CM-002

可搬型3次元空間センシングデバイスを用いた軽量なリアルタイム物体検出

大河内 悠磨[1]　Hamada Rizk[1]　山口 弘純[1]

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上で動作する，リアルタイム実行可能な 3 次元点群の物体検出手法を提案する．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，それらをグリッド分割したうえでクラスタリングを行い，物体のセグメンテーションを行うとともに，各セグメント内の点群分布を GMM と Fisher Vector に基づく特徴量でコンパクトに表現し，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，各点に対しそれが人，壁，椅子のいずれかをクラスラベルを付与する．同デバイスで取得したデータセットを用いた評価実験において 96.1 の mAP を達成し，高精度で物体検出できることを示した．また，同デバイス上での処理レートは 59.3 フレーム/秒であり，小型軽量の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もわかった．

Light-weight and Real-time Object Detection Using Portable 3-D Spatial Sensing Device

YUMA OKOCHI[1]　HAMADA RIZK[1]　HIROZUMI YAMAGUCHI[1]

1. はじめに

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，人や環境の存在理解のニーズが高まっている．それらの技術は，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，場合によってはオプト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．

我々のグループでは 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 [5]．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や人物トラッキングなどに利用されている [6]．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]．

一方で，我々は軽量可搬型 3 次元空間センシングデバイス「ひとなび-μ」の開発も進めている．ひとなび-μ は超

[1] 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　ひとなび-μの利用シーン例

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とを想定している．ひとなび-μの利用シーンを図 1 に示す．移動支援のユースケースでは，視覚障碍者や高齢者などが移動をする際にこのデバイスを装着することで，前方の障害物を検知し，装着者に通知することが可能になる．人流検知・行動把握のケースでは，デバイスをどこにでも設置できるという利便性を活かし，通過人数や通過した人の属性・行動把握を全てデバイス上で行うことができる．

我々はこのデバイスを利用し，机上で学習・作業する人間の作業姿勢検知手法 [8]，および人間の識別手法 [9] を開発している．これらの手法は高精度で検知・識別を行えるが，人がデバイスの正面に存在している状況を前提としており，人や壁，家具などを含むシーン全体が含まれた 3 次元点群の処理は考慮していない．したがって，それらの手法を実環境に適用したり，また前述のような利用シーンで用いるためには，デバイスが捉える周辺環境の物体切り出し（セグメンテーション）と分類を行い，人間に対応するの点群のみを切り出す手法の提案が必要である．

本稿では，ひとなび-μのような小型デバイス上でリアルタイム実行可能な，3 次元点群からの物体認識手法を提案する．本稿が扱う物体認識は，デバイスから得られる空間の 3 次元点群において，空間内の各物体を構成する部分点群（点群クラスタ）を特定し，その点群クラスタに対応する物体の種別を特定する作業である．提案手法では，取得した点群を Pillar-grid と呼ばれる柱状グリッドセルに分割し，それを用いた比較的簡易なクラスタリングで，点群をオブジェクトに相当するクラスタに分割する．得られた各クラスタから，GMM（混合ガウスモデル）と 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．なお，我々の先行手法 [10] では，クラスタの精度に課題を残していたが，本稿では切り出した 1 オブジェクト全体の特徴を 1 つの GMM と Fisher Vector で表現することで精度向上を図っている点が異なり，実際の精度も大きく異なる．

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 96.1 の mAP 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．柱状グリッドによる空間と点群の分割により，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 59.3 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，柱状グリッド分割およびグリッドベースのクラスタリングににかかる時間はよく知られたクラスタリング手法である 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「ひとなび-μ」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

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[11] 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．VoteNet[12] 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．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．

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 を装着した視覚障碍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

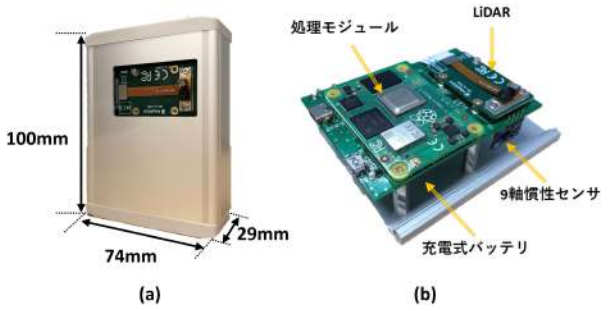

図 2 (a) ひとなび-μの寸法・(b) コンポーネント

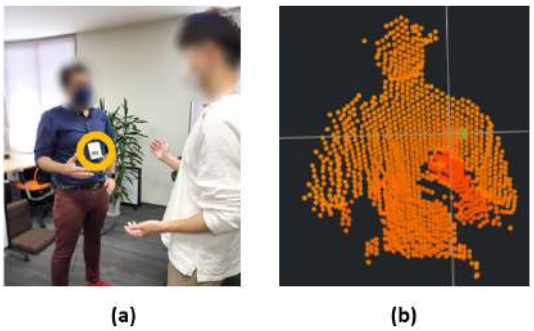

図 3 (a) デバイスの着用イメージ・(b) 取得点群

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．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. ひとなび-μ

利用するデバイス「ひとなび-μ」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 $100mm$ × 横 $74mm$ × 高さ $29mm$ 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あけられており，底面にはバッテリ充電用の 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(W)，処理中は 7(W) である [15].

搭載する LiDAR は市販されており，縦 $24mm$ × 横 $44mm$ × 高さ $8mm$ 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．またひとなび-μは 9 軸慣性センサ LSM9DS1 を搭載しており，動作周波数は 14.9Hz から 952Hz 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．さらに，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を搭載している．

ひとなび-μの着用イメージ及び取得した点群を図 3 に示す。(a) のように首掛け式での着用を想定しており、検出範囲内にいる人や物の形状を (b) のような 3 次元点群の形式で表現する。点群は姿勢や手の形を精細にとらえることができる。

4. ひとなび-μ を用いた物体認識

提案する 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，グリッドベースのクラスタリング処理を行う（4.1 節）．得られた各クラスタに対し Fisher Vector ベースの特徴量を抽出し，各クラスタの点群の分布を把握する（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，各クラスタのクラス推論を行い，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）．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・クラスタリング

オブジェクト検出手法である PointPillars[16] 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 $\{L_i|i = 1, ..., n\}$ 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 s マス，横 t マスの柱状グリッドに分割する．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，分割された点群 $G_{m,n}$ から，グリッドベースのクラスタリング処理を行う．処理の流れを図 5 に示す．クラスタリングの前処理としてノイズ除去を行う．各グリッドに含まれる点数を用いて，点数が閾値以上であればオブジェクトが存在するグリッド（図 5 における緑色グリッド）とみなす．閾値未満であればノイズのみのグリッド（図 5 における灰色グリッド）とみなし，以降のクラスタリング処理の対象に含めない．

そして，グリッドベースのクラスタリング処理を行う．上下左右に隣接するグリッドがどちらもオブジェクト存在グリッド（緑色グリッド）であれば，それらを一つのクラスタとみなす．この処理を繰り返すことによって，図 5 のようにオブジェクト単位のクラスタに分割できる．

これらの処理の時間計算量は，点群分割の時間計算量

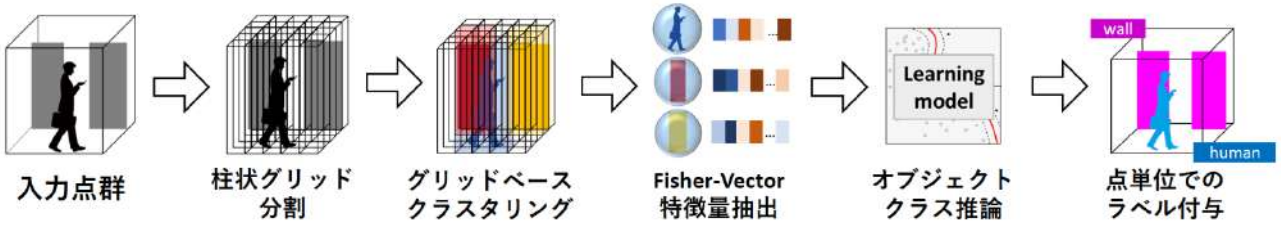

図 4　提案手法の処理フロー

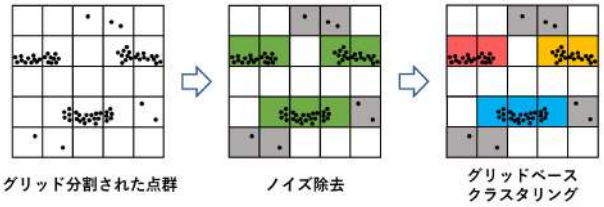

図 5　提案手法のクラスタリング（柱状グリッドを上から見た図）

$O(n)$ 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[17]（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，各グリッドに含まれる点群がノイズか否かの判定が困難になる．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 $20cm \times 20cm$ 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，各クラスタ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．クラスタ i のサイズ T の点群を $X_i = \{\boldsymbol{p}_t \in \mathbb{R}^3, t = 1, ..., T\}$，GMM のパラメータ λ を $\lambda = (\mu, \Sigma)$ とおく．μ はガウシアンの期待値，Σ は共分散行列を表す．μ 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．Σ 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 $\boldsymbol{p}$ がこのガウシアンに属する尤度は，

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

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 $\mathscr{G}$ は，

$$\mathscr{G} = \sum_{t=1}^{T} L\nabla logu(\boldsymbol{p_t}) \quad (2)$$

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，

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

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

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

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，各クラスタ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，

$$\mathscr{X}_{input} = \begin{bmatrix} \sum_{t=1}^{T} L\nabla logu(\boldsymbol{p_t})|_{\lambda=\alpha,\mu,\sigma} \\ max(L\nabla logu(\boldsymbol{p_t}))|_{\lambda=\alpha,\mu,\sigma} \\ min(L\nabla logu(\boldsymbol{p_t}))|_{\lambda=\mu,\sigma} \\ max(y|(x_t, y_t, z_t) \in X_i) \\ T \end{bmatrix} \quad (6)$$

となる，次元数は，μ および σ 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 [18][19] 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関数を決定する．SVM はパターン認識のための強力なツー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 [20]．SVM の出力は各クラスタが表す物体のクラス（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) DBSCAN によりクラスタを生成する．
(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) クラスタ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クラスタのクラスラベルとし，それを正解値とする．

ここで，訓練データの作成には提案手法のクラスタリングの代わりにより精度の高い DBSCAN を用いる．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．各クラスタのクラスを推論後，クラスタを構成する全点に対しクラスラベルを付与することで点毎のクラス分類を完了する．

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-μ 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．

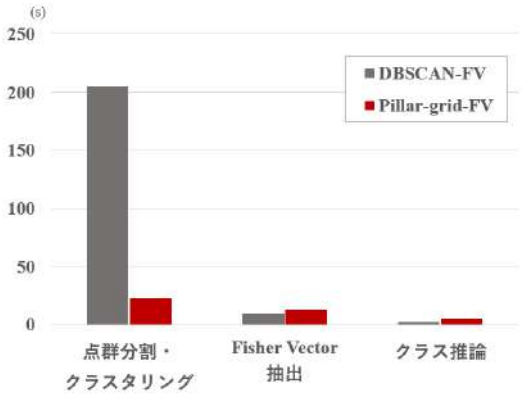

図 6 各モジュールの処理時間比較（2400 フレーム）

5.2 比較手法

本稿では，4.1 節で説明した柱状グリッド分割とそのクラスタリング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 [21] 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 [22] 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なび-μ 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 6 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，すべてのクラスについて同等の精度で検出していることが分

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°			

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	96.0	92.5	100	96.2
Pillar-grid-FV（提案手法）	97.3	91.0	100	96.1

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)	**40.5**	**59.3**

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 81.3% 削減できており，59.3（フレーム/秒）で処理可能であった．Pillar-grid による離散化及びグリッドベースのクラスタリングにかかる時間は DBSCAN 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 1.5 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 7 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知している一方で，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タであると判定される．その一方で提案手法は先に柱状グリッド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，クラスタリングにミスが発生していることに起因する．提案手法で用いたクラスタリングの手法は，「上下左右に隣接するグリッドにオブジェクトが存在すれば，それらを一つのクラスタに含める」である．そのため，壁がグリッドに対し斜め方向に伸びている場合，壁の点群が複数のクラスタに分割されてしまうという問題が発生する．この問題が多く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 が他と比べて低下している．我々は，過去フレームの情報を利用し，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柱状グリッドによる離散化とクラスタリングの組み合わせが，DBSCAN クラスタリングよりも高速であった．これは 4.1 節で述べた通り柱状グリッドによる離散化とクラスタリングの時間計算量が $O(n)$ 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．これは提案手法のクラスタリング処理に誤りに起因する．その誤りの多くは，DBSCAN が出力するクラスタの個数よりも多くのクラスタに分割してしまったことに起因する．しかし，特徴量抽出とクラス推論にかかる実行時間の差は 1 フレームあたり 2.5(ms) 程度であり，提案手法が満たすべきリアルタイム性にはほぼ影響がないといえる．

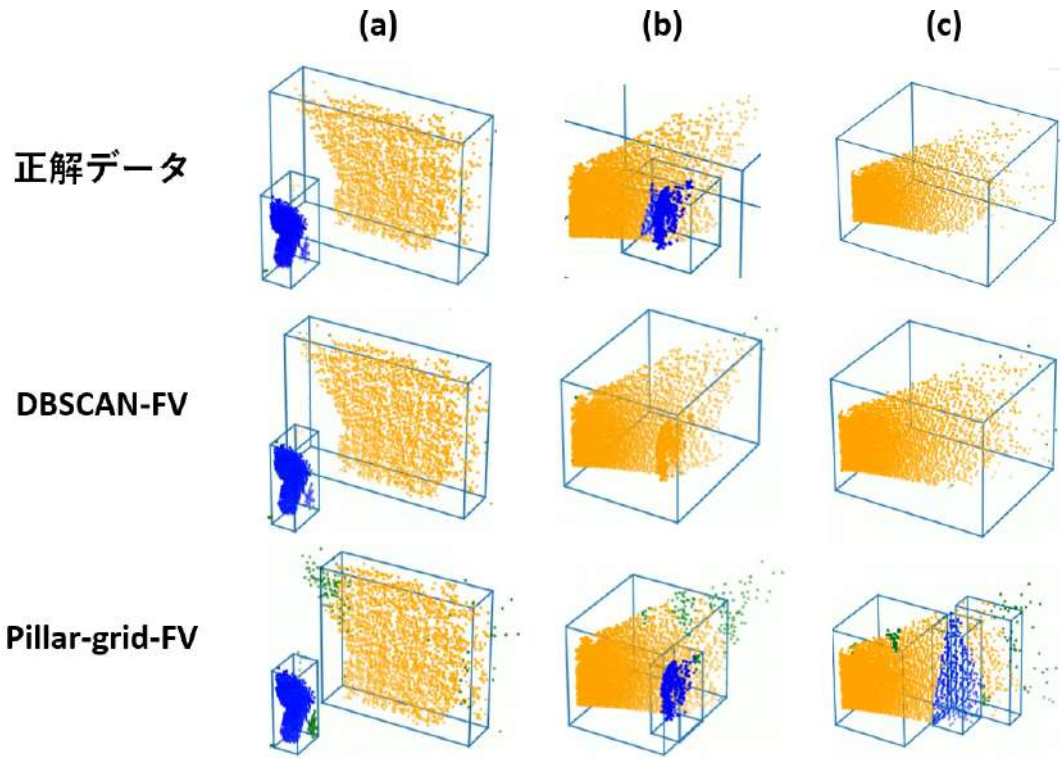

図 7　正解データ及び各手法のセグメンテーション結果: 黄色は壁，青色は人，緑色はノイズの点を示す

5.5　ウェアラブルデバイスとしての利用

開発中のひとなび-μ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出来るかを検証する．歩行中デバイスは不安定な状態にあり 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．

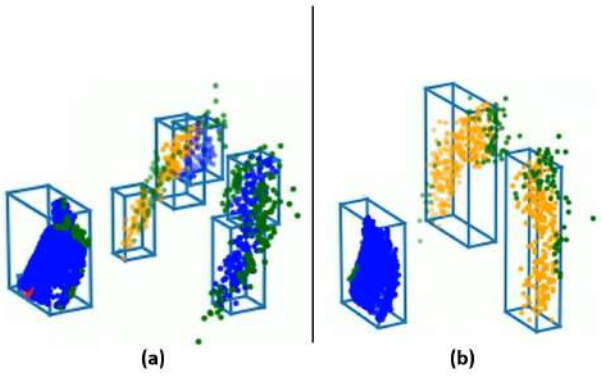

図 8　歩行時に取得した点群データのセグメンテーション結果

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 8 に示す．(a) に示す点群はデバイスが上方に 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「ひとなび-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 による離散化とグリッドベースのクラスタリング，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，検出精度は 96.1 mAP を達成し，エッジデバイス上での処理速度は 59.3（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された．今後は，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．

謝辞

本研究は，JST A-STEP JPMJTR20RV の助成を受けたものです．

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⟩（参照 2022-06-24）．

[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] Yamaguchi, H., Hiromori, A. and Higashino, T.: A Human Tracking and Sensing Platform for Enabling Smart City Applications, *Proceedings of the Workshop Program of the 19th International Conference on Distributed Computing and Networking*, Workshops ICDCN '18, New York, NY, USA, Association for Computing Machinery, (online), DOI: 10.1145/3170521.3170534 (2018).

[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 Bennamoun, 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] Yamada, S., Rizk, H. and Yamaguchi, H.: An Accurate Point Cloud-Based Human Identification Using Micro-Size LiDAR, *2022 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*, pp. 569–574 (online), DOI: 10.1109/PerComWorkshops53856.2022.9767322 (2022).

[10] 大河内悠磨，Hamada Rizk，山口弘純：軽量可搬型 3 次元空間センシングデバイスの設計開発，IPSJ マルチメディア、分散、協調とモバイル (DICOMO2022) (2022).

[11] 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).

[12] 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).

[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] 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).

[16] 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).

[17] 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).

[18] 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).

[19] 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).

[20] 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).

[21] 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).

[22] 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).