

3D ゲームエンジンを用いた全周魚眼カメラにおける人物検出と位置推定用データセットの作成と評価

秋田悠河¹ 小宮嗣朗¹ 阿倍博信¹

概要: 新型コロナウイルスの流行により、感染対策として密閉・密集・密接の回避が求められている。そのため、映像解析技術を用いて人物間の密集および密接を監視するシステムが求められている。対象システムの開発には、大量の全周魚眼画像が必要になるが、本論文では3D ゲームエンジンであるUnityを用いることで人物検出と位置推定のためのデータセットの作成作業の効率化に取り組むとともに、その有効性の評価について報告する。

キーワード: 映像監視システム, 人物検出, 位置推定, 全周魚眼, データセット

Creation and evaluation of dataset for person detection and position estimation in full-dome fisheye camera using 3D game engine

YUGA AKITA^{†1} SHIRO KOMIYA^{†1} HIRONOBU ABE^{†1}

Abstract: Due to the epidemic of the new coronavirus, it is required to avoid sealing, crowding, and close contact as an infection control measure. Therefore, there is a demand for a system for monitoring crowding and closeness between people using video analysis technology. A large amount of all-around fisheye images are required to develop the target system, but in this paper, we use Unity, which is a 3D game engine, to improve the efficiency of data set creation work for person detection and position estimation. As we work on it, we will report on its effectiveness evaluation.

Keywords: Video surveillance system, person detection, position estimation, all-around fisheye, dataset

1. はじめに

新型コロナウイルスの流行により、感染症対策の要点として3密を回避する社会の課題が存在している。この3密とは密閉・密集・密接を指している。つまり屋内においてこの3密を回避する必要がある。特に密集・密接においては各個人での判断で屋内の状況を適切に判断することは難しい。そのため密集・密接の状態を定量的に分析することが可能であれば屋内の状況を評価することが可能である。具体的には屋内を対象とした密集度の計測が必要であると考えられる。密集度の計測を自動でかつ低コストでなるべく広い範囲で行おうとすると魚眼カメラを利用した映像監視システムと解析を担う機械学習の実装が必要である。

しかし、魚眼カメラから得られた映像に対して人物検出や位置推定を機械学習によって実現するためには、学習に必要な全周魚眼画像データセットが必要になる。大量かつバリエーション豊富なラベル付きデータセットの作成を実写で行うには多大なコストや手間がかかる。一方、3D ゲームエンジンを利用することができれば、必要なモデルとスクリプトを用意することで大量かつ豊富なバリエーションのデータセットの自動生成が可能であると考えられる。またアノテーション作業も自動化しやすいと考えられる。本

論文では、3D ゲームエンジンを利用した人物検出及び位置推定のためのデータセット作成とその有効性の確認を行う。

2. 関連研究

関連研究として、魚眼画像上の人物検出に関するものとゲームエンジンを利用したデータセット作成の研究が存在する。

青木による視覚センサによる見守り技術について[1]という研究では、深層学習であるFaster R-CNN[2]を用いた広角レンズ上の人物検出を行っている。広角レンズからの映像を深層学習で学習させ広角レンズ画像での人物検出が可能であると示しているが具体的なデータセットと評価手法について不明である。

一方ゲームエンジンを利用したデータセット作成の研究では、磯井らによる機械学習向けに生活空間動画データセット構築の検討[3]が存在している。ゲームエンジンを利用して生活空間内の人物検出及び動作を分類するためのデータセットを作成している。それを用いて機械学習を行い動作分類の有効性について確認を行っている。しかし、魚眼画像を目的とした3D ゲームエンジンの利用によるデータセット作成及び人物検出の評価ではない。

位置推定に関する研究では魚眼カメラを利用した研究が存在する。松尾による研究[4]では、機械学習的手法ではなく、魚眼カメラの高さが既知と設定した上で数式モデルを用いて位置推定を行う方法を取っている。一方花崎らの

¹ 東京電機大学
Tokyo Denki University

研究[5]では複数の魚眼カメラから得られた背景差分を用いて最小 20cm の誤差で位置推定を行っている。

以上の研究より本研究では深層学習による魚眼画像からの人物検出、位置推定及びそれに必要なデータセットの作成に注目をして研究を進めることになる。

3. データセット作成方式の提案

3.1 基本方針

本研究では、魚眼画像上での機械学習を用いた人物検出と位置推定のためのデータセット作成を目的として、3D ゲームエンジンを用いた合成全周魚眼画像データセットの作成を行う。人物検出は物体検出タスクであり機械学習に必要な学習データとして全周魚眼画像を作成する。位置推定は最終的に数値データを用いて回帰的に機械学習を行うが、座標データに変換するための画像も作成する。

3.2 3D ゲームエンジン内での屋内の再現

3D ゲームエンジン内に実際のオフィスを再現している UnityJapanOffice[6]を配置して部屋の天井に魚眼カメラを下向きに設置し、そこに人物モデル・魚眼カメラを配置する。オフィスには複数の部屋が存在しており、光量の調整は3段階が可能である。使用する3D ゲームエンジンはUnity[7]である。描画を行うレンダリングパイプラインは最高解像度で描画が可能な High definition Render Pipeline を採用する。人物や建物のモデル、魚眼カメラについてはフリーのアセットを使用する。使用したゲームエンジンとアセットについては表1にまとめる。

表1 ゲームエンジンと使用したアセット

Table 1 Game engine and assets used

Item	Content
3D game engine	Unity ver2019.24fl
Rendering pipeline	High Definition Render Pipeline
Building model	UnityJapanOffice
Human model	6 types
Fish eye camera	FullDomeCameraForUnity [8]

3.3 人物検出用データセットの作成

人物検出用データセットの作成にあたって魚眼画像の生成とアノテーションの2つの段階がある。魚眼画像の生成では屋内の状況を再現している学習用画像と魚眼画像のどの部分が人物モデルかを示すために人物モデルと黒い背景のみの Ground Truth 画像の2種類の画像を生成する。その後生成した画像を用いてアノテーションを行う。生成する学習用画像と Ground Truth 画像を図1に示す。学習用画像生成の場合は以下の方法で行う。

- Unity 内にある建物モデルの中の目的の部屋に人物モデルと魚眼カメラを配置する。必要であれば日光を調整する。
- 人物モデルの座標と向きを動かすためのスクリプト

を準備して実行可能状態にする。

- Unity Recorder[9]を使い、ゲームプレイ状態に移行してスクリプトを自動実行してフレーム毎に画像を生成する。

次に Ground Truth 画像生成の場合は以下の方法で行う。

- 学習用画像時と魚眼カメラの位置が同じになる様に Unity 内に人物モデルと魚眼カメラを配置する。
- 建物モデルは使用せず、また人物モデルの全ての部分を同一モデルでは一色になるようにテクスチャを変更する。複数モデルを同時に使用して生成する場合は他のモデルと違う色に設定する。
- 学習用画像生成時と同じ設定で Unity Recorder を使い、ゲームプレイ状態に移行してスクリプトを自動実行してフレーム毎に画像を生成する。

次にアノテーションについて説明する。Ground Truth 画像を用いて学習用画像のどの部分が該当の人物モデルの部分かの情報を矩形座標にして保存する。保存する形式は COCO フォーマット[10]と呼ばれるフォーマットで保存する。アノテーションを行う際は Python を用いて行う。その際の簡単なアルゴリズムを以下に整理する。

- 関連している学習用画像と Ground Truth 画像を読み込む。
- Ground Truth 画像においてあらかじめ決められた RGB 値を検索して同じ RGB 値ごとにピクセル座標をまとめる。
- それぞれまとめたピクセル座標から矩形座標を計算して学習用画像と Ground Truth 画像のファイル名と共に COCO フォーマットでファイルに追加・保存する。

以上で人物検出用データセット作成の基本的な方法をまとめた。人物モデルの動かし方は、本研究の前の3D ゲームエンジンを用いた全周魚眼カメラ用映像監視向けデータセットの作成と評価[11]という研究において座標と向きを乱数で動かすランダム方式を本研究では採用した。また一定値ごとに動かすシナリオ方式の2つで作成している。また学習用画像のサイズは縦：横で 500:500 Pixel であり、人物モデルの種類は14種類で部屋のサイズは 5×5m である。

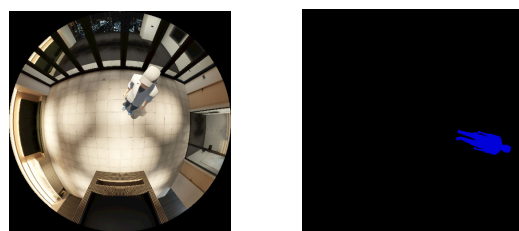


図1 学習用画像(左)と Ground Truth 画像(右)

Figure 1 Image for learning(left) and Ground Truth image(right)

3.4 位置推定用データセットの作成

位置推定用データセット作成には Ground Truth 画像の生成とアノテーションの2段階がある。Ground Truth 画像の生成では、球体モデルと黒い背景のみの Ground Truth 画像と Unity 内での球体モデルの座標を生成・出力する。アノテーションの段階では Ground Truth 画像を用いて魚眼画像上の球体モデルのピクセル座標を計算して Unity 内との座標と関連付けを行う。Ground Truth 画像の生成の場合は以下の方法で行う。

- 建物モデルは使用せず魚眼カメラが人物検出用データセット作成時と同じ条件になる様に配置する。
- 球体モデルを Unity 内に配置する。球体モデルのテクスチャは反射せず全ての部分が同じ RGB で出力される様な設定で一色にする。
- スクリプトで球体モデルの Unity 内での座標を CSV 形式で出力できる様にする。
- Unity Recorder を使い、ゲームプレイ状態に移行してスクリプトを実行してフレーム毎に画像を生成する。この際 CSV ファイルに球体モデルの Unity 内の座標が出力される。

次にアノテーションについて説明する。Ground Truth 画像から球体モデルの矩形座標の中心を球体モデルの魚眼画像上のピクセル座標とする。アノテーションを行う際は Python を用いて行う。その際の簡単なアルゴリズムを以下に整理する。

- Ground Truth 画像においてあらかじめ決められた球体モデルの RGB 値を検索して RGB のピクセル座標をまとめる。
- まとめたピクセル座標から矩形座標を計算して更に中心のピクセル座標を CSV 形式のファイルに追加・保存する。ピクセル座標ファイルと Unity 内の座標ファイルを合わせて位置推定用データセットとする。

以上で位置推定用データセット作成の基本的な方法をまとめた。本研究では、球体モデルの動かさずとして乱数を利用したランダム方式を採用した。

4. 評価

4.1 人物検出用データセットによる人物検出機能の評価

Mask R-CNN[12]という物体検出アルゴリズムで学習を行い、人物検出モデルを作成する。学習をする際は COCO データセット[13]を学習したパラメータ[14]を用いて転移学習を行う。評価方法として N 人だけ写っている画像を用意して評価を行う。N は 1~3 の三段階の画像を用意して合計 290 人検出可能なテスト用データセットを作成して評価を行う。検出の際は検出時の矩形のスコア値の閾値を 0.5 と 0.7 の 2 種類で行う。評価の指標として mIoU と AP, Recall と Precision を採用した。mIoU の計算方法として検

出できた矩形に対して重なり具合を算出して合算したものを検出数で平均を算出する。評価を行ったデータセットの内容が表 2 で学習曲線が図 2, 推論結果が表 3~8 である。

表 2 人物検出用データセット

Table 2 Dataset for person detection

Specification items	Scenario	Scenario: Random	Random	COCO
Number of images	8500	8500	8500	330K
Person model pose	Walking	Walking	Walking	great variety
Type of person model	14	14	14	great variety
How to move	Scenario	Scenario: random	random	great variety
Room size(m)	5×5	5×5	5×5	great variety

表 3 スコア閾値 0.5 で IoU=0.7 人物検出結果

Table 3 Person detection result when score threshold is 0.5 and

IoU = 0.7

	mIoU	AP	Recall	Precision
Scenario	0.566	0.276	0.144	0.159
Scenario:random	0.648	0.538	0.348	0.417
Random	0.616	0.343	0.293	0.352
COCO	0.795	0.928	0.52	0.843

表 4 スコア閾値 0.5 で IoU=0.5 人物検出結果

Table 4 Person detection result when score threshold is 0.5 and

IoU = 0.5

	mIoU	AP	Recall	Precision
Scenario	0.566	0.792	0.648	0.712
Scenario:random	0.648	0.935	0.72	0.863
Random	0.616	0.894	0.679	0.817
COCO	0.795	0.961	0.562	0.91

表 5 スコア閾値 0.5 で IoU=0.3 人物検出結果

Table 5 Person detection result when score threshold is 0.5 and

IoU = 0.3

	mIoU	AP	Recall	Precision
Scenario	0.566	0.952	0.855	0.939
Scenario:random	0.648	0.974	0.793	0.95
Random	0.616	0.964	0.755	0.908
COCO	0.795	0.976	0.586	0.949

表 6 スコア閾値 0.7 で IoU=0.7 人物検出結果

Table 6 Person detection result when score threshold is 0.5 and IoU = 0.7

	mIoU	AP	Recall	Precision
Scenario	0.57	0.289	0.141	0.166
Scenario:random	0.646	0.555	0.358	0.435
Random	0.638	0.33	0.3	0.373
COCO	0.816	0.957	0.486	0.875

表 7 スコア閾値 0.7 で IoU=0.5 人物検出結果

Table 7 Person detection result when score threshold is 0.5 and IoU = 0.5

	mIoU	AP	Recall	Precision
Scenario	0.57	0.827	0.62	0.731
Scenario:random	0.646	0.944	0.724	0.878
Random	0.638	0.907	0.686	0.854
COCO	0.816	0.977	0.52	0.937

表 8 スコア閾値 0.7 で IoU=0.3 人物検出結果

Table 8 Person detection result when score threshold is 0.5 and IoU = 0.3

	mIoU	AP	Recall	Precision
Scenario	0.57	0.983	0.817	0.963
Scenario:random	0.646	0.984	0.786	0.953
Random	0.638	0.974	0.762	0.948
COCO	0.816	0.992	0.537	0.968



図 2 loss(左)と val_loss(右)の学習曲線

Figure 2 Learning curve of loss(left) and val_loss(right)

4.2 人物検出用データセットの改善と評価

人物モデルの魚眼画像上の分布を確認した所、図 3 の通りとなった。x と y において -100~100 の範囲に全体の 20% 前後しか存在していないことが分かった。カメラに近い部分に人物モデルが写っている画像が不足していることが学習不足の原因だと仮定して、対象範囲に 1,400 個データを追加して再度評価を行った。改善したデータセットの内容は表 9 で推論結果は表 10~15 の通りである。また学習曲線は図 4 の通りとなった。

表 10 より Scenario:Random:addition は COCO より各指標は低いですが他のデータセットと比較すると比較的高い値を出

している。表 11 と 12 では Scenario:Random:addition は Recall が比較的低いものの AP は高い。一方 Random:addition は表 12 により Recall は比較的高いものの、結果的には Scenario よりも低い値となっている。表 13~15 においては Scenario:Random:addition の Recall は比較的低いものの AP は COCO を除けば一番高い結果になった。図 4 の学習曲線を見ると loss は下がってはいるが、val_loss は高い値のままであることから、データセットの画像枚数やバリエーションに問題がある可能性があることが分かる。

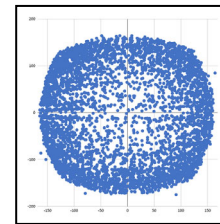


図 3 人物モデルの魚眼画像上の分布

Figure 3 Distribution of human model on fisheye image

表 9 改善された人物検出用データセット

Table 9 Improved dataset for person detection

Specification items	Scenario:Random:addition	Random:addition
Number of images	9900	9900
How to move	Scenario:Random	random

表 10 スコア閾値 0.5 で IoU=0.7 人物検出結果

Table 10 Person detection result when score threshold is 0.5 and IoU = 0.7

	mIoU	AP	Recall	Precision
Scenario:Random :addition	0.668	0.647	0.313	0.439
Random:addition	0.59	0.358	0.227	0.255
COCO	0.795	0.928	0.52	0.843

表 11 スコア閾値 0.5 で IoU=0.5 人物検出結果

Table 11 Person detection result when score threshold is 0.5 and IoU = 0.5

	mIoU	AP	Recall	Precision
Scenario:Random :addition	0.668	0.969	0.658	0.922
Random:addition	0.59	0.852	0.675	0.759
COCO	0.795	0.961	0.562	0.91

表 12 スコア閾値 0.5 で IoU=0.3 人物検出結果

Table 12 Person detection result when score threshold is 0.5 and IoU = 0.3

	mIoU	AP	Recall	Precision
Scenario:Random :addition	0.668	0.986	0.693	0.971
Random:addition	0.59	0.94	0.813	0.914
COCO	0.795	0.976	0.586	0.949

表 13 スコア閾値 0.7 で IoU=0.7 人物検出結果

Table 13 Person detection result when score threshold is 0.5 and IoU = 0.7

	mIoU	AP	Recall	Precision
Scenario:Random :addition	0.673	0.667	0.293	0.461
Random:addition	0.603	0.382	0.224	0.276
COCO	0.816	0.957	0.486	0.875

表 14 スコア閾値 0.7 で IoU=0.5 人物検出結果

Table 14 Person detection result when score threshold is 0.5 and IoU = 0.5

	mIoU	AP	Recall	Precision
Scenario:Random :addition	0.668	0.988	0.589	0.929
Random:addition	0.59	0.897	0.641	0.791
COCO	0.795	0.977	0.52	0.937

表 15 スコア閾値 0.7 で IoU=0.3 人物検出結果

Table 15 Person detection result when score threshold is 0.5 and IoU = 0.3

	mIoU	AP	Recall	Precision
Scenario:Random :addition	0.668	0.996	0.617	0.972
Random:addition	0.59	0.978	0.768	0.948
COCO	0.795	0.992	0.537	0.968

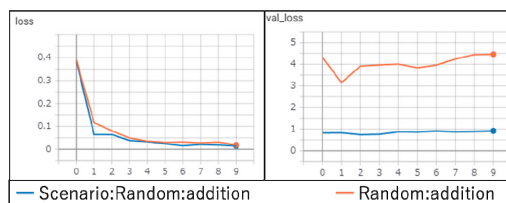


図 4 loss(左)と val_loss(右)の学習曲線

Figure 4 Learning curve of loss(left) and val_loss(right)

4.3 位置推定用データセットによる位置検出機能の評価

作成したデータセットを用いて機械学習を行い、位置推

定モデルを作成する。学習に使うアルゴリズムはランダムフォレストとニューラルネットワークで行う。データセットの座標範囲は x と y 共に -20~20 の範囲である。学習に使用したデータセットの内容は表 16 で推論結果が表 17 と表 18 である。

表 17 より MAE を参考にすると 1,500 個前後が比較的性
能が高い可能性がある。表 18 より学習アルゴリズムとして
はニューラルネットワークの方が比較的適切であることが
分かった。

表 16 位置推定用データセット

Table 16 Dataset for position estimation

	A	B	C	D
Number of coordinates	800	1000	1500	2000

表 17 位置推定の結果

Table 17 The result of position estimation

	A	B	C	D
RMSE(train)	0.287	0.370	0.084	0.084
RMSE(test)	0.287	0.550	0.160	0.160
MAE of x(m)	0.229	0.152	0.08	0.08
MAE of y(m)	0.231	0.151	0.09	0.09
standard deviation	0.015	0.009	0.011	0.009

表 18 2つのアルゴリズムの比較結果

Table 18 The Comparison result of two algorithm

	random forest	neural network
RMSE(train)	0.450	0.370
RMSE(test)	1.143	0.550
MAE of x(m)	0.866	0.152
MAE of y(m)	0.855	0.151
standard deviation	0.015	0.009

4.4 位置推定用データセットの改善と評価

位置推定用データセットの魚眼画像上のデータ分布を
確認した所、x と y において -100~100 の範囲が特にデー
タが少ないことが分かった。この部分に 200 個データを追加
することで、データの偏りを修正したデータセットの場合
の性能の評価を行った。推論結果は表 19 の通りとなった。

表 19 より平均誤差を示す MAE が改善前と比較して改善
後の方が低いことからデータの偏りを修正する改善策が有
効であることが分かった。

表 19 改善された位置推定機能の結果

Table 19 The results of improved position estimation

	Before	After
RMSE(train)	0.370	0.228
RMSE(test)	0.550	0.342
MAE of x(m)	0.152	0.08
MAE of y(m)	0.151	0.08
standard deviation	0.009	0.010

5. 考察

5.1 人物検出機能の評価に対する考察

本研究方式によるデータセットの中で AP だけに着目すると Scenario:Random:addition が一番高く、性能が高い人物検出機能を作成することが可能であると言える。しかし、Recall においては他のデータセットと比較すると全体的に高いとは言えない。屋内の密集度を計測するという観点においてより多くの人物を検出する必要があると考えた場合 Scenario:Random:addition が最適なデータセットであると結論付けることはできない。また Recall だけ着目すると Scenario か Scenario:Random が評価する際の IoU が 0.5 と 0.3 においては比較的高いが、IoU が 0.7 の場合に着目すると Scenario:Random の方が Recall の方が高く、また全体的に見ても AP は Scenario より高く、また低い方に位置している訳ではない。このことから本研究方式のデータセットの中で検出時の信頼性においては Scenario:Random:addition がより最適であり、より人を検出することを考慮すると Scenario:Random が比較的適切であると考えられる。

COCO と比較すると IoU が 0.7 の時は全ての項目で一番高い結果になっているが、IoU が 0.5 及び 0.7 においては本研究方式のデータセットの方が AP や Recall は高いため、より人物を検出したい場合を考慮すると本研究方式のデータセットの方が適切であると考えられる。

5.2 画像に着目した人物検出機能の評価に対する考察

人物検出機能の推論結果に対して、IoU の低くなった画像や全体的に検出できていない部分や特徴のある画像について調査した。その結果、特徴的なケースとして下記の 3 パターンが見つかった。

- ケース 1：体を検出できていても頭を検出できていない、または頭と体を別人物として検出しているケース(図 5)
- ケース 2：人物モデルの影を検出したり、その影も含めて検出してしまうケース(図 6)
- ケース 3：人物モデルの服装の色と背景の色、もしくは他の人物モデルの服装の色が似ているケース、それらを区別できずに検出するケースや、検出できていないケース(図 7)



図 5 ケース 1 の特徴

Figure 5 The features of case 1

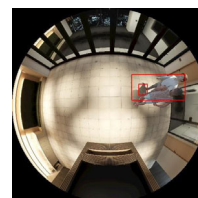


図 6 ケース 2 の特徴

Figure 6 The features of case 2



図 7 ケース 3 の特徴

Figure 7 The features of case 3

以下に、各ケースについて発生原因を考察する。

- ケース 1：
 ケース 1 の特徴である頭の検出が難点であることについては、頭を検出できていない場合や頭だけの検出の場合が存在している。これは人物モデルのバリエーションが少ないために、頭と体のバリエーションが少ないデータで学習をしていることが原因である可能性が考えられる。
- ケース 2：
 ケース 2 の特徴である影を検出している場合が存在することについては、今回学習している人物モデルのバリエーションが少ないことによって人物モデルの一部に似ている影と区別がついていない可能性が考えられる。しかし、これは部屋の明るさを変更するといった画像で学習することで影と区別する様な学習が可能ではないかと考えられる。
- ケース 3：
 ケース 3 の特徴である人物モデルの服装の色と背景の色、もしくは他の人物モデルの服装の色が似ている場合に検出できない場合については、人物モデルのバリエーションの少なさに加えて、部屋の明るさが 1 種類しか無かったことも学習不足の原因である可能性が考えられる。この点については部屋の明るさを変更した画像を用いることで改善することが可能であると考えられる一方で、本研究全体の課題として人物モデルのバリエーションの少なさについても改善した評価が必要になってくる可能性が考えられる。

5.3 位置推定機能の評価に対する考察

本研究の位置推定用データセットを用いて作成した位置推定機能は最小の MAE が 0.08m であることから、本研究方式で十分な性能の位置推定機能の作成が可能であることが考えられる。ランダムフォレストよりもニューラルネットワークの方が MAE は小さいことから魚眼画像からの位置推定機能にはニューラルネットワークの方が比較的最適であることが考えられる。また改善後の 1,200 個のデータセットで高い精度を達成しているためデータ数よりも、むしろ座標の分布具合の調整の方が重要であると考えられる。

6. おわりに

本研究では 3D ゲームエンジンである Unity を用いて合成全周魚眼画像データセットを作成するとともに、魚眼画像に対応した人物検出機能を作成した。3 人までの検出であれば本研究方式による人物検出機能は有効性が高いということが分かった。しかし、推論結果の画像について検証してみたところ、頭の検出や影との区別、同色の物体等に対する検出能力の弱さが存在しているため、人物モデルの種類を増やすことやデータセットの拡張が必要である。

また、同様に Unity を用いて位置推定用データセットを作成するとともに、魚眼画像から実空間内の位置推定を行うための位置推定機能の作成を行った。本研究方式による位置推定機能は MAE が 0.08m となり、小さい誤差で位置推定が可能であることが分かった。更に MAE を小さくする改善の余地としてデータの偏りに注目して作成していく必要がある。

参考文献

- [1] 青木義満: 視覚センサによる見守り技術について, 電子情報通信学会 通信ソサイエティマガジン, Vol. 11, No.1, pp.30-38 (2017).
- [2] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun: Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, <https://arxiv.org/pdf/1506.01497.pdf> (2016).
- [3] 磯井葉那, 竹房あつ子, 中田秀基, 小口正人: 機械学習向け生活空間動画データセット構築の検討, 情報処理学会 第 82 回全国大会 1R-02, pp.243-244 (2020).
- [4] 松尾寿: 魚眼カメラによる歩行者の分布と流れ推定, 奈良先端科学技術大学院大学 情報科学研究科 情報処理学専攻 修士論文 (2005).
- [5] 花崎厚年, 谷口倫一郎, 島田敬士, 内山英昭: 複数の魚眼カメラを用いた人物の位置推定, 平成 28 年度電気・情報関係学会九州支部連合大会 (2016).
- [6] Unity: UnityJapan オフィス, <https://assetstore.unity.com/packages/3d/environments/unityjapanoffice-152800?locale=ja-JP#description>, (参照 2021-12-16).
- [7] Unity: Unity, <https://unity.com>, (参照 2021-12-16).
- [8] Roger Sodr : Fulldome Camera for Unity 2019, <https://github.com/rsodre/FulldomeCameraForUnity>, (参照 2021-12-16).
- [9] Unity: Unity Recorder, <https://docs.unity3d.com/ja/Packages/com.unity.recorder@2.6/manual/index.html>, (参照 2021-12-16).

- [10] Tsung-Yi Lin, Genevieve Patterson, Matteo R. Ronchi, Yin Cui, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, Larry Zitnick and Piotr Doll r: COCO(Common Objects in Context), <https://cocodataset.org/#format-data>, (参照 2021-12-16).
- [11] 秋田悠河, 阿倍博信, 古宮嗣朗: 3D ゲームエンジンを用いた映像監視向け合成全周魚眼画像データセットの作成と評価, 第 20 回情報科学技術フォーラム (2021).
- [12] Kaiming He, Georgia Gkioxari, Piotr Doll r and Ross Girshick: MASK R-CNN, <https://arxiv.org/abs/1703.06870> (2018).
- [13] Tsung-Yi Lin, Genevieve Patterson, Matteo R. Ronchi, Yin Cui, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, Larry Zitnick and Piotr Doll r: COCO(Common Objects in Context), <https://cocodataset.org/#download>, (参照 2021-12-16).
- [14] Matterport: MASK R-CNN 2.1, https://github.com/matterport/Mask_RCNN/releases, (参照 2021-12-16).