする。

Lightweight OpenPose は Daniil Osokin 氏が開発した多人数人物姿勢推定モデル[11][12]である。元々、OpenPose という多人数人物姿勢推定モデルが存在し、それを CPU でもリアルタイム動作をすることを目標に軽量化したものが本モデルである。2つのモデルでは使われている畳み込みニューラルネットワークが異なり、OpenPose では VGG-19 が使われている一方、Lightweight OpenPose では MobileNet v1 が使われている。また、OpenPose で行われているリファインメント段階が 5 回であるのに対して、Lightweight OpenPose ではそれが 1 回にされている。これらの変更によって OpenPose の演算量を減らしたことで、軽量化を実現している。

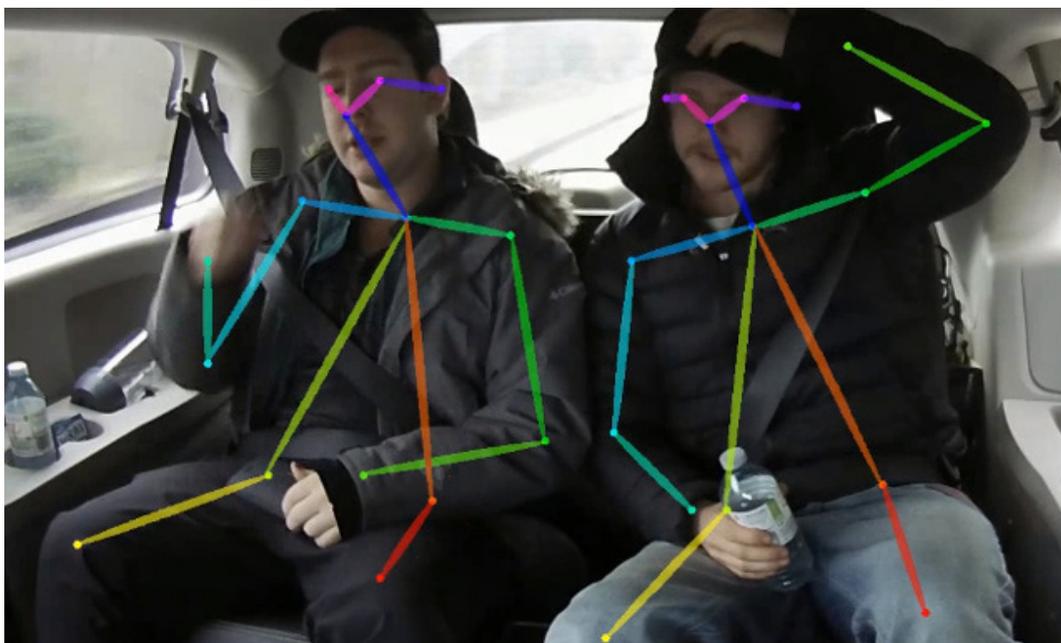


図 8 Lightweight OpenPose の姿勢推定機能[11]

### 3.1.3 グラフ描画ライブラリ Matplotlib

ここではグラフ描画ライブラリである Matplotlib について説明する。

Matplotlib は Python 向けのグラフ描画ライブラリの 1 つである。折れ線グラフや散布図、ヒストグラムなどの他、3D のグラフも描画することができる。

本研究では図 9 のように、人物の座標( $X, Z$ )を散布図として描画することで、実空間上の人物の位置をリアルタイムに確認できるようにしている。

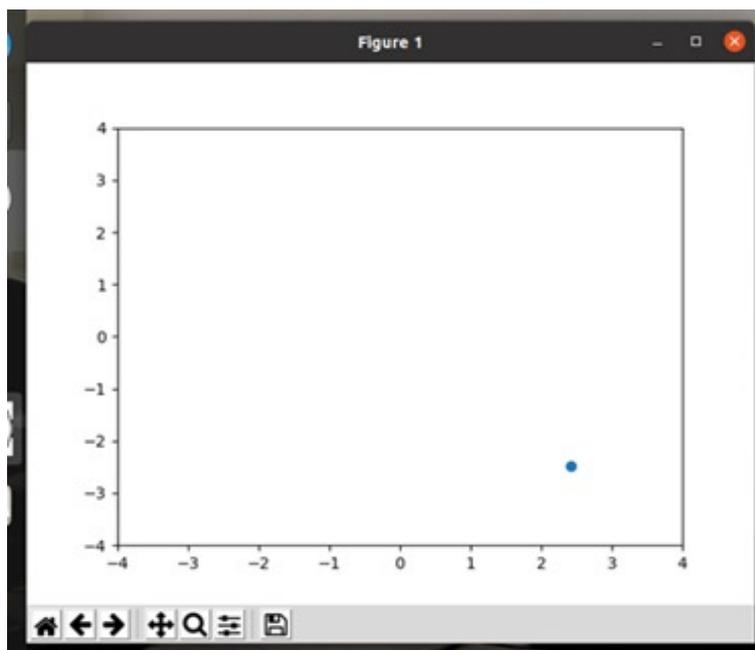


図 9 Matplotlib を使い描いた散布図

## 3.2 ハードウェア

### 3.2.1 システム動作 GPU 非搭載パソコン

システム動作の GPU 非搭載パソコン(以下 GPU 非搭載 PC)としてパナソニック製の Let's Note(図 10)を使用した。この PC には Linux 系のオープンソース OS である Ubuntu20.04 をインストールしており、主に 4.2 節の MediaPipe を利用する実験で用いる。GPU 非搭載 PC のスペックは表 1 の通りである。

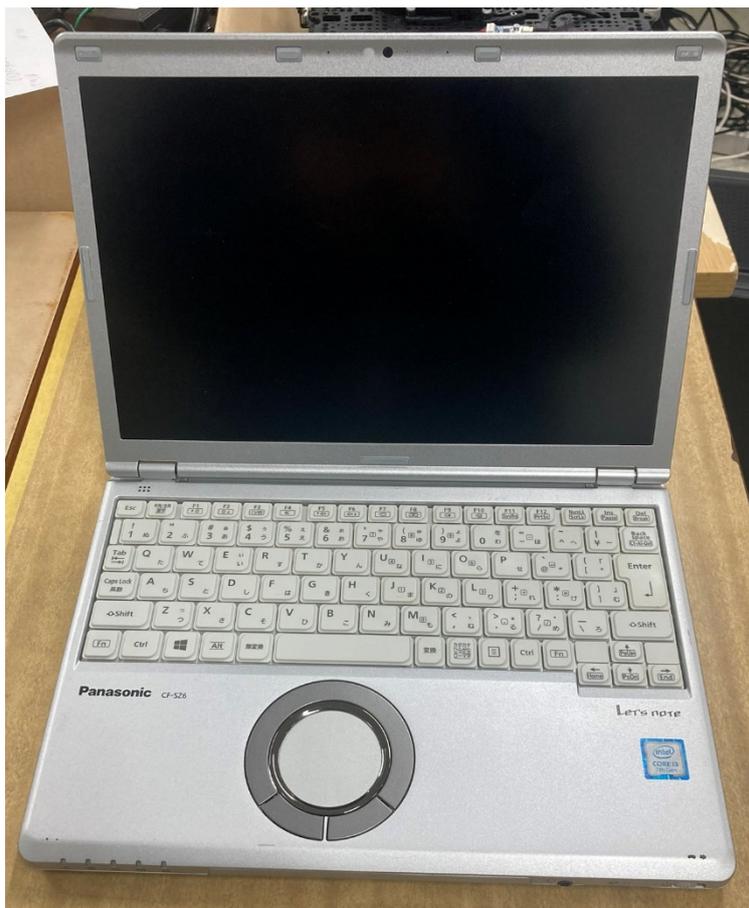


図 10 GPU 非搭載 PC

表 1 GPU 非搭載 PC のスペック

名称	Let's note CF-SZ6
搭載 OS	Ubuntu 20.04.6 LTS
CPU	Intel Core i3-7100U
メモリ	8GB

### 3.2.2 システム動作用 GPU 搭載パソコン

システム動作用の GPU 搭載パソコン(以下 GPU 搭載 PC)として iiyama 製の PC(図 11)を使用した。この PC にも Linux 系のオープンソース OS である Ubuntu20.04 をインストールしており、主に 4.3 節の Lightweight OpenPose を利用する実験で用いる。GPU 非搭載 PC のスペックは表 2 の通りである。



図 11 GPU 搭載 PC

表 2 GPU 搭載 PC のスペック

名称	iiyama NH55HP
搭載 OS	Ubuntu 20.04.6 LTS
CPU	Intel Core i7-11800H
GPU	NVIDIA GeForce RTX 3060 Laptop GPU
メモリ	16GB

### 3.2.3 全天球カメラ RICOH THETA V

本研究では全天球カメラとして RICOH 製の THETA V(図 12)を使用した。



図 12 RICOH THETA V

今回、Ubuntu 上で THETA の機能を使うために RICOH から公開されているライブストリーミング用ライブラリ[13]と codetricity 氏が公開している Python 向けの libuvc-theta-sample[14]を使用した。v4l2loopback を読み込んで Dummy video device にライブストリーミングモードで接続した THETA からの映像を送ることで他のウェブカメラと同じように THETA を使うことができるようになっている。

全天球カメラは周囲 360 度の全方位を同時に撮影することが可能なカメラである。通常のカメラは一定の視野角のみとらえることができる一方、全天球カメラでは 180 度以上映る 2 つの魚眼レンズを組み合わせることで水平方向、及び垂直方向の両方を記録することができる。この 2 つのカメラ画像を正距円筒図法に変換することで、上下左右 360 度を 1 つの写真や動画に収めることが可能になっている。図 13 は THETA を用いて撮影した八王子キャンパス 4 号館 8 階廊下の写真である。

映像中央の正面がZ軸上の負の方向を表している。

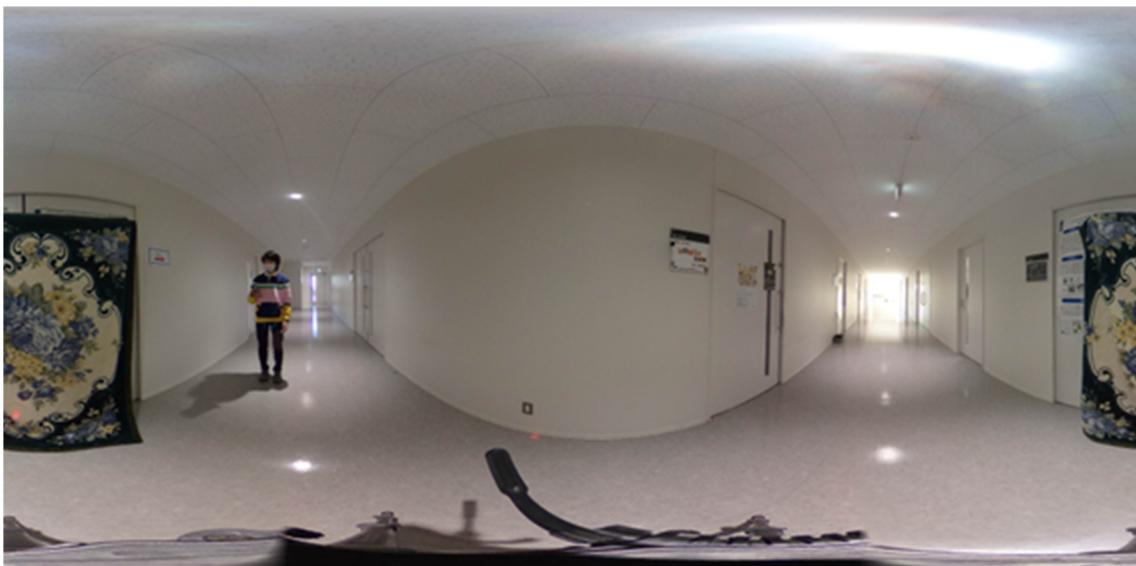


図 13 THETA を用いて撮影した全天球カメラの写真

## 4. 実験

### 4.1 手動計測による予備実験（向野）

#### 4.1.1 実験概要

2.3 節で述べた、全天球カメラを用いた人物位置推定の有効性の検証を行う。2.3 節および 3.1 節で述べたように、本研究の人物位置推定には、人物の足元の座標 $(a, b)$ が必要となる。 $(a, b)$ を求めるためには通常、骨格認識モデルを用いたプログラムを用いる。しかし、本節では簡易に検証を行うため、手動で $(a, b)$ を求める。具体的には全天球カメラから出力された正距円筒図法の静止画を印刷し、 $(a, b)$ の長さを定規で測定する。人物の足元位置は目視にて判定した左足のつま先を採用するものとする。

本実験は工学院大学八王子キャンパス 4 号館 8 階のラウンジと廊下にて行う。

#### 4.1.2 人物距離推定の精度検証

屋内にて 1.0m の高さに固定した全天球カメラの床上の位置を原点とし、そこからの距離を床に記す。そして、カメラ位置からの距離を変えながらZ軸上で静止した人物の写真を撮影した。人物の足下の位置を目視で測り、 $(a, b)$ を求めた上で(1),(2),(3)式により、カメラからの距離を $d$ を算出する。なお、(1),(2)式の $H$ と $W$ は、印刷した画像の高さと幅を定規で測定することで求める。

算出した距離 $d$ と実際の距離 $d'$ の比較結果を図 14 に示す。0.5~5.0m の範囲において、算出距離の誤差は最大でも約 7.3%であり、すべての測定点で誤差が 1 割未満に収まることを確認できた。以上より、2.3 節で示した全天球カメラを用いた人物距離測定手法は、屋内環境において高い測定精度を有することが確認できた。

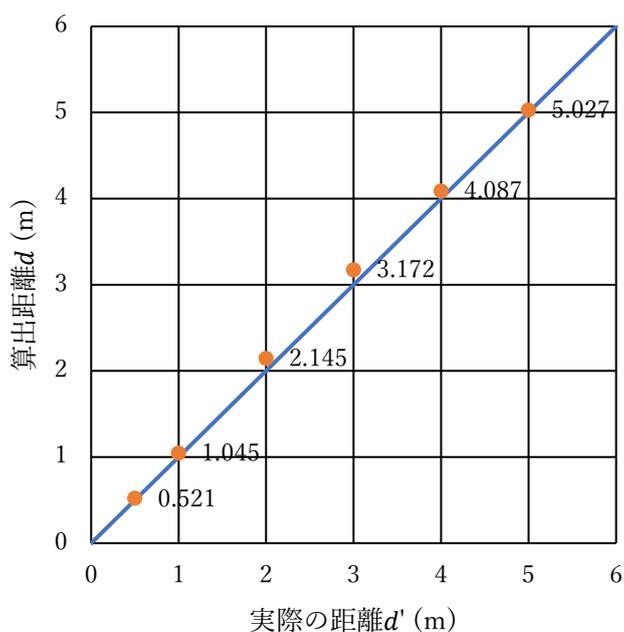


図 14 カメラから人物までの算出した距離

### 4.1.3 人物位置推定の精度検証

4.1.2 ではカメラからの距離 $d$ の精度を調べた。

本節では、全天球カメラを用いた人物位置推定の有効性を知るため、人物位置 $(X, Z)$ の精度を調べる。図 6 の方位角 $\theta_2$ の値が  $0 \sim \pi$  までの人物の範囲での人物の位置を求めるものとする。

4.1.2 と同様、人物の画像領域における足下位置を目視で決定して定規で測り、(1)~(5)式により、カメラ位置から人物の位置 $(X, Z)$ を算出した。

結果として算出した人物の位置 $(X, Z)$ と実際の位置を表 3 に示す。また、全天球カメラを原点とした人物の配置を図 15 に示す。

図 15 に示す人物配置から、人物位置 $(X, Z)$ の推定結果には実際の配置との差が大きいことがあると確認できる。そこで、人物位置推定の精度が低い要因を調べるため $d$ を図 16、 $\theta_2$ を図 17 に示し、それぞれについて精度を評価する。

カメラから人物までの距離 $d$ については、測定条件によっては実寸値に対して 1 割を超える誤差が生じており、高い精度で推定できているとは言えない。

一方で、人物の方位角 $\theta_2$ については、算出値と実寸値の差は最大でも  $5^\circ$  以内に収まっており、人物の方向は高い測定精度を有している。

次に、距離 $d$ の推定誤差が大きくなる要因について考察する。図 5 より、画像の縦方向の足元位置 $a$ は、人物がカメラから遠ざかるほど画像上で圧縮される。そのため、(1),(3)式より、遠距離になるほど画像上で判定した足元位置の誤差が、算出した距離 $d$ の誤差を大きくすると考えられる。

以上より、距離 $d$ については遠距離で誤差が大きく、精度が高いとは言えない一方で、人物の方向を表す $\theta_2$ については、高い精度で推定可能であることが確認できた。

表 3 推定した人物位置( $X, Z$ )と実際の人物位置( $X', Z'$ )

X	X'	Z	Z'
0.957	1	-0.005	0
1.92	2	-0.053	0
2.57	3	-0.07	0
-0.033	0	-0.999	-1
0.935	1	-0.993	-1
1.896	2	-1.004	-1
2.783	3	-0.965	-1
-0.032	0	-2.946	-3
1.056	1	-2.991	-3
2.005	2	-2.785	-3
2.929	3	-2.85	-3
0.062	0	-5.671	-5
1.292	1	-5.522	-5
2.12	2	-5.26	-5
-0.022	0	1	1
0.946	1	0.983	1
1.738	2	1.046	1
2.75	3	1.056	1
-0.069	0	3.171	3
1.023	1	3.002	3
1.785	2	2.93	3
2.528	3	2.745	3
-0.296	0	4.501	5
0.689	1	4.458	5
1.946	2	5.327	5
3.135	3	5.691	5

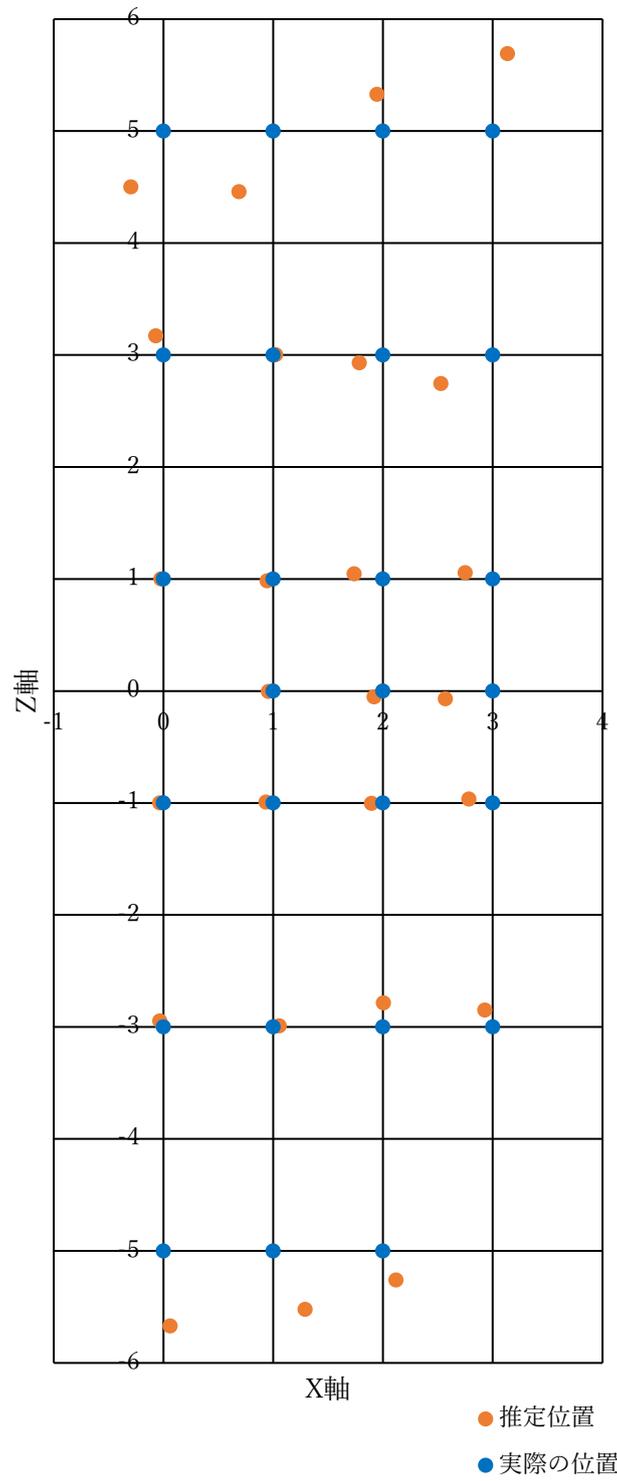


図 15 推定した人物位置( $X, Z$ )と実際の人物位置( $X', Z'$ )

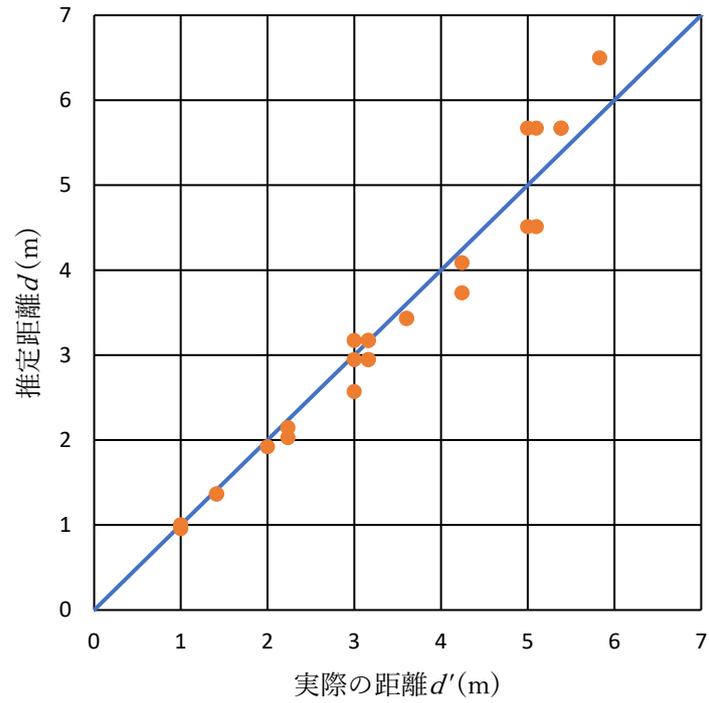


図 16 推定距離 $d$ と実際の距離 $d'$

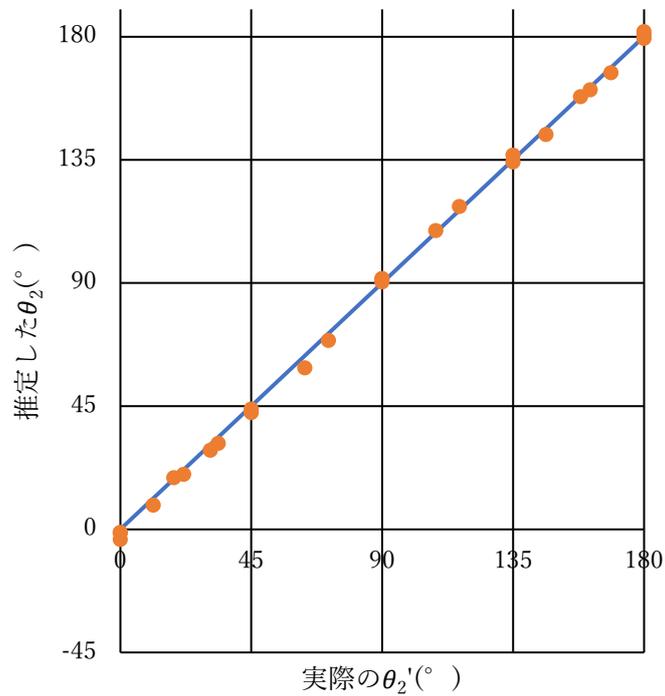


図 17 推定した角度 $\theta_2$ と実際の角度 $\theta_2'$

#### 4.1.4 カメラを設置する高さによる精度の検証

実際に移動ロボットに全天球カメラを搭載することを考慮し、全天球カメラを設置する高さ $h$ が人物位置推定の精度への影響を与える可能性について考える必要がある。そこで、本節ではカメラを設置する高さ $h$ を変更した場合に人物位置推定の精度が変わるかを調べる。本実験では、カメラ設置高さ $h$ として 0.5 m, 1.0 m, 1.5 m の 3 つを用いる。

結果を表 4 に示す。また、この表をグラフ表示したものが図 18 である。いずれの $h$ においても、大きな精度差は見られなかった。以上の結果から、以降の実験ではカメラ設置高さ $h$ を 1.0 m として実験を行うこととする。

表 4 3つの高さhにおける人物推定位置(X,Z)と実際の人物位置(X',Z')

カメラの 高さ h	0.5m	1m	1.5m		0.5m	1m	1.5m	
	X	X	X	X'	Z	Z	Z	Z'
1	1.006	0.957	0.909	1	0.127	-0.005	-0.045	0
2	1.86	1.92	1.868	2	0.153	-0.053	-0.051	0
3	2.831	2.57	2.881	3	0.171	-0.07	-0.016	0
4	0.073	-0.033	-0.033	0	-0.957	-0.999	-1.001	-1
5	1.065	0.935	0.942	1	-0.868	-0.993	-1	-1
6	1.899	1.896	1.808	2	-0.753	-1.004	-0.958	-1
7	2.735	2.783	2.701	3	-0.751	-0.965	-1.003	-1
8	0.107	-0.032	-0.033	0	-3.247	-2.946	-3.042	-3
9	1.215	1.056	1.071	1	-3.013	-2.991	-3.034	-3
10	2.219	2.005	1.985	2	-3.082	-2.785	-3.029	-3
11	3.272	2.929	2.89	3	-3.183	-2.85	-2.938	-3
12	0.15	0.062	0.061	0	-4.563	-5.671	-5.598	-5
13	1.424	1.292	1.156	1	-5.535	-5.522	-5.477	-5
14	2.366	2.12	2.291	2	-5.202	-5.26	-5.686	-5
15	0.105	-0.022	0.066	0	0.954	1	1	1
16	1.055	0.946	0.932	1	0.88	0.983	0.927	1
17	1.873	1.738	1.904	2	0.815	1.046	0.982	1
18	3.142	2.75	2.905	3	0.827	1.056	0.902	1
19	0.31	-0.069	0.133	0	2.819	3.171	3.039	3
20	1.048	1.023	1.168	1	3.075	3.002	2.809	3
21	2.319	1.785	2.022	2	3.008	2.93	2.745	3
22	3.506	2.528	2.985	3	2.924	2.745	2.841	3
23	0.499	-0.296	0.225	0	4.538	4.501	5.141	5
24	1.377	0.689	1.241	1	4.352	4.458	4.241	5
25	2.536	1.946	2.182	2	5.122	5.327	4.661	5
26	3.288	3.135	3.12	3	4.674	5.691	4.648	5

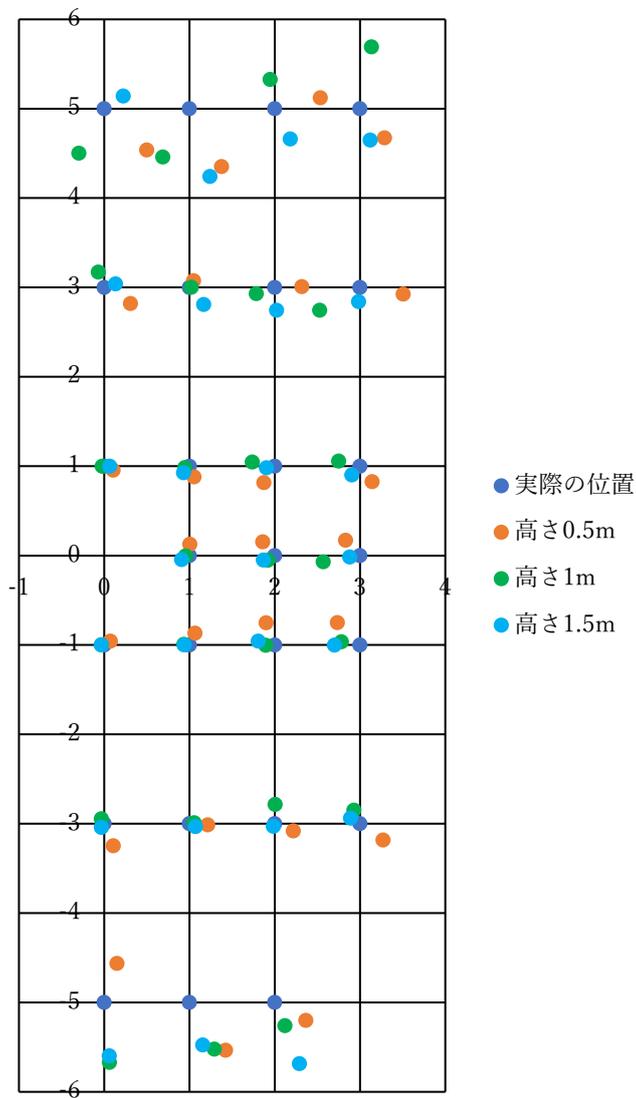


図 18 3つの高さ $h$ における人物推定位置( $X, Z$ )

## 4.2 MediaPipe による実験 (勝山)

本節では、MediaPipe を用いて取得した人物のカメラ座標系での足元座標 $(a, b)$ を求め、それにより人物位置 $(X, Z)$ を求めたときの精度について調べる。

### 4.2.1 実験概要

2.3 節の手法を用いて RICOH THETA V から得られた映像に対して MediaPipe を使い人物の足元位置の推定を行う。全天球カメラ画像の左上を原点とした座標系で足元座標が $(x, y)$ と書けるとき、 $(a, b)$ はそれぞれ $a = H - y$ ,  $b = W - x$ と書ける。この $(a, b)$ を用いて $(X, Z)$ を計算できる。図 19 はこのシステムの流れを表す図である。

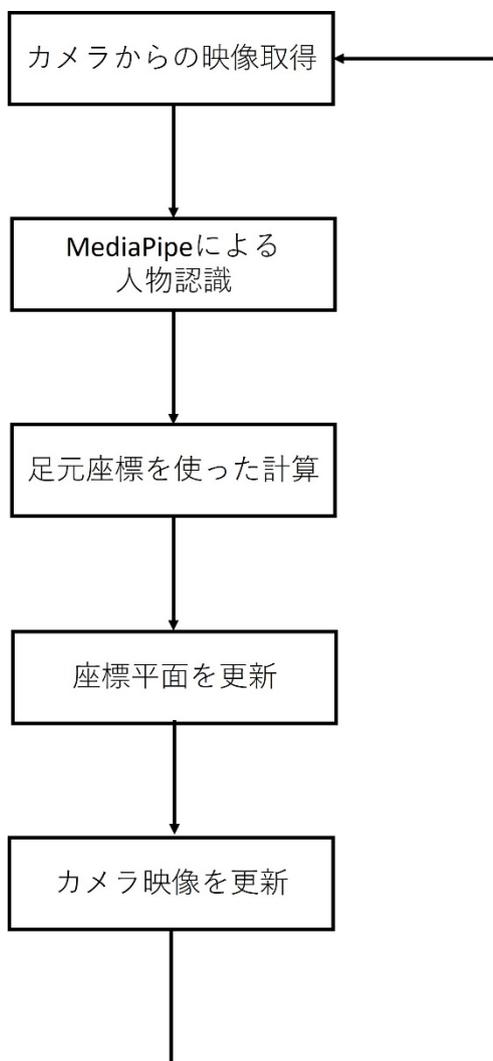


図 19 MediaPipe を使った人物位置推定システムの流れ

#### 4.2.2 実験環境

本実験は八王子キャンパスの4号館8階のラウンジ(図 20)で行った。6m×3mの空間に1m毎にビニールテープで印をつけた。そこに対象となる人物を立たせて人物位置推定を行う。人物位置の真値はビニールテープの位置とする。その真値と提案手法を用いた人物の足元位置を比較し、精度の評価を行う。ただし、以下の実験では OpenCV の flip 関数の機能により、カメラ画像がミラー表示になっている。そのため、2.3 節の図 5、図 6 と X 軸の向きが異なっていることに注意する必要がある。

具体的には、図 21 のように X 軸の負の範囲を使って実験を行う場合、画面上での X 座標は図中の  $X^{(m)}$  軸での座標を表している。同様に、図 22 のように X 軸の正の範囲を使って実験を行う場合、画面上での X 座標は図中の  $X^{(m)}$  軸での座標を表している。

以下、4.2 節及び 4.3 節で現れる X 座標は全て実の  $X^{(m)}$  軸での座標である。



図 20 八王子キャンパスの4号館8階のラウンジ

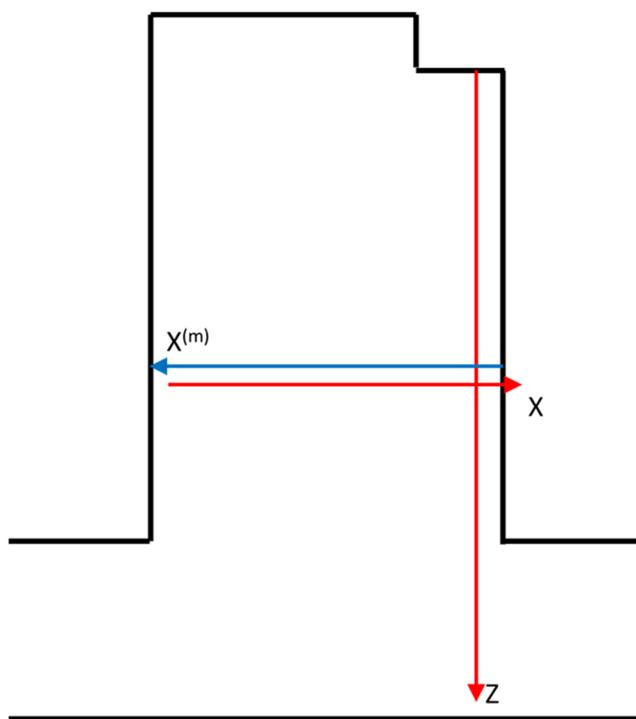


図 21 ラウンジ上の座標軸( $X < 0$ )

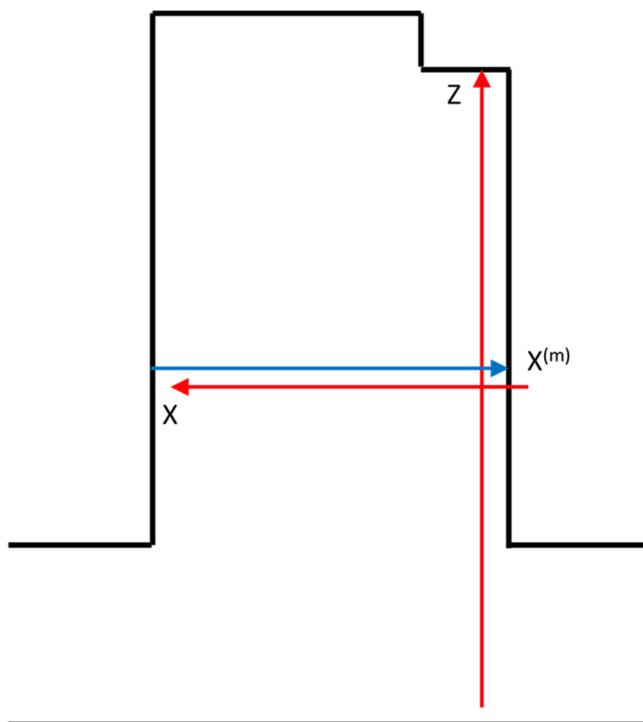


図 22 ラウンジ上の座標軸( $X > 0$ )

### 4.2.3 実験方法

本実験では屋内にて1.0mの高さに固定した全天球カメラを基準として、ビニールテープで印をつけた位置に静止した人物をライブストリーミングで撮影し、リアルタイムでの人物位置の測定処理を行った。認識対象者1名を室内の目標地点に立たせ人物の認識が終わったところでPCのスクリーンショットを撮影する。それをX座標-3~3, Z座標-3~3での原点を除いた計48点で行った。図23はX座標2、Z座標-2の位置に認識対象が立っており、足元座標が映像上のグラフに表示されているのがわかる。

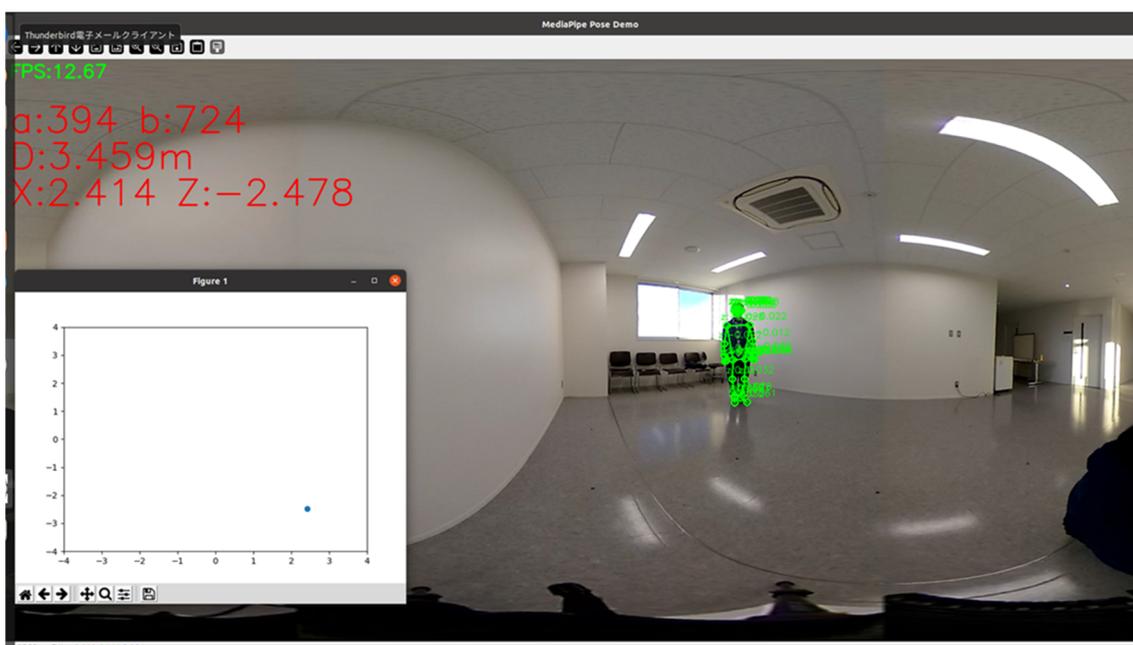


図 23 MediaPipe を用いたプログラムの動作例

#### 4.2.4 実験結果

実験で得られた足元座標を表 5 に示す。

表 5 MediaPipe で得られた足元座標( $X, Z$ )

$X$	$X'$	$Z$	$Z'$
3.796	3	-3.974	-3
3.707	3	-2.53	-2
3.534	3	-1.2	-1
3.5	3	-0.126	0
3.305	3	1.158	1
3.144	3	2.191	2
3.098	3	3.058	3
2.482	2	-3.822	-3
2.414	2	-2.478	-2
2.332	2	-1.217	-1
2.245	2	-0.066	0
2.22	2	1.113	1
2.122	2	2.136	2
2.03	2	3.017	3
1.204	1	-3.585	-3
1.187	1	-2.348	-2
1.139	1	-1.185	-1
1.121	1	2.98	3
1.103	1	-0.022	0
1.091	1	1.056	1
1.09	1	2.056	2
0.024	0	-3.591	-3
0.023	0	-2.327	-2
0.015	0	-1.171	-1
-	0	-	1
-	0	-	2
-	0	-	3
-0.827	-1	2.861	3
-0.896	-1	1.996	2
-0.961	-1	1.026	1

-1.04	-1	0	0
-1.078	-1	-1.099	-1
-1.138	-1	-2.233	-2
-1.214	-1	-3.428	-3
-1.769	-2	2.973	3
-1.871	-2	2.037	2
-1.981	-2	1.059	1
-1.993	-2	0.085	0
-2.089	-2	-1.073	-1
-2.176	-2	-2.176	-2
-2.272	-2	-3.283	-3
-2.719	-3	3	3
-2.758	-3	2.017	2
-2.859	-3	1.044	1
-2.912	-3	0.124	0
-3.05	-3	-1.035	-1
-3.201	-3	-2.2	-2
-3.254	-3	-3.191	-3

ただし、図 24 のように人物がZ軸上かつ $Z > 0$ の時は全天球画像の切れ目に人物が立っているため MediaPipe で検出することができない(画像中央の正面がZ軸上かつ $Z < 0$ の範囲)。すなわち、 $(X, Z) = (0, 1), (0, 2), (0, 3)$ の場合は測定不可能なものとして除外してある。

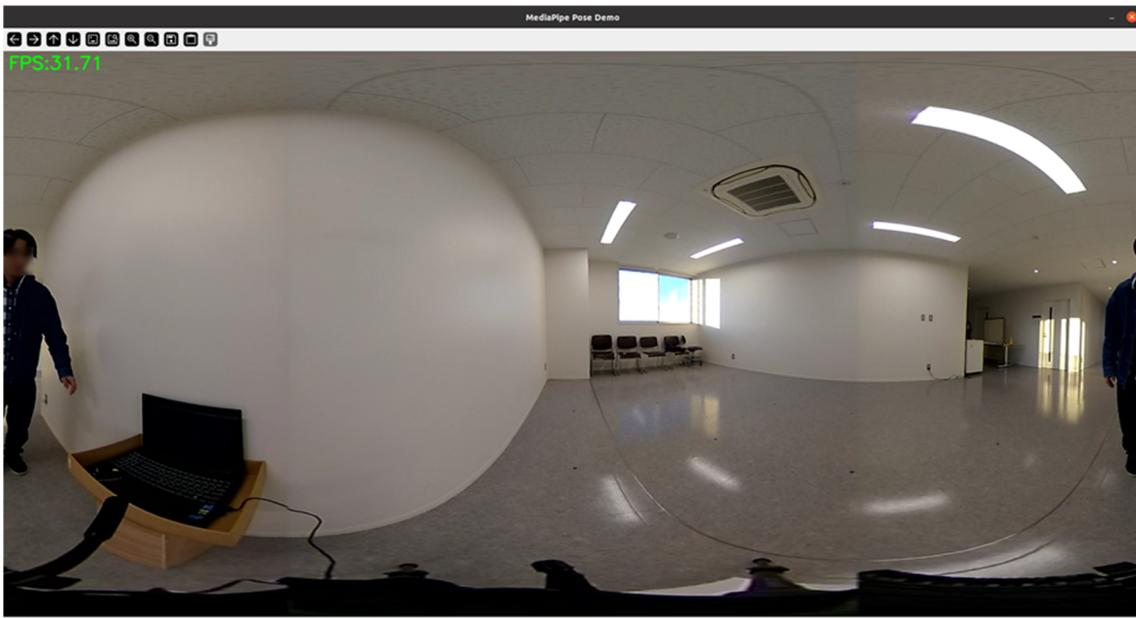


図 24 人物が見切れている例

また、表 5 を基に人物の立ち位置を表示したものを図 25 に表す。

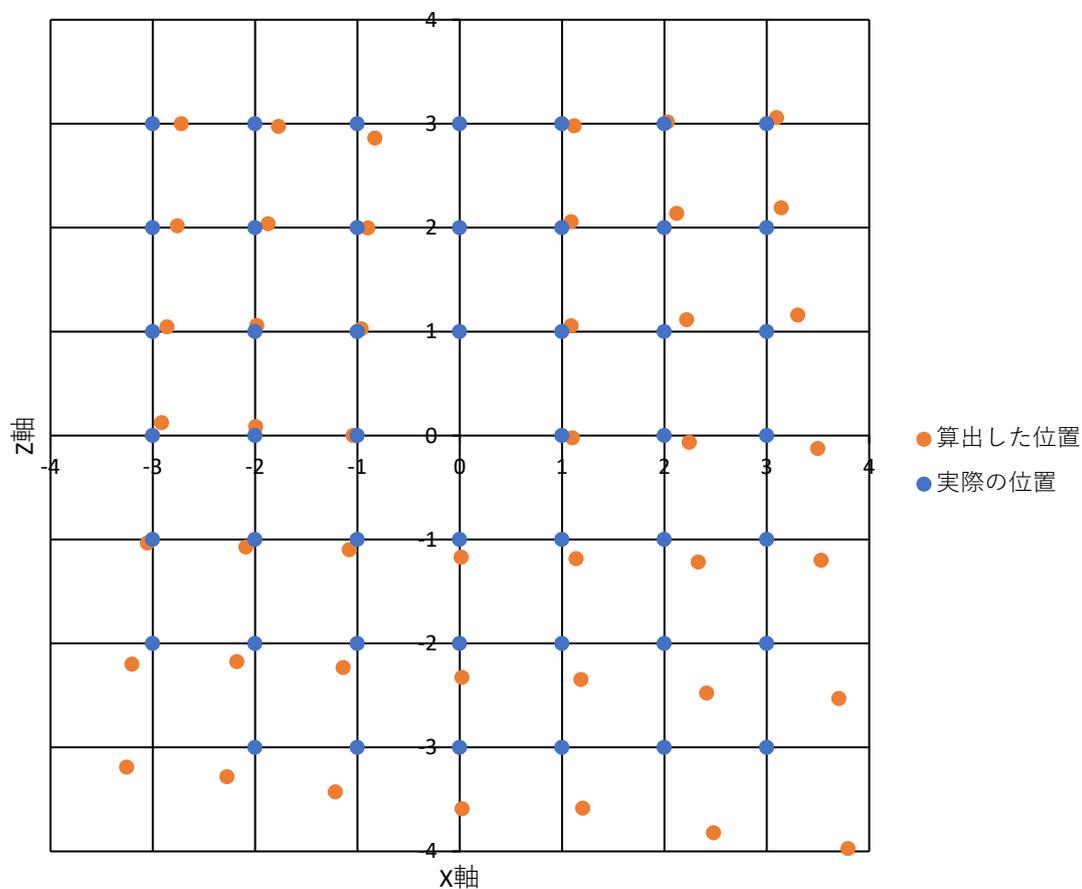


図 25 算出した人物位置( $X, Z$ )と実際の人物位置( $X', Z'$ )(MediaPipe)

傾向は 4.1.3 節での手動での実験結果である図 15 と同等に見える。

### 4.3 Lightweight OpenPose による実験 (勝山)

#### 4.3.1 実験概要

本節では、4.2 節のシステムを複数の人物の認識に対応させる。そのために、RICOH THETA V から得られた映像をに対して Lightweight OpenPose を使い人物の足元位置の推定を行う。図 26 はこのシステムの流れを表す図である。

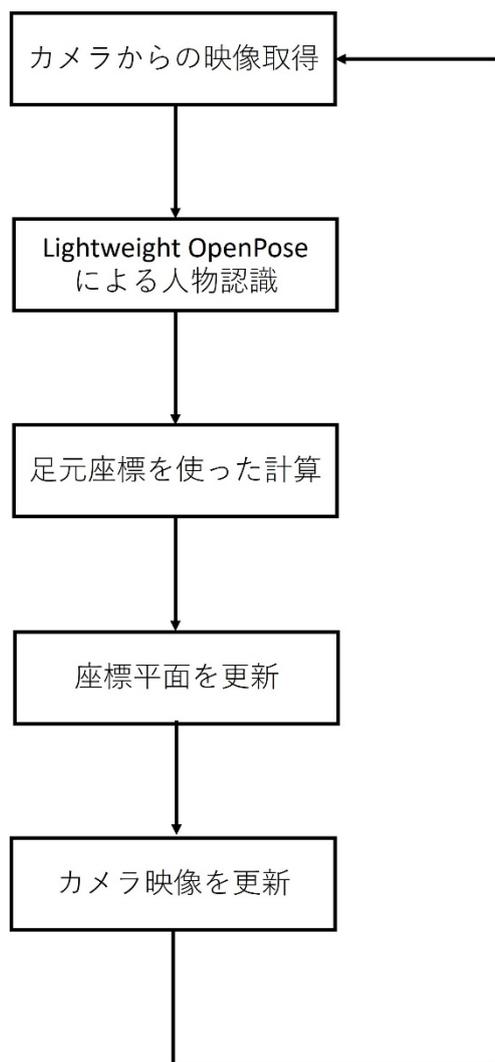


図 26 Lightweight OpenPose を使った人物位置推定システムの流れ

### 4.3.2 実験環境

本実験も、4.2.2 で述べた八王子キャンパスの4号館8階のラウンジで行った。

### 4.3.3 実験方法

本実験では屋内にて1.0mの高さに固定した全天球カメラを基準としてメジャーを設置し、カメラ位置から各位置で静止した人物をライブストリーミングで撮影し、リアルタイムでの人物位置の測定処理を行った。

認識対象者が1名の場合は人物を室内の目標地点に立たせ人物の認識が終わったところでPCのスクリーンショットを撮影する。それをカメラ座標のZ軸を基準とした人物の足元の方位角 $\theta_2$ が $0^\circ \sim 180^\circ$ までの範囲、つまりX座標 $0 \sim 3$ 、Z座標 $-3 \sim 3$ までの原点を除いた計27点で行った。図27はX座標2、Z座標-1の位置に認識対象者が立っており、足元座標が取得されている。

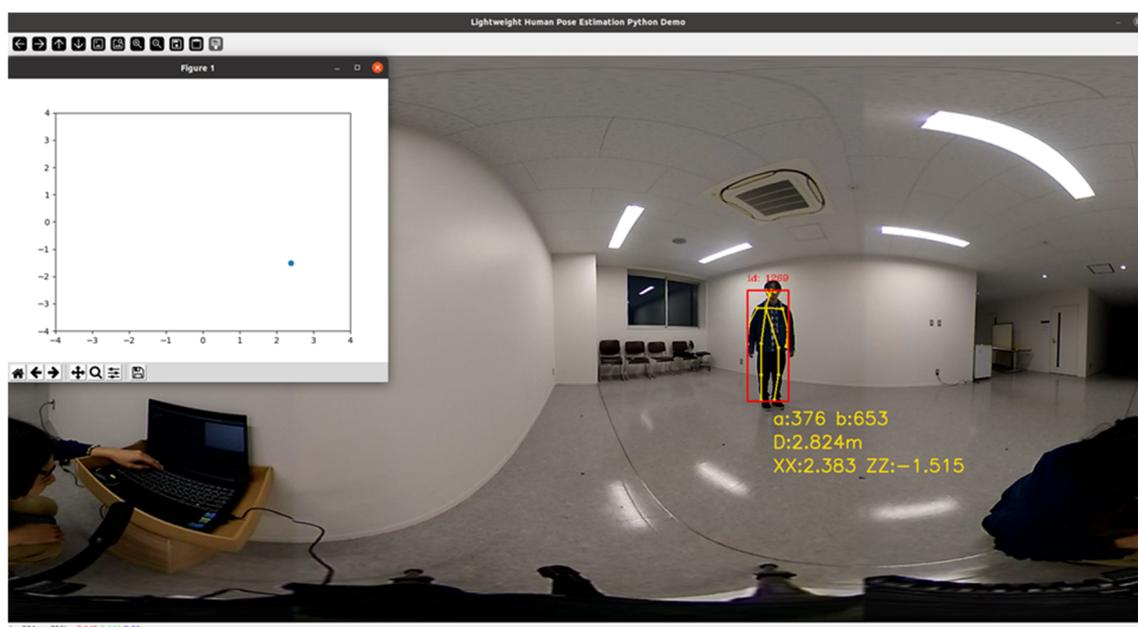


図 27 Lightweight OpenPose を使ったプログラムの動作例

認識対象者が2名の場合の実験では表6に表す5つのパターンで実験をした。ただし、全天球カメラは部屋の中央に置かれており、4.2及び4.3の認識対象者が1名のときのカメラ位置とは異なる。

表6 複数人物の足元座標を取得する実験時の人物の立ち位置

	人物1のX座標	人物1のZ座標	人物2のX座標	人物2のZ座標
パターン1	-1	-1	-1	1
パターン2	1	-2	-2	1
パターン3	-1	1	-2	1
パターン4	-1	1	-2	2
パターン5	-1	1	-2	3

#### 4.3.4 認識対象者が1人の場合の実験結果

認識対象者1名の場合の実験で得られた足元座標を表7に示す。

表7 Lightweight OpenPose で得られた足元座標( $X, Z$ ) (1名の場合)

X	X'	Z	Z'
3.431	3	2.892	3
2.602	2	3.656	3
1.277	1	3.082	3
0.095	0	3.635	3
3.984	3	2.643	2
2.692	2	2.376	2
1.452	1	2.422	2
0.284	0	2.269	2
3.85	3	1.251	1
2.752	2	1.301	1
1.408	1	1.187	1
0.192	0	1.263	1
3.926	3	-0.296	0
2.619	2	-0.249	0
1.28	1	-0.194	0
3.698	3	-1.647	-1
2.383	2	-1.515	-1
1.013	1	-1.454	-1
-0.117	0	-1.43	-1
3.625	3	-2.763	-2
2.593	2	-3.036	-2
1.187	1	-2.84	-2
-0.139	0	-2.821	-2
3.796	3	-4.564	-3
3.151	2	-5.031	-3
1.035	1	-3.914	-3
-0.337	0	-4.475	-3

また、表 7 を基に全天球カメラを原点とした人物の立ち位置を図 28 に表す。これまでの図 15 や図 25 と比べると精度が低いことがわかる。

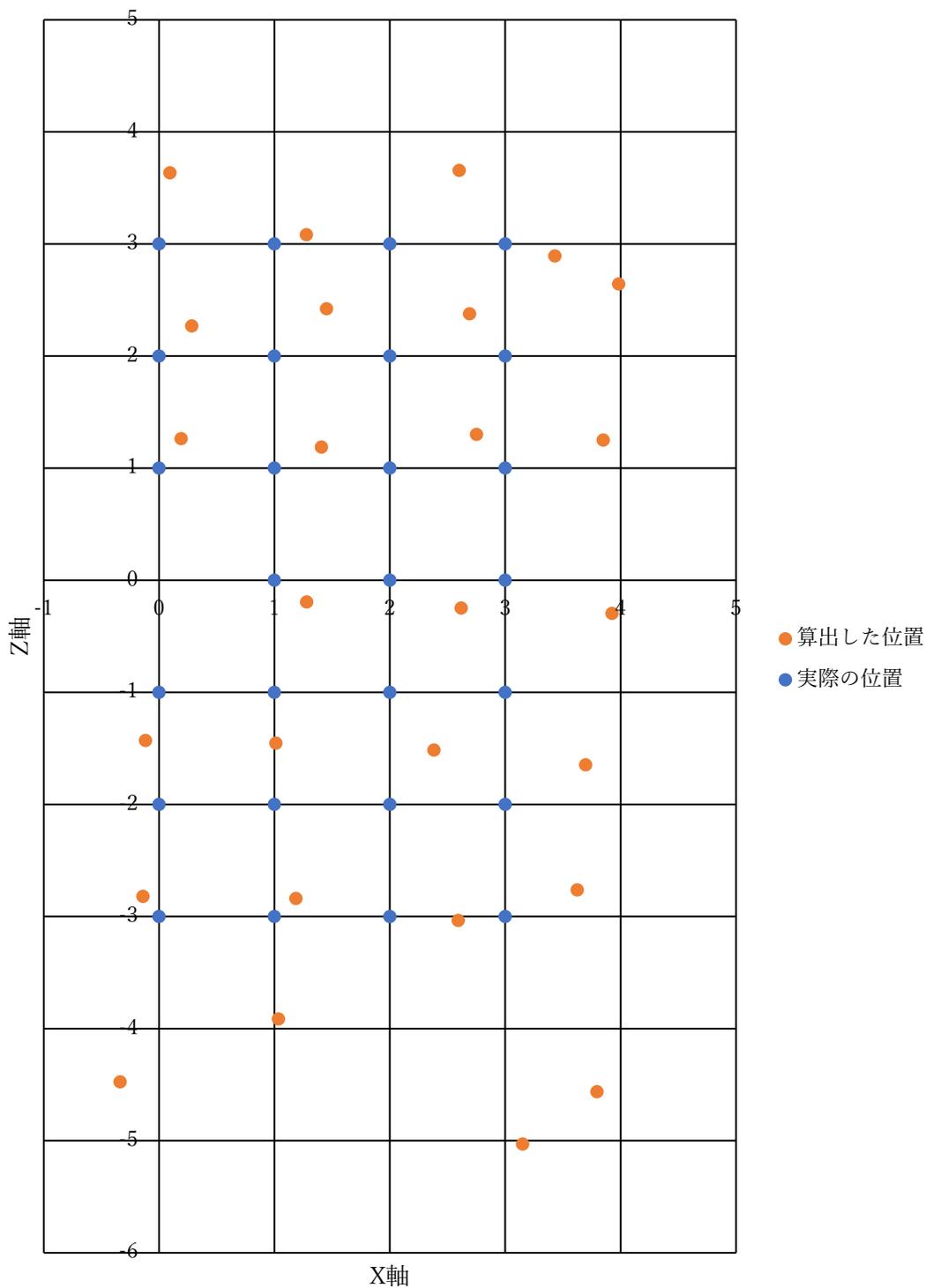


図 28 算出した人物位置( $X, Z$ )と実際の人物位置( $X', Z'$ )(1名の場合)

#### 4.3.5 認識対象者が2人の場合の実験結果

各パターンの足元座標は表 8～表 12 に、人物の立ち位置は図 29～図 38 に表す。  
パターン 1

表 8 Lightweight OpenPose で得られた足元座標( $X, Z$ ) (パターン 1)

	X	X'	Z	Z'
人物 1	-1.353	-1	-1.081	-1
人物 2	-1.498	-1	1.037	1

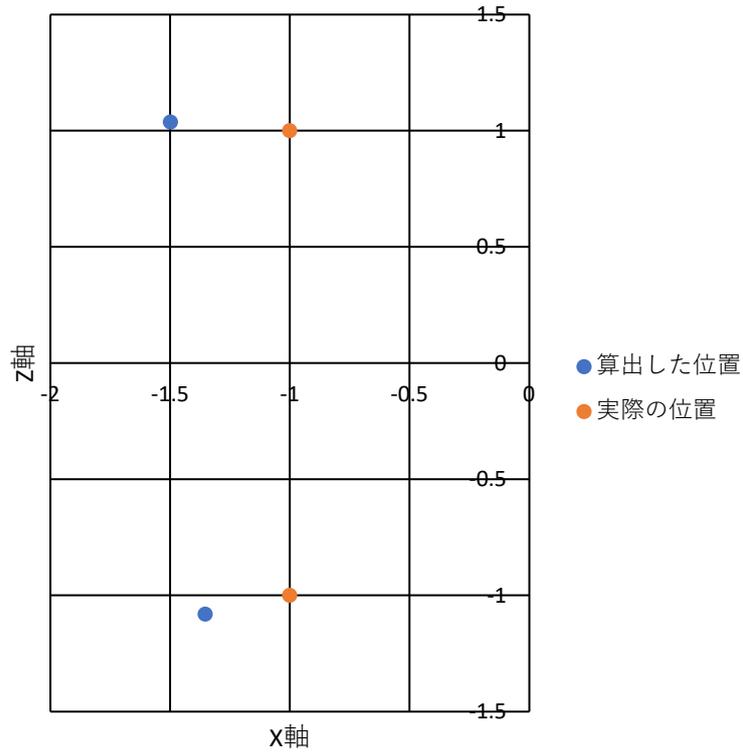


図 29 算出した人物位置( $X, Z$ )と実際の人物位置( $X', Z'$ ) (パターン 1)



図 30 2名の人物認識(パターン 1)

パターン 2

表 9 Lightweight OpenPose で得られた足元座標(X, Z) (パターン 2)

	X	X'	Z	Z'
人物 1	1.202	1	-2.797	-2
人物 2	-2.351	-2	1.122	1

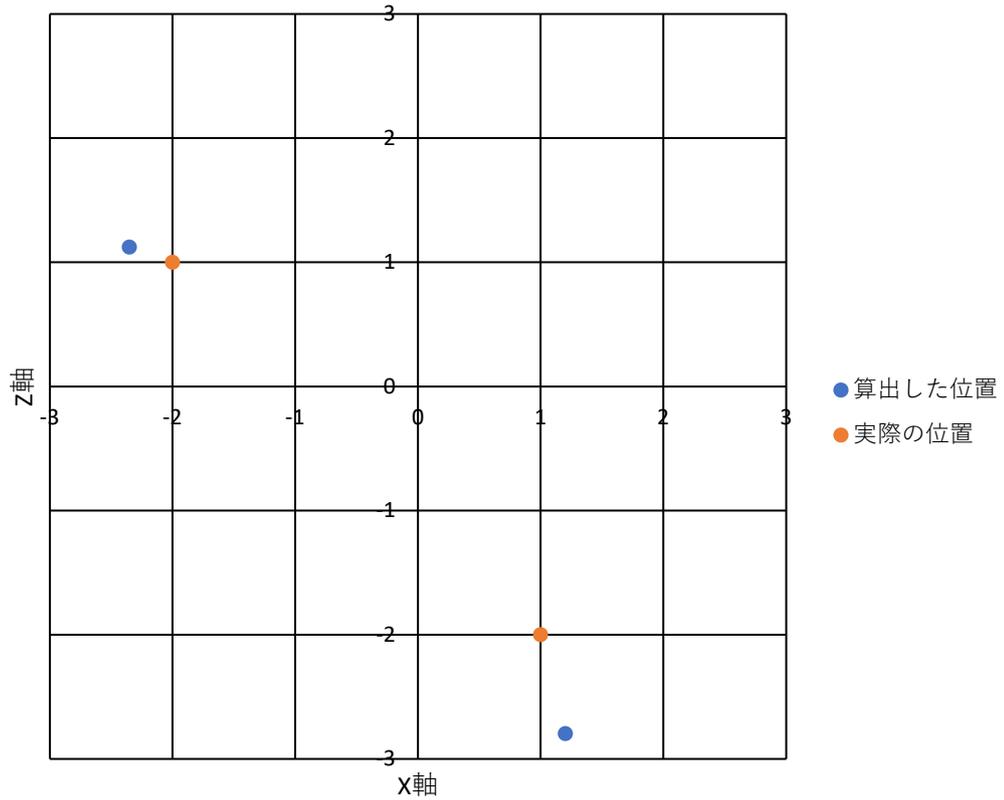


図 31 算出した人物位置(X, Z)と実際の人物位置(X', Z') (パターン 2)



図 32 2名の人物認識(パターン 2)

パターン 3

表 10 Lightweight OpenPose で得られた足元座標( $X, Z$ ) (パターン 3)

	X	X'	Z	Z'
人物 1	-1.262	-1	1.092	1
人物 2	-2.351	-2	1.001	1

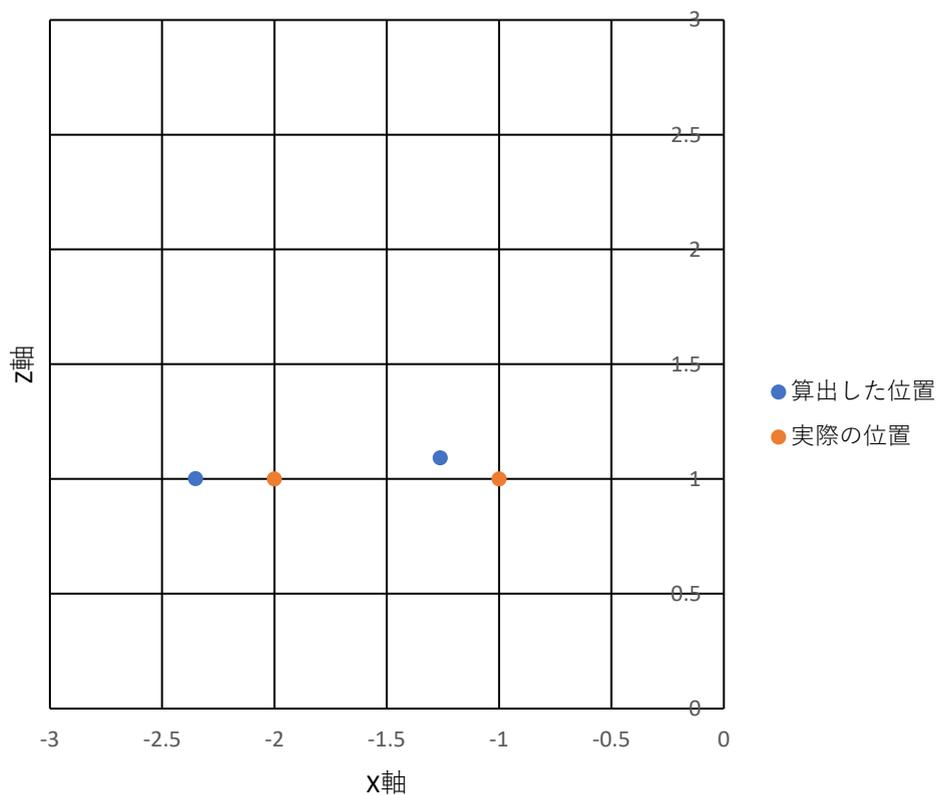


図 33 算出した人物位置( $X, Z$ )と実際の人物位置( $X', Z'$ ) (パターン 3)

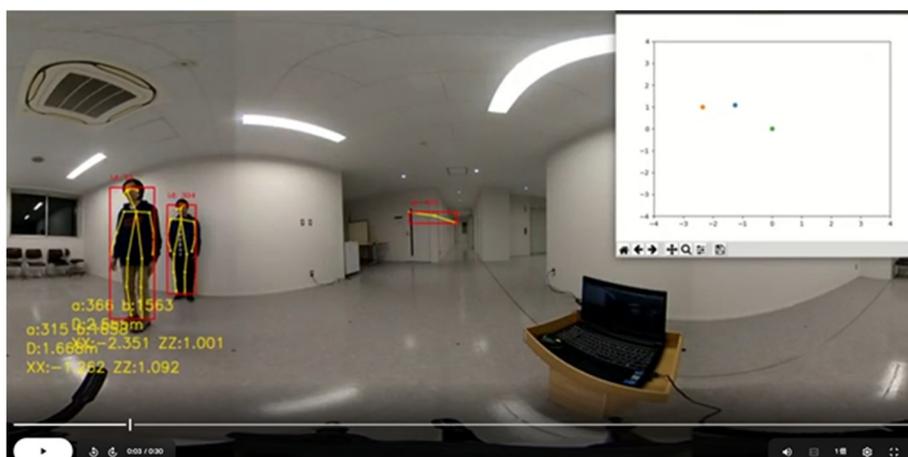


図 34 2名の人物認識(パターン 3)

パターン 4

この時は人物2が人物1の陰に完全に隠れてしまい人物2の骨格点を取得することができなかった。

表 11 Lightweight OpenPose で得られた足元座標( $X, Z$ ) (パターン 4)

	X	X'	Z	Z'
人物 1	-1.154	-1	1.103	1
人物 2	-	-2	-	2

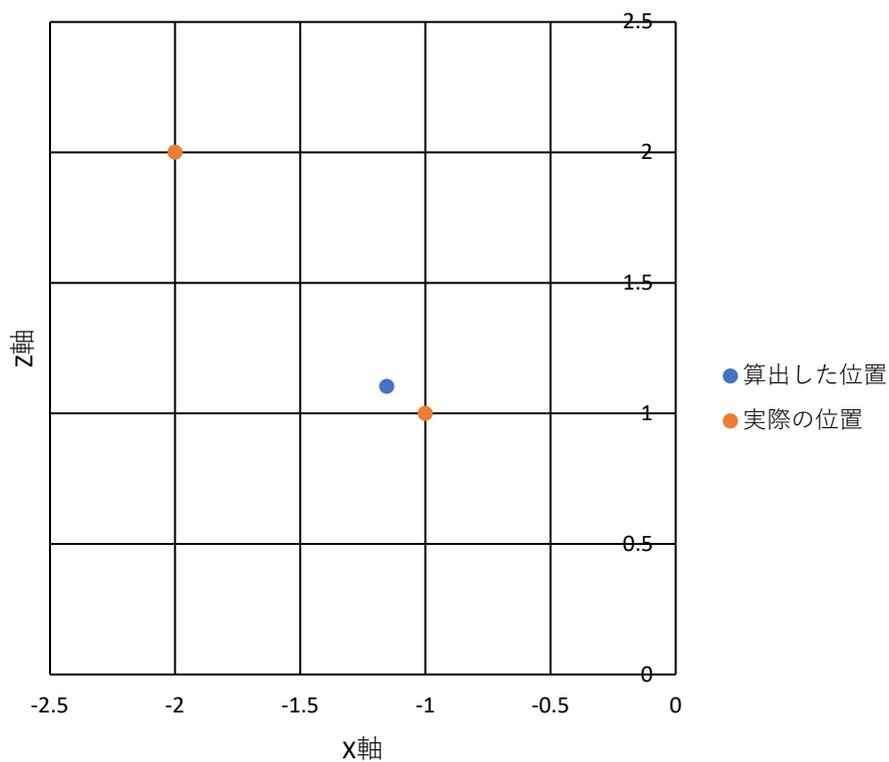


図 35 算出した人物位置( $X, Z$ )と実際の人物位置( $X', Z'$ ) (パターン 4)



図 36 2名の人物認識(パターン 4)

パターン5

表 12 Lightweight OpenPose で得られた足元座標( $X, Z$ ) (パターン5)

	X	X'	Z	Z'
人物 1	-1.163	-1	1.111	1
人物 2	-2.211	-2	3.53	3

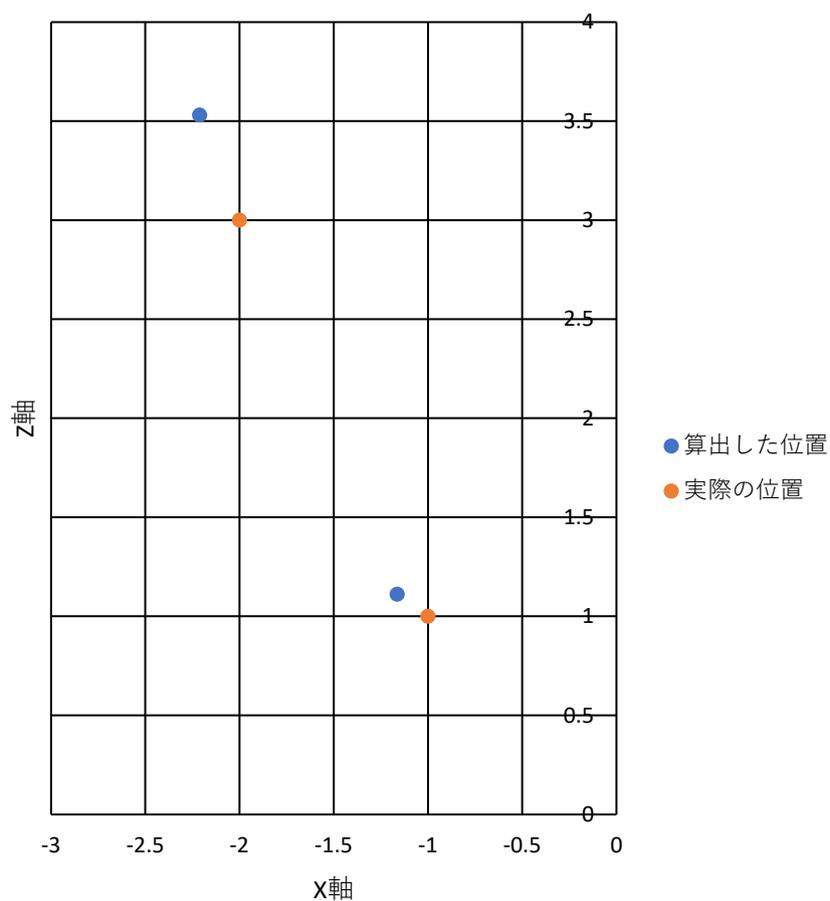


図 37 算出した人物位置( $X, Z$ )と実際の人物位置( $X', Z'$ ) (パターン5)

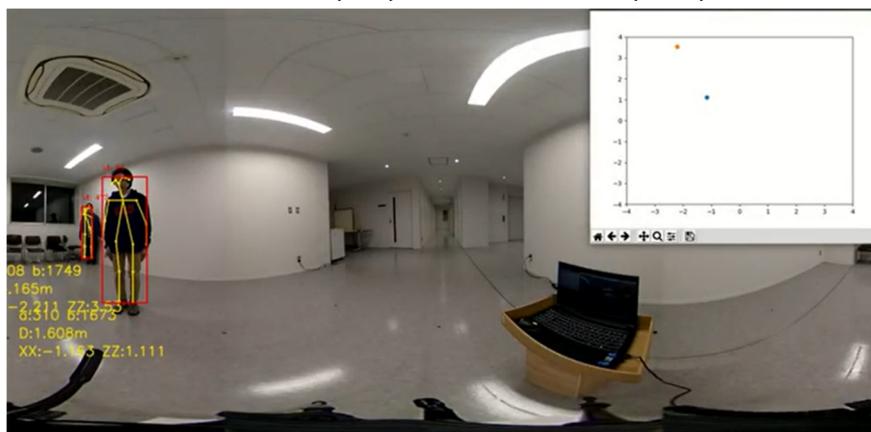


図 38 2名の人物認識(パターン5)

#### 4.4 考察 (勝山)

認識対象者が1名の場合は、2つの人物認識モデルを使い、人物位置推定手法を用いて認識を行った。図 39 は前述の MediaPipe を用いて求めた人物位置(図 25)と Lightweight OpenPose を用いて求めた人物位置(図 28)を1つの図に統合したものである。

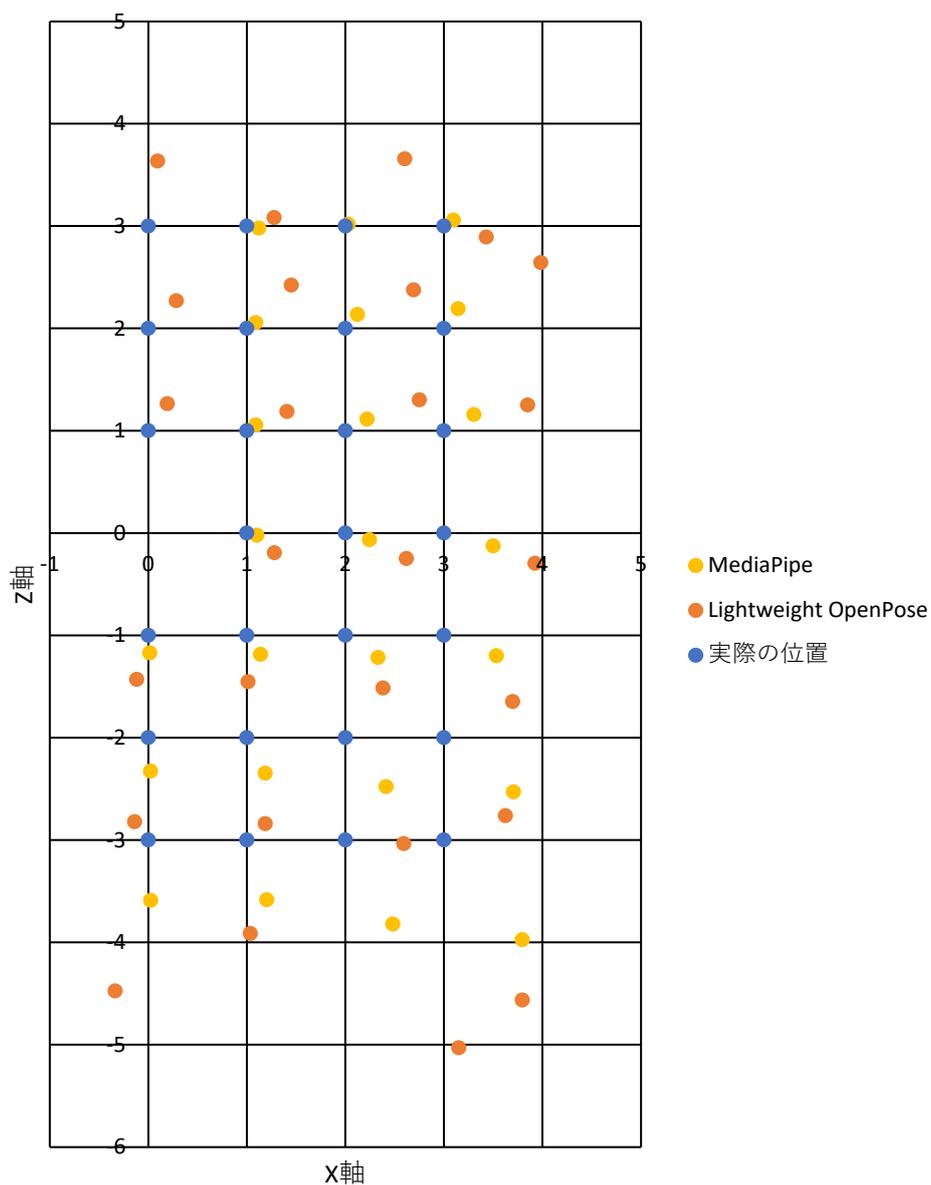


図 39 MediaPipe と Lightweight OpenPose で算出した人物位置(X,Z)

この図から MediaPipe を使って求めた足元座標の方が Lightweight OpenPose を使ったものよりも実際の位置に近い値が出る傾向にあると言える。この要因としては、2つの骨格認識モデルで計算に用いるキーポイントの違いが考えられる。

図 40 は MediaPipe での認識、図 41 は Lightweight OpenPose での認識をしている写真のうち足元を拡大したものだが、MediaPipe の方ではつま先のランドマークが与えられている一方で、Lightweight OpenPose では足首のランドマークのみ与えられている。本来であれば人物の足先と地面の接地点を用いて計算する必要があるが、骨格認識モデルで得たキーポイントはそれよりも画面奥に置かれている場合が多く、Lightweight OpenPose ではその差が特に大きい。このことが、Lightweight OpenPose の方が実際よりも原点から遠くに立っていると認識する傾向がみられる原因ではないかと考えている。



図 40 MediaPipe 動作時の足元拡大図



図 41 Lightweight OpenPose 動作時の足元拡大図

4.3.5 の認識対象者が 2 名の場合の実験は人物の重なりが発生しないパターン 1～3 と重なりが発生するパターン 4,5 を撮影した。人物の重なりが起こる場合はロボットと誘導対象の間を人が横切る場合を想定している。

パターン 1～3 の場合では複数の人物を同時に認識する際に特に誤認識がなく、1 名の場合と異なる挙動をすることは無かった。パターン 4 のように片方の人物がもう片方の人物に隠れてしまう場合は Lightweight OpenPose で検出することは不可能であった。

パターン 5 の場合図 38 のように後ろの人物の足元を正確に求めることができているフレームがある一方で、図 42 のように前の人物が重なってしまい骨格推定がされていないフレームも存在した。

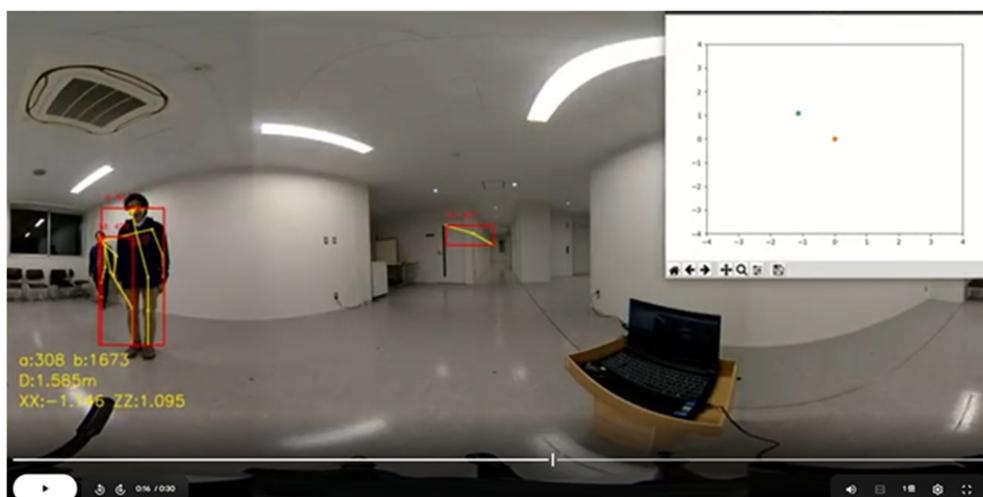


図 42 重なりによって骨格推定ができない例

## 5. 結言

### 5.1 結論 (勝山)

本研究では適切な距離を保ちながら誘導や案内を行う移動ロボットのための人物位置推定手法を検証した。実験では骨格認識モデルから求めた足元座標を基に全天球カメラを用いた人物位置推定を行い、全天球カメラから半径3m程度の範囲までは手法が十分に有効であると分かった。その一方で、足元座標の取得方法によっては大きな誤差が出てしまうこともあり、改善が必要であるとも言える。

## 5.2 課題（向野）

MediaPipe を用いて推定した人物位置推定の実験結果から、推定された位置には一定の傾向をもつ誤差が確認され、カメラから遠ざかる方向へズレが生じる傾向が見られた。図 25 より、 $X$ 座標が正、 $Z$ 座標が負のとき、特にズレが大きく見える。

この結果は、本研究で先行研究として紹介した Masuda 氏らの研究[8]において報告されている誤差傾向と似ており、全天球カメラ画像を用いた位置推定において一般的に生じる傾向と考えられる。Masuda 氏らの論文では、正距円筒図法によって出力された全天球画像において、幾何学的な歪みが発生し、位置推定誤差の主な要因となることが指摘されている。本研究においても、人物位置推定をする過程で、こうした歪みの影響を受け、 $X$ および $Z$ 座標方向にズレが生じたと考えられる。

この問題に対する改善策として、誤差補正を行うことが精度向上につながると先行研究により示されている。Masuda 氏らの研究では、全天球画像を用いた位置推定において、歪みによる誤差が発生することを指摘した上で、実測値と推定値を事前に測定し、その誤差に基づいて補正を行う手法を提案している。また、この補正処理を適用することで、位置推定精度が大幅に向上することが報告されている。本研究における人物位置推定においても、全天球画像の歪みや推定結果の偏りを考慮した誤差補正を行うことで精度が向上すると考えられる。

### 5.3 応用（向野）

本研究の応用例として、当研究室の波山により行われている「動的障害物の存在を考慮した頑強な自己位置推定」が挙げられる。

この研究は、動的環境下におけるロボットの自己位置推定の頑健性向上を目的としている。一般的に移動ロボットは、測域センサから取得した周囲の障害物との距離情報を、あらかじめ作成しておいた地図と照合することで自己位置を推定する。しかし、地図に存在しない障害物（人物など）が周囲に出現すると、測域センサから得られる距離情報と地図との対応付けが正確に行えず、位置推定が不安定になる場合がある。

この問題を解決するためには、人物などを事前に検出し、それらに対応する測域センサの距離情報を自己位置推定から除外することが有効であると考えられる。

そこで、本研究で用いた全天球カメラによる人物位置推定手法を、この研究に応用することで、人物の存在する方向を特定することができる。それにより、人物の影響を低減し、自己位置推定を安定化できると期待できる。

## 参考文献

[1]東京スカイツリー (R) で初のロボットによるお客さま案内誘導の実証実験を実施 | 2024 年 | KDDI 株式会社

<https://biz.kddi.com/topics/2024/08/7505/>

[2]歴史・沿革 | 企業情報 | すかいらくホールディングス

<https://corp.skylark.co.jp/about/history/>

[3]製品紹介 | SEQSENSE (シークセンス) | Security Robot System

<https://www.seqsense.com/product>

[4]日本の自律移動ロボット市場規模は 2033 年までに 10 億 8000 万ドルに達すると予測 - CNET Japan

<https://japan.cnet.com/release/31107287/>

[5]山倉 和樹, 今井 順一, (2018), “動的環境下において複数人の同行者を誘導する自律移動ロボット” 知能と情報 30 巻 5 号: 725-733.

[6] Carballo, A., Ohya, A., & Yuta, S. (2009), "Multiple People Detection from a Mobile Robot using Double Layered Laser Range Finders." Proceedings of the IEEE ICRA 2009 Workshop on People Detection and Tracking, Kobe, Japan, May 2009.

[7]村岡 叶夢, 窪田 諭, 安室 喜弘, (2023), “単眼全天球カメラを用いた自己位置推定と人の位置情報のマッピング” 土木学会論文集 79 巻 22 号: 22017.

[8] Masuda, J., Hayashida, K., & Yokoyama, S., (2021), “Position Estimation for Objects in the High Latitude Region of 360° Panoramic Images”, Proceedings of the 13th International Conference on Management of Digital EcoSystems, MEDES 2021, 1-7.

[9]: Google AI Edge. "MediaPipe ソリューション ガイド"

[https://ai.google.dev/edge/mediapipe/solutions/vision/pose\\_landmarker?hl=ja](https://ai.google.dev/edge/mediapipe/solutions/vision/pose_landmarker?hl=ja)

[10] Kazuhito00/mediapipe-python-sample, GitHub

<https://github.com/kazuhito00/mediapipe-python-sample/>

[11] Daniil-Osokin/lightweight-human-pose-estimation.pytorch, GitHub  
<https://github.com/Daniil-Osokin/lightweight-human-pose-estimation.pytorch?tab=readme-ov-file>

[12] Daniil Osokin (2018), “Real-time 2D Multi-Person Pose Estimation on CPU: Lightweight OpenPose”  
<https://arxiv.org/abs/1811.12004>

[13] ricohapi/libuvc-theta, GitHub  
<https://github.com/ricohapi/libuvc-theta?tab=readme-ov-file>

[14] codetricity/libuvc-theta-sample, GitHub  
<https://github.com/codetricity/libuvc-theta-sample>

## 謝辞

本論文の完成にあたり、研究テーマの設定から研究全体を通して論文完成に至るまで、終始変わらぬご指導をいただきました指導教員の金丸隆志教授に深く感謝申し上げます。

また、日々の研究作業から論文執筆に至るまで、多大なご助力とご助言をいただきました波山先輩に深く感謝いたします。

さらに、多くのアドバイスや気づきを与えてくれた研究室の皆様、機械理工学科の皆様にも心より感謝いたします。

最後に、ここまで生活を支えてくれた家族に大変感謝いたします。

ありがとうございました。