

# 軽量可搬型 3次元空間センシングデバイスの設計開発

大河内 悠磨<sup>1</sup> Hamada Rizk<sup>1</sup> 山口 弘純<sup>1</sup>

**概要:** IoT・センシング技術の進化により, 人々の周辺環境のデータを取得し活用するアプリケーションが数多く開発されている. 特に3次元距離センサから得られる3次元点群は被写体の形状を立体的に取得できる利点があり, プライバシー侵害リスクも低い. 一方で, 既存の距離センサの多くは高性能コンピューティングを前提とした据置型デバイスであるため, 使用形態に制約が多い. 本稿では, 我々の研究グループで開発している, 小型の3次元距離センサを搭載した軽量可搬型の3次元空間センシングデバイスの設計開発について述べるとともに, 同デバイス上でリアルタイム実行可能な3次元点群の物体認識手法を提案する. 同手法では, センサから得られる距離情報から周辺環境の3次元点群を取得し, それをグリッド分割した点群に対する特徴量抽出を行う. 次に特徴空間上で事前にパラメータ推定したGMMに対し, Fisher Vectorに基づく特徴量表現を行い, 被写体の形状を識別するためのシグネチャを得る. 得られたグリッド毎の特徴量に対し, サポートベクターマシンでクラス分類を行い, 点群に対するクラスラベルを付与する. 開発したデバイスを用いたデータセットによる評価の結果, 提案手法は人や椅子に対し91.8以上のAPを達成した. また, 同デバイス上での処理レートは28.2フレーム/秒であり, エッジデバイスでも十分高速に物体検知処理を実行可能であることがわかった.

## Design and Development of Spatial Sensing Device

YUMA OKOCHI<sup>1</sup> HAMADA RIZK<sup>1</sup> HIROZUMI YAMAGUCHI<sup>1</sup>

### 1. はじめに

近年, IoTデバイスの発展や普及, ならびにセンシング技術の進歩により, 人間や屋内環境のデータを利用した多くの新しいアプリケーションが開発されてきている. これらのアプリケーションは安心して豊かな生活環境を構築し, 個人の生活の質を向上させる次世代の人間中心の知的環境の実現へとつながる. 特に人間のセンシングにおいては, COVID-19をはじめとする感染症の拡大防止に向けた人との距離や接触を把握するアプリケーション [1] や, オフィスや商業施設内の人流把握 [2], 高齢者見守り支援のための家庭内の転倒検知 [3] などのニーズが高まっている.

これらの検知システムの多くは, RGBカメラによる画像解析技術を利用することが多いが [4], 被撮影者の顔などの個人情報や, 外見や服装などのプライバシー機敏な情報が取得されるため, 特にプライベート空間での利用は受容

されない. 商業施設などの空間においても, 同意を得ないデータ取得においては利用目的が限定されるとともに, 訪問者への撮影通知や利用目的の周知, 場合によってはアウトの仕組みを導入する必要があるなど, 導入への障壁は低くない.

これに対し, 我々の研究グループでは3次元測域センサ (LiDAR) によって取得される3次元点群を用いた人流検知システム「ひとなび」を開発している. LiDARは多くの場合赤外線を用いており, 測域内の各方位に対し, 最も近い物体までの距離を取得することで, 周辺物体の存在を3次元空間に存在する点の集合で表現する. LiDARで取得した3次元点群は色情報を持たず, 被撮影者のプライバシー侵害リスクは低い. 大型で高精度のLiDARは広範囲の被写体の形状を精細に取得できるため, 屋内・屋外環境の物体検出 [5] や人物トラッキング [6] などに利用されている. しかし, これらのLiDARはデータ量の大きい高密度の点群を取得するため, 高性能な計算資源を用いて処理されることが多い [7].

これに対し, 我々の研究グループでは軽量可搬型3次元

<sup>1</sup> 大阪大学大学院情報科学研究科  
Graduate School of Information Science and Technology of  
Osaka University



図 1 hitonabiki- $\mu$ の利用シーン例

空間センシングデバイス「hitonabiki- $\mu$ 」の開発も同時に進めている。hitonabiki- $\mu$ は超小型かつ省電力のLiDAR型デバイスとマイコンボードを搭載しており、周辺環境の3次元点群を取得し、周辺環境の物体検知を行う。このデバイスを歩行者が装着することで、例えば図1のように装着者を中心とした空間のシーン理解が可能になる。また机上で学習・作業する人間の前にて、作業姿勢を検知するアプローチも開発している [8]。

本稿では、hitonabiki- $\mu$ のアーキテクチャの設計開発について述べるとともに、hitonabiki- $\mu$ のようないわゆるエッジデバイス上でリアルタイム実行可能な3次元点群による物体認識手法を提案する。3次元点群からの物体認識は、3次元点群が捉えた空間において、空間内の各物体を構成する部分点群(点群クラスター)を特定し、その点群クラスターに対応する物体の種別を特定するタスクである。提案手法では、取得した点群をPillarと呼ばれる柱状のグリッドに分割する。ノイズ除去を行った後、Fisher Vectorによる点群分布の特徴表現を得る。得られたFisher Vectorをサポートベクターマシンの入力とすることでオブジェクト種別(クラス)を推定する。

開発したデバイスを用いたデータセットによる評価の結果、提案手法は人や椅子の小さいオブジェクトに対しては91.8以上のAPを達成し、点群分布の特徴によって十分高精度なクラスの推定が可能であることが示された。柱状グリッドの分割により、オブジェクト間の距離が近い場合も互いを正しく区別できることも分かった。また、同デバイス上での処理レートは28.2フレーム/秒であり、エッジデバイス上で十分リアルタイム実行可能であることがわかった。提案手法の各モジュールの処理時間を比較したところ、柱状グリッド分割にかかる時間はよく知られたクラスターリング手法であるDBSCANと比較しても9分の1以下であり、これが全体の処理時間短縮に大きく寄与していることが分かった。

以降の章構成は以下の通りである。2章では、3次元点群セグメンテーション及びモバイルデバイスを利用した点群処理に関連する研究について述べる。3章では、本研究

で利用するデバイス「hitonabiki- $\mu$ 」の仕様について述べる。4章では、提案するセグメンテーション手法について説明し、5章でその手法のデータセットによる精度及び処理時間の評価を行い、またウェアラブルデバイスとしての利用における課題についても考察する。

## 2. 関連研究

### 2.1 3次元点群のセグメンテーションとクラス分類

3次元点群の物体認識は一般に、各点群がどのオブジェクトに属するかを決定する分類問題として定義され、これまでに数多く提案されている。良く知られているアプローチであるPointNet[9]では、3次元点群を入力とした多層パーセプトロンにより点群の特徴量を抽出し、対称関数であるMax-poolingを適用することで、点群の順不変性を保持したまま屋内環境の点群のセグメンテーションを行う手法を提案している。PointNet++[10]はPointNetの改良版であり、各点の近傍点群の情報を取り込むことにより精度を向上させている。VoteNet[11]は測域センサにより取得される点群は物体の表面のみを捉える性質に着目し、スキャンが困難な物体の中心点を投票メカニズムにより決定することで高精度での物体検出を可能にしている。

Chenら[12]は、自動運転車が道路上の移動する車と駐車車両を区別するため、車載LiDARから取得した時系列点群データの利用により、シーンを移動オブジェクトと静止オブジェクトに区別する手法を提案している。同手法では連続するフレーム間の情報から差分を取得し、現在のスキャンと共に深層ニューラルネットワークの入力とすることで識別性能を向上させている。

これらの手法は、高精度な物体識別を可能としている一方で、計算コストが高い深層学習を利用しているため、高性能なGPUの利用は必須である。したがって、GPUを持たない非力かつ省電力なエッジデバイスにこれらの手法を適用することは困難である。これに対し本研究では、軽量かつ周辺環境認識に十分な精度を達成可能なアプローチを探究する。

### 2.2 モバイルデバイスを利用した点群処理

3次元点群を、モバイルデバイスを利用して処理する手法もいくつか提案されている。Kimら[13]はリアルタイム3次元点群処理のためにグラフ畳み込みネットワークを採用し、高速化のための独自プロセッサを開発することでリアルタイムかつ低消費電力な処理を実現している。同手法は高精度を達成する一方、処理そのものの大きな軽量化には至っておらず、非力な汎用エッジデバイス上での実行効率の観点では課題が残る。

Liuら[14]は、LiDARを装着した視覚障害者向けの移動支援システムを提案している。LiDARを胴に、ノートPCを背に装備し、SLAMによる前方の3次元環境地図の作成

や物体認識、周辺環境のキャプショニング及び音声通知の処理を可能としている。同システムは歩行者向けの LiDAR 利用を実現した点で新しいが、大型の LiDAR や GPU を搭載したノート PC や RGB データを利用するなど、装備や処理の軽量化は実現されていない。

Shao ら [15] は、点群が示す物体のクラス分類アーキテクチャの前段部分を LiDAR を搭載したエッジデバイスが担い、後段部分をエッジサーバが担うことにより、通信データ量を抑えつつエッジサーバの計算処理を削減することに成功している。ネットワークを用いて処理性能の高いサーバと通信するというアイデアは Wi-Fi 環境が整備された学校やオフィス環境では有用であるが、Wi-Fi が整備されていない環境での利用は、移動体通信が可能なモジュールを搭載したマイコンを必要とし、かつ通信コストも必要となるため、コストや運用環境の問題が生じる。

これらの先行研究を踏まえ、本研究では市販マイコンボードと小型 LiDAR を組み合わせ、点群の取得・処理をオンボードで行うデバイスを開発し点群処理手法を提案する。

### 3. 3次元空間センシングデバイスの概要

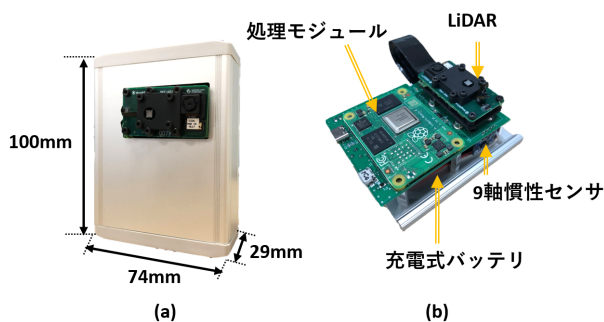


図 2 デバイスの寸法及びコンポーネント

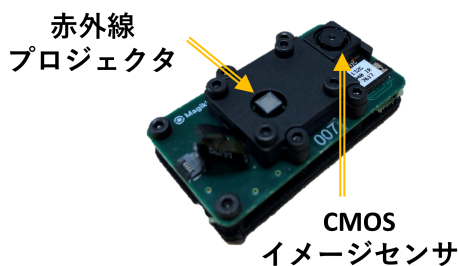


図 3 搭載する LiDAR の仕様・寸法

利用するデバイス「ひとたび- $\mu$ 」について、外箱の仕様及び搭載するマイコンボード・各種センサの仕様を示す。デバイスの外観は図 2 の通りである。外箱サイズは縦  $100\text{mm}$  × 横  $74\text{mm}$  × 高さ  $29\text{mm}$  であり、アルミニウム素材を利用している。前面には LiDAR センサ装着のための穴がけられており、底面にはバッテリー充電用の Type-C

ポート及び電源スイッチを有する。デバイスに搭載するマイコンボードは、機器組み込み向けの処理モジュールである Raspberry Pi Compute Module 4 (CM4) と、各種センサ接続及び電源供給のみを提供する I/O ボードを組み合わせて用いる。CM4 の CPU は ARM Cortex-A72 (1.50GHz, 4 コア) で、4GB の RAM を備えている。消費電力は OS 及び処理タスクにより大きく変化するが、アイドル状態では  $2\text{(W)}$ 、処理中は  $7\text{(W)}$  である [16]。

搭載する LiDAR は市販されており、図 3 に示すように、縦  $24\text{mm}$  × 横  $44\text{mm}$  × 高さ  $8\text{mm}$  の超小型のものを採用している。取得する点群は 3 次元空間における位置情報のみを保持し、色情報は持たない。

またひとたび- $\mu$  は 9 軸慣性センサ LSM9DS1 を搭載しており、動作周波数は  $14.9\text{Hz}$  から  $952\text{Hz}$  の範囲で取得できる。加速度、角速度、磁気強度の取得により、装着者及びデバイスの位置姿勢推定が可能である。

また、可搬型デバイスとしての利用のため、電源供給を行うバッテリーを搭載している。

### 4. システムの概要

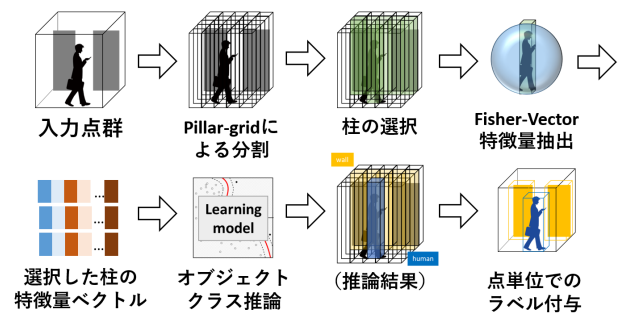


図 4 システムの処理フロー

3章で述べたデバイスに搭載する 3次元点群の処理フローを図 4 に示す。提案手法では、LiDAR から得られる 3次元点群を入力とし、各データフレームについてまず Pillar-grid による点群分割処理を施し、オブジェクトが存在する柱を選択する (4.1 節)。選択された各柱に対し Fisher Vector ベースの特微量を抽出し、柱に存在するオブジェクトの形状を把握する (4.2 節)。得られた特徴量ベクトルを機械学習モデルの入力とし、オブジェクトのクラス推論を学習させ、推論結果に応じて各点にクラスラベルを付与する (4.3 節および 4.4 節)。以降の節で各処理の詳細な説明を行う。

#### 4.1 柱状グリッドによる点群分割とグリッド選択

オブジェクト検出手法である PointPillars[17] では柱状グリッドを用いて点群を柱状に分割し処理の高速化を図っている。本手法でも同様の方法を採用する。

LiDAR から取得した 3次元点群を  $\{L_i | i = 1, \dots, n\}$  とおく。各点  $L_i$  は 3次元空間における位置情報  $(x_i, y_i, z_i)$

を保持する．この点群を地面と平行な  $x$ - $z$  平面上の等間隔な縦  $m$  マス，横  $n$  マスのグリッドに分割し，柱の集合  $P(|P| = m \times n)$  を得る．柱の高さ，すなわち  $y$  軸方向には制限を設けないものとする．

次に，得られた柱の集合  $P$  からオブジェクトが存在すると思われる柱を選択する．存在の判定には単純に各柱に含まれる点数を用いる．点数が閾値以上であればその柱を選択し，そうでなければその柱に含まれる点はノイズ由来のものとして判定し以降の特徴量抽出の対象としない．これにより選択された柱の集合を  $P'$  とする．これらの処理の時間計算量は  $O(n)$  であり，オブジェクト検出でしばしば用いられるクラスタリング手法である DBSCAN[18] (最悪計算量  $O(n^2)$ ) などと比較し，高速なクラスタリング処理を実現できる．なお，柱状グリッドサイズが小さすぎると形状を捉えることができず，取得される特徴量ベクトル数が増え処理時間が増加する．一方で大きすぎると1つの柱状グリッドに2種類以上のオブジェクトを示す点群が含まれる可能性が増加し，認識精度が低下する恐れがある．我々の実装では，これらを勘案し，柱状グリッドの底面サイズを  $20\text{cm} \times 20\text{cm}$  としている．

#### 4.2 Fisher Vector 表現による特徴量抽出

次に，4.1 節で選択された各柱から Fisher Vector (FV) 表現に基づく特徴量抽出を行う．一般に点群データは，データ領域が一定でありかつデータの近接関係も明確である画像データとは異なり，非順序的・非構造的で点数などのサイズも異なるため，特徴抽出が容易でないという課題がある．FV 表現は入力サンプルサイズ (点数) に依存しない特徴表現が可能であり，点群データの特徴を表現するのに適すると考えられる．

本手法では，各柱状グリッド内における点群の分布から得られる特徴量ベクトルを用いる．これに対し，様々なシーンの点群から得られる特徴量ベクトルが構成する特徴空間において，それらの分布からパラメータ学習を行った混合ガウスモデル (GMM) を用い，その GMM からの3次元点群の偏差として定義される FV を用いてコンパクトに特徴量を表現する．具体的には FV は GMM のパラメータ (各ガウス分布の重み，平均，共分散行列) に対し，サンプルの対数尤度の勾配を用いることで得られる．FV は固定長ベクトルで表現されるため，分類器への入力として有用である．

提案手法での FV の導出を説明する．選択した柱状グリッド  $i$  に存在するサイズ  $T$  の点群を  $X_i = \{\mathbf{p}_t \in R^3, t = 1, \dots, T\}$ ，GMM のパラメータ  $\lambda$  を  $\lambda = (\mu, \Sigma)$  とおく． $\mu$  はガウシアン期待値， $\Sigma$  は共分散行列を表す． $\mu$  の値は選択した柱に応じて変化し， $x, z$  の値は選択した柱の  $x, z$  の最大値・最小値の平均を設定し， $y$  の値は 1000 (実世界における 1m の長さに相当する) で固定している． $\Sigma$  は計

算簡略化のため単位行列とする．提案手法では GMM を構成するガウシアンは1つのみであるとする．ある点  $\mathbf{p}$  がこのガウシアンに属する尤度は，

$$u(\mathbf{p}) = \frac{1}{(2\pi)^{3/2} |\Sigma|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{p}-\mu)^T \Sigma^{-1}(\mathbf{p}-\mu)\right\} \quad (1)$$

と表される．この時，Fisher Vector は，正規化された勾配の合計である． $L$  を Fisher 情報行列の逆行列とすると，Fisher Vector  $\mathcal{G}$  は，

$$\mathcal{G} = \sum_{t=1}^T L \nabla \log u(\mathbf{p}_t) \quad (2)$$

と表せる．正規化された勾配について，勾配を求める変数別に書き分けると，

$$\mathcal{G}_\alpha = \sum_{t=1}^T (u(\mathbf{p}_t) - 1) \quad (3)$$

$$\mathcal{G}_\mu = \sum_{t=1}^T u(\mathbf{p}_t) \left( \frac{\mathbf{p}_t - \mu}{\sigma} \right) \quad (4)$$

$$\mathcal{G}_\sigma = \sum_{t=1}^T u(\mathbf{p}_t) \left( \frac{(\mathbf{p}_t - \mu)^2}{\sigma^2} - 1 \right) \quad (5)$$

と表すことができる．4.3 節で説明する機械学習モデルの入力として，これらの値のほか，式 (3)-(5) の合計前の各値の最大値，および式 (4)-(5) の最小値を追加する．また，各柱状グリッドに含まれる点群の高さ及び点の個数も追加する．すなわち

$$\mathcal{X}_{input} = \begin{bmatrix} \sum_{t=1}^T L \nabla \log u(\mathbf{p}_t) |_{\lambda=\alpha, \mu, \sigma} \\ \max(L \nabla \log u(\mathbf{p}_t)) |_{\lambda=\alpha, \mu, \sigma} \\ \min(L \nabla \log u(\mathbf{p}_t)) |_{\lambda=\mu, \sigma} \\ \max(y | (x_t, y_t, z_t) \in X_i) \\ T \end{bmatrix} \quad (6)$$

となる．次元数は， $\mu$  および  $\sigma$  が  $x, y, z$  の3次元についてそれぞれ計算されるため，式 (6) の上から順に 7, 7, 6, 1, 1 であり，計 22 次元となる．この特徴量ベクトルを選択された各柱について求める．その他のパラメータは文献 [19][20] のアプローチに基づいている．

#### 4.3 機械学習モデルによる特徴量学習・クラス推論

4.2 節で得た FV を機械学習モデルの入力とし，教師あり学習モデルであるサポートベクターマシン (SVM) による分類を行う．SVM は分類・回帰タスクのためのカーネルベースの機械学習モデルであり，特徴空間において既存の判定境界間のマージンが最大になるように判定境界の閾数を決定する．SVM はパターン認識のための強力なツ

表 1 作成したデータセットの詳細 ※: 複数の角度 (30°, 60°, 90°) から計測している (LiDAR の検知方向と壁が正対している状態を 0 度とする) ☆: 複数の対象オブジェクトとの距離 (60cm, 120cm, 180cm) から計測している

シナリオ	フレーム数	存在オブジェクト (M = 動く, S = 静止)								
		human			wall			chair		
総フレーム数	12105	M	S	(note)	M	S	(note)	M	S	(note)
01 壁のみ	2272					1	※			
02 立っている人のみ	1893		1	☆						
03 壁の前を通過する人	397	1		通過		1				
04 通過する人	436	1		通過						
05 壁の前から接近してくる人	352	1		接近		1				
06 接近してくる人	362	1		接近						
07 壁の方向に離れていく人	370	1		離反		1				
08 離れていく人	234	1		離反						
09 椅子 (前面)	2392							1		前面☆
10 椅子 (側面)	1321							1		側面☆
11 壁と椅子	973					1			1	
12 人と壁と椅子	572		1			1			1	
13 壁と立っている人	531	1				1	45°			

ルとして利用されており, 他の教師あり学習法よりも優れていることが示されている [21]. SVM の出力は柱状グリッドに存在する物体のクラス (人や壁, 家具など) である. モデルの訓練は事前に用意したデータセットを用いて行う. データセットは点群の各点に対し, それがどの物体クラス (例えば人や壁) に属するかをアノテーションしたものである. 訓練データにおける各点群フレーム (同時刻に得られた点群の集合) に対し, 以下の手順に従い入力ベクトルと正解ラベルの組を用意する.

- (1) 4.1 節に基づき柱を選択する.
- (2) 4.2 節に基づき FV を得る. 学習モデルの入力ベクトルとする.
- (3) 選択した柱に存在する点につけられたクラスラベルのうち, 最も多いものをその柱のクラスラベルとし, それを正解値とする.

訓練済みモデルによるクラス推論は, 図 4 に示す手順で行われる. クラス推論後は, 選択された柱すべてにクラスラベルが付与される.

#### 4.4 点単位のクラスラベル付与

柱に付与されたクラスラベルを, 柱内に存在する全点に対し付与することで点毎のクラス分類を完了する. なお, 後述する評価のため, 3次元バウンディングボックスを生成しておく. 生成方法は以下の通りである. 各柱について隣接した柱 (斜め方向を除く) のクラスラベルが同一であれば, 含まれる全ての点群を同一のバウンディングボックスに含める. これをすべての柱に対して行う.

## 5. 評価

### 5.1 データセット

4章で述べたシステムの性能評価を行うために, データセットを作成した. ひとなび- $\mu$  デバイスは三脚によって 1m の高さに固定され, 15m  $\times$  25m の十分に広い部屋で 13 種類のシナリオの点群を取得した. 平均フレームレートは 10 (フレーム/秒) であった. 撮影された点群は 12,105 フレームであり, すべてのフレームに存在するオブジェクトに対し 3次元バウンディングボックスを設定した. このアノテーション処理は提案手法の学習およびテストに不可欠である. 設定したオブジェクトのクラスラベルは「human」「wall」「chair」のいずれかであり, どのバウンディングボックスにも属していない点はノイズとみなし, クラスラベルが付与されない. データセットの詳細は表 1 に示している.

### 5.2 比較手法

本稿では, 4.1 節で説明した柱状グリッドの処理をクラスタリングアルゴリズム DBSCAN に置き換え, 各クラスタから FV 表現に基づく特徴量抽出を行い, それ以降の処理は提案手法と同じである手法 (DBSCAN-FV) を実装した. 両方の機械学習モデルの訓練には, 各シナリオの前半 80% のフレームを, 性能評価には後半 20% のフレームを利用した.

評価はセグメンテーションの精度及び実行時間の観点から行った. 精度評価には Song ら [22] が提案した各クラスの平均適合率 (Average Precision, AP) 及び各クラスの平均 AP である mean Average Precision (mAP) を利用した. これらの指標は, 正解データと予測された 3次元バウ

表 2 セグメンテーション手法の精度評価結果

	human	wall	chair	mAP
DBSCAN-FV	95.7	96.1	100	97.3
Pillar-grid-FV (提案手法)	91.8	57.9	92.1	80.6

表 3 処理時間比較 (2400 フレーム)

	処理時間 (s)	フレーム/秒
DBSCAN-FV	216.6	11.1
Pillar-grid-FV (提案手法)	<b>85.0</b>	<b>28.2</b>

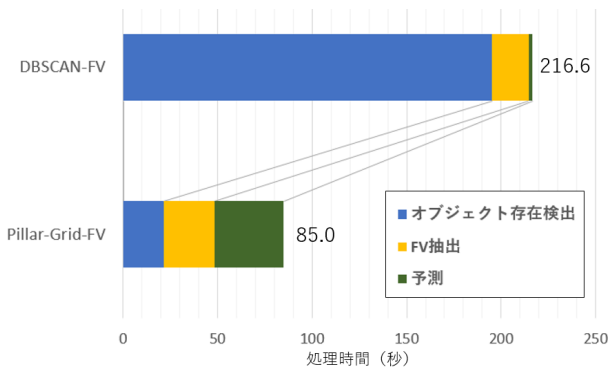


図 5 各モジュールの処理時間比較 (2400 フレーム)

ンディングボックスの重なり度合いである IoU (Intersect over Union) を利用して、検出の精度を評価している。本評価では、AP の計算における IoU の閾値を 0.25 とした。評価プロトコルは [23] の手法に従っているが、利用される 3 次元バウンディングボックスは  $y$  軸周りの回転は行わない。

実行時間は予測にかかる時間を指す。入力として与えた性能評価用の 2400 フレームの 3 次元点群データをすべて Python の変数として格納してから計測を開始し、すべての点群に対しセグメンテーションを終えた直後に計測を終了する。この処理を 20 回実行し実行時間の平均をとる。この処理は全て Raspberry Pi 4 Model B 上で実行される。このマイコンはひとたび- $\mu$  で用いる処理モジュールの性能と等しい。

### 5.3 結果

精度評価を表 2, 実行時間を表 3 及び図 5 に示す。セグメンテーション精度に関しては、提案手法である Pillar-grid-FV は比較対象手法の DBSCAN-FV と比較して、人 (human) や椅子 (chair) の検出はほぼ同等の精度を出しているが、壁 (wall) の AP は 38.2 落ちており、mAP の値は 16.7 低いことが分かる。

実行時間に関しては提案手法が優れていることが分かる。全体の処理時間が 60.8% 削減できており、28.2 (フレーム/秒) で処理可能であった。これは Pillar-grid による離散化及び柱の選択にかかる時間は DBSCAN のクラスタリングの 9 分の 1 以下に抑えられており、これが処理時間の短縮に大きく貢献しているといえる。一方でその後の処理にか

かる時間は提案手法の方が 3 倍長いことも分かる。

### 5.4 実験結果解析

5.3 章で得られた結果をもとに、その原因について述べる。セグメンテーションの正解データ及び各手法によるセグメンテーションの結果を図 6 に示す。(a) に示すフレームでは、どの点群もおおむね正しくセグメンテーションされていることが分かる。(b) は壁の前に人がいるフレームだが、提案手法が二つのオブジェクトを正しく検出している一方で、DBSCAN-FV は一つのオブジェクトと見做し壁と人を分離できていないことがわかる。これはクラスタリングアルゴリズムの処理が原因である。DBSCAN は密度ベースのクラスタリングアルゴリズムであり、人の点群と壁の点群の距離が近い場合、それらは一つのクラスであると判定される。その一方で提案手法は先に Pillar-grid による離散化を施しているため、人と壁がグリッドにより適切に分割される限りは、オブジェクト間の距離の影響は無いことが分かる。

(c) に示すフレームでは提案手法のバウンディングボックスが複数に分割されている。これは、一部の柱のクラス分類が誤っていることに起因する。提案手法ではバウンディングボックスの構成方法について「隣り合う柱のクラスラベルが同じであれば、同一のバウンディングボックスに含める」という処理を行っている。そのため、(b) 壁であると予測されたことを示す黄色の点群の間に、人だと予測されたことを示す青色で示された点群がある場合は、壁の点群が 2 分割されてしまうという問題が発生する。この問題が多量のフレームで発生しているため、評価指標における壁の AP が低下している。バウンディングボックスの構成方法や、隣り合う柱の情報を、機械学習モデルによるクラス推論に利用することでこの精度は改善されると考える。

実行時間は提案手法のモジュールである Pillar-grid の離散化がクラスタリングよりも高速であった。これは 4.1 節で述べた通り時間計算量のオーダーが  $O(n)$  であるため、全体処理時間の短縮に大きく貢献したと言える。一方で、特徴量抽出およびクラス推論にかかる時間は提案手法の方が長くなっている。これは DBSCAN-FV がクラスタリング処理を行うためオブジェクトの個数だけ特徴量ベクトルが生成されるのに対し、Pillar-grid-FV における特徴量ベクトルは離散化により選択された柱の数生成されることに起因する。離散化による柱の数は通常オブジェクト数よりも多いため、結果としてその後の処理に時間がかかると考えられる。

### 5.5 ウェアラブルデバイスとしての利用

開発中のひとたび- $\mu$  はウェアラブルデバイスとしての利用シーンを想定しているため、実際にデバイスを装着し歩行していても周辺環境の点群を正しくセグメンテーション

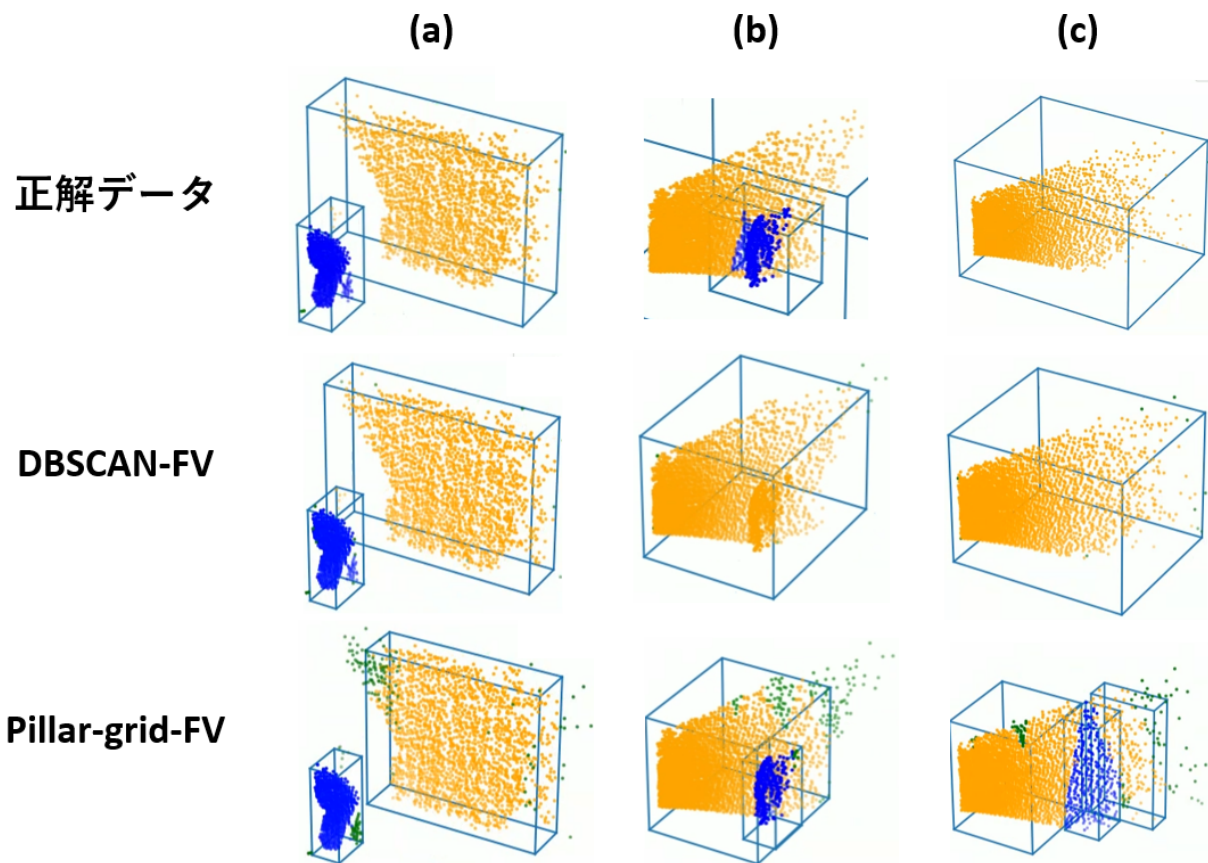


図 6 正解データ及び各手法のセグメンテーション結果  
黄色は壁、青色は人、緑色はノイズの点を示す

出来るかを検証する。歩行中デバイスは不安定な状態であり LiDAR の方向はいつも地面に平行であるとは限らないため、データセットとは異なる状態の点群が取得される。

試験データの取得は以下の手続きにより行われる。デバイスの装着者 1 人と被撮影者 1 人が 2.5m の間隔を空けて立ち、装着者はデバイスを首に掛け、立って静止した状態でセンシングを開始する。装着者は被撮影者の方向に歩行を開始し、被撮影者の側方を通過したのち計測を終了する。

イスが上方に 20° 傾いている際に取得された点群データである。この点群をそのまま提案手法の入力としても正しくセグメンテーションができないことが分かる。一方で (b) に示す点群は、デバイスの傾きに応じて、点群全体に回転処理を加え、点群の  $x-z$  平面が実世界の地面と平行になるようにしたものである。これを提案手法の入力とすることで、セグメンテーションの精度が (a) よりも向上していることが分かる。この回転処理をデバイスに搭載された慣性センサの計測値をもとに、リアルタイムに行うことでウェアラブルデバイスとしての利用が可能になると考える。

## 6. おわりに

本稿では軽量可搬型 3 次元空間センシングデバイス「ひとたび-μ」の概要、及びそのようなエッジデバイス上でリアルタイム動作可能な 3 次元点群のセグメンテーション手法を提案した。提案手法は計算能力が限られたエッジデバイス上での実行に対応するため、処理時間コストの小さい Pillar-grid による離散化及び Fisher Vector に基づく特徴量抽出を採用した。データセットにより精度及び実行時間の観点で評価を行った結果、精度については人や椅子など小型のオブジェクトに対しては 91.8 以上の AP を達成し、エッジデバイス上での処理速度は 28.2 (フレーム/秒) であり、十分にリアルタイム実行可能であることが示

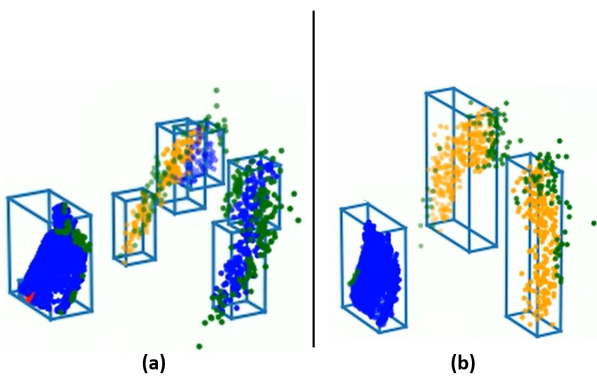


図 7 歩行時に取得した点群データのセグメンテーション結果

取得したデータを提案手法により処理し、得られたセグメンテーション結果を図 7 に示す。(a) に示す点群はデバ

された。将来的には壁のような大型オブジェクトのセグメンテーションについても精度向上を図り、ウェアラブルデバイスとしての利用を想定し、慣性センサの計測値を利用した処理フロー全体の向上に取り組む予定である。

## 参考文献

- [1] 厚生労働省: 新型コロナウイルス接触確認アプリ (COCOA), , 入手先 ([https://www.mhlw.go.jp/stf/seisakunitsuite/bunya/cocoa\\_00138.html](https://www.mhlw.go.jp/stf/seisakunitsuite/bunya/cocoa_00138.html)) (参照 2022-05-19).
- [2] Voigtlaender, P., Krause, M., Osep, A., Luiten, J., Sekar, B., Geiger, A. and Leibe, B.: Mots: Multi-object tracking and segmentation, *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7942–7951 (2019).
- [3] Shu, F. and Jeff, J.: An eight-camera fall detection system using human fall pattern recognition via machine learning by a low-cost android box, *Scientific Reports* (2021).
- [4] Zhao, Z.-Q., Zheng, P., Xu, S.-T. and Wu, X.: Object Detection With Deep Learning: A Review, *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 30, No. 11, pp. 3212–3232 (online), DOI: 10.1109/TNNLS.2018.2876865 (2019).
- [5] Zhu, X., Zhou, H., Wang, T., Hong, F., Ma, Y., Li, W., Li, H. and Lin, D.: Cylindrical and Asymmetrical 3D Convolution Networks for LiDAR Segmentation, *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9939–9948 (2021).
- [6] Zhao, J., Xu, H., Liu, H., Wu, J., Zheng, Y. and Wu, D.: Detection and tracking of pedestrians and vehicles using roadside LiDAR sensors, *Transportation Research Part C: Emerging Technologies*, Vol. 100, pp. 68–87 (online), DOI: <https://doi.org/10.1016/j.trc.2019.01.007> (2019).
- [7] Guo, Y., Wang, H., Hu, Q., Liu, H., Liu, L. and Benamoun, M.: Deep Learning for 3D Point Clouds: A Survey, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 43, No. 12, pp. 4338–4364 (online), DOI: 10.1109/TPAMI.2020.3005434 (2021).
- [8] Katayama, H., Mizomoto, T., Rizk, H. and Yamaguchi, H.: You Work We Care: Sitting Posture Assessment Based on Point Cloud Data, *2022 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*, pp. 121–123 (online), DOI: 10.1109/PerComWorkshops53856.2022.9767292 (2022).
- [9] Qi, C. R., Su, H., Mo, K. and Guibas, L. J.: PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation, *arXiv preprint arXiv:1612.00593* (2016).
- [10] Qi, C. R., Yi, L., Su, H. and Guibas, L. J.: PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space, *arXiv preprint arXiv:1706.02413* (2017).
- [11] Qi, C. R., Litany, O., He, K. and Guibas, L. J.: Deep Hough Voting for 3D Object Detection in Point Clouds, *Proceedings of the IEEE International Conference on Computer Vision* (2019).
- [12] Chen, X., Li, S., Mersch, B., Wiesmann, L., Gall, J., Behley, J. and Stachniss, C.: Moving Object Segmentation in 3D LiDAR Data: A Learning-Based Approach Exploiting Sequential Data, *IEEE Robotics and Automation Letters*, Vol. 6, No. 4, pp. 6529–6536 (online), DOI: 10.1109/LRA.2021.3093567 (2021).
- [13] Kim, S., Kim, S., Lee, J. and Yoo, H.-J.: A Low-Power Graph Convolutional Network Processor With Sparse Grouping for 3D Point Cloud Semantic Segmentation in Mobile Devices, *IEEE Transactions on Circuits and Systems I: Regular Papers*, Vol. 69, No. 4, pp. 1507–1518 (online), DOI: 10.1109/TCSI.2021.3137259 (2022).
- [14] Liu, H., Liu, R., Yang, K., Zhang, J., Peng, K. and Stiefelhagen, R.: HIDA: Towards Holistic Indoor Understanding for the Visually Impaired via Semantic Instance Segmentation With a Wearable Solid-State LiDAR Sensor, *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pp. 1780–1790 (2021).
- [15] Shao, J., Zhang, H., Mao, Y. and Zhang, J.: Branchy-GNN: A Device-Edge Co-Inference Framework for Efficient Point Cloud Processing, *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8488–8492 (online), DOI: 10.1109/ICASSP39728.2021.9414831 (2021).
- [16] Ltd., P. P.: Raspberry Pi Compute Module 4 A Raspberry Pi for deeply embedded applications, , available from (<https://datasheets.raspberrypi.com/cm4/cm4-datasheet.pdf>) (accessed 2022-05-20).
- [17] Lang, A. H., Vora, S., Caesar, H., Zhou, L., Yang, J. and Beijbom, O.: PointPillars: Fast Encoders for Object Detection From Point Clouds, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019).
- [18] Li, S.-S.: An Improved DBSCAN Algorithm Based on the Neighbor Similarity and Fast Nearest Neighbor Query, *IEEE Access*, Vol. 8, pp. 47468–47476 (online), DOI: 10.1109/ACCESS.2020.2972034 (2020).
- [19] Ben-Shabat, Y., Lindenbaum, M. and Fischer, A.: 3DmFV: Three-Dimensional Point Cloud Classification in Real-Time Using Convolutional Neural Networks, *IEEE Robotics and Automation Letters*, Vol. 3, No. 4, pp. 3145–3152 (2018).
- [20] Sanchez, J., Perronnin, F., Mensink, T. and Verbeek, J.: Image Classification with the Fisher Vector: Theory and Practice, *International Journal of Computer Vision*, Vol. 105, No. 3, pp. 222–245 (online), DOI: 10.1007/s11263-013-0636-x (2013).
- [21] Cervantes, J., Garcia, F., Rodríguez, J. and Lopez, A.: A comprehensive survey on support vector machine classification: Applications, challenges and trends, *Neurocomputing*, Vol. 408, pp. 189–215 (online), DOI: <https://doi.org/10.1016/j.neucom.2019.10.118> (2020).
- [22] Song, S., Lichtenberg, S. P. and Xiao, J.: SUN RGB-D: A RGB-D scene understanding benchmark suite, *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 567–576 (online), DOI: 10.1109/CVPR.2015.7298655 (2015).
- [23] Song, S. and Xiao, J.: Deep Sliding Shapes for Amodal 3D Object Detection in RGB-D Images, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016).