情報処理学会論文誌　Vol.62　No.1　246–253 (Jan. 2021)

人検出タスクにおける単体LRF環境での深層学習モデルの提案とその評価

小原 裕輝[1]　中沢 実[1,a)]

受付日 2020年3月31日, 採録日 2020年10月6日

概要：単体のレーザレンジファインダ（LRF）を用いた人の検出手法は，貨物運搬ロボットなどに使われる．これらの検出では，入力が検出対象であるかどうかの判定に，hand-crafted 特徴量と正であるか偽であるか分類する 1 クラス分類モデルを使用した手法が適用されている．本論文では，1 クラス分類モデルへの入力を hand-crafted 特徴量ではなく，深層学習モデルによって生成された特徴量に置き換えた手法を提案する．実験では，hand-crafted 特徴を使用した手法との検出率の比較を行い，パフォーマンスが一部向上したことを示す．また，ロボットの利用など実応用のために，Jetson Nano を使用して提案手法の処理速度の評価を行い実用可能性を評価した．

キーワード：レーザレンジファインダ，深層学習，点群，1 クラス分類

Proposal and its Evaluation of Deep Learning Model for Human Detection Task in a Single LRF Environment

Yuki Kohara[1]　Minoru Nakazawa[1,a)]

Received: March 31, 2020, Accepted: October 6, 2020

Abstract: Human detection using a single laser range finder (LRF) has been used for autonomous vehicles such as cargo carrying robots. In these detections, a method using hand-crafted feature values and a one-class classification model is applied to determine whether an object is a detection object or not. In this paper, we propose a method in which the input to a one-class classification model is replaced by features generated by a deep learning model instead of hand-crafted features. In the experiment, the detection rate was compared with the method using the hand-crafted feature. Furthermore, the processing speed was measured using the Jetson Nano to verify its practical applicability.

Keywords: laser range finder, deep learning, point cloud, one-class classification

1. はじめに

レーザレンジファインダ（LRF）は周囲の環境の形状を読み取ることができ，自己位置推定と環境地図の作成を同時に行う Simulateous Localization and Mapping（SLAM）や設置されているオブジェクトや障害物を認識する物体検知 [2]，歩行計測 [3] などが可能であり，研究の様々な場面で利用されている．その LRF を使った研究の中で，周囲の形状の中から追跡対象者を検知し，追跡する研究がある．この研究で養われた技術は人の代わりに荷物を運ぶ貨物運搬ロボットに使われ，実際に Doog 社の自動追従運搬ロボットのサウザー [4] は単体の LRF で人や同型のサウザーを検知，追跡することができる．従来の研究で LRF を用いた研究は広く行われており，単体の LRF を用いて人の検知と追跡を行っている研究も存在する [5], [6], [7], [8]．これらの研究が最新になるにつれて，**図 1** に示す処理の流れをとるようになり，点の分類（図 1 の (2)）では hand-crafted 特徴量と 1 クラス分類を採用するようになった．

近年，コンピュータビジョンの分野で既存の手法を超

[1] 金沢工業大学
Kanazawa Institute of Technology, Nonoichi, Ishikawa 921–8501, Japan
[a)] nakazawa@infor.kanazawa-it.ac.jp

© 2021 Information Processing Society of Japan

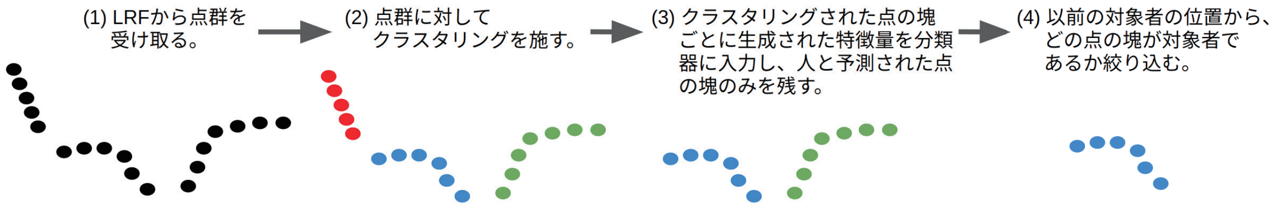

図 1 LRF による人の検出の処理フロー
Fig. 1 The processing flow of person detection by LRF.

える結果を生み出している深層学習と呼ばれる機械学習の手法がある．深層学習は既存の hand-crafted 特徴量を使わず，深層学習モデル自身が自動的に最適な特徴量を見つけ，それを学習することができる．深層学習モデルは hand-crafted 特徴量では表現しきれない箇所もとらえることができるため，hand-crafted 特徴量を使用した結果を超えることができる [9]．本論文では，単体の LRF を用いて人の検知と追跡を行うタスクの一処理である点の分類（図 1 の (2)）で扱われる特徴量の生成に深層学習手法を使用する．具体的には，点群を入力とする深層学習モデルから得られる特徴量を用いて，追跡対象者の検知手法で使われる特徴量との比較実験を行う．

また，単体 LRF による 1 クラス分類タスクは，豊富な計算処理資源を積むことができないロボット上でリアルタイムに行われることがある [4]．そのため，この 1 クラス分類タスクをこなすための本提案も，ロボットに積むことが可能なデバイスに制限して処理速度を計測する必要がある．しかしながら，深層学習手法は CPU だけでなく，GPU などの並列演算処理が可能なデバイスを使用しなければ高速な処理が不可能であることが多く，本提案手法もこれに該当する．近年，NVIDIA [10] が手頃な価格で GPU を載せた組み込みコンピュータを販売している．これらのコンピュータはロボットや車などで使用されることを想定しており，なかでも Jetson Nano [1] は小さいロボットの演算端末として利用することを可能にしている [11]．これほどのサイズと出力であるなら，様々なロボットのタスクに利用することができる．本実験では，この Jetson Nano を演算処理端末として利用することを想定した本提案手法の処理速度の測定を行う．

本論文の貢献は以下のとおりである．

(1) 単体 LRF 点群の 1 クラス分類に対する，深層学習モデルから得られる特徴量を利用した 1 クラス分類手法を提案した．
(2) 既存の手法から得られる特徴量と比較し，検出部位によっては深層学習モデルから得られる特徴量の方が高い精度を得られることを 1 クラス分類器を用いて実験的に示した．
(3) 提案手法における Jetson Nano 上での実効速度について示した．

本論文では，2 章で本研究に関連する研究の紹介，3 章で深層学習を用いた特徴量抽出モデル，4 章で訓練またはテストデータの取得手法，5 章で実験，最後に 6 章でまとめについて説明していく．

2. 関連研究

2.1 検知部位に関する研究

単体の LRF で追跡対象者を検知する既存の手法において，LRF で検知する部位が足と胴体で分かれている．単体 LRF を用いて追跡対象者を検知，追跡するタスクを部位ごとに説明する．

2.1.1 足の検知

奥迫らの手法 [5] は LRF を中心とした視点で，LRF から見て影となる部分のパターンをあらかじめ取得しておく．影の部分のパターンは LRF を中心とした区間ごとに種類が分けられており，その区間に影がある場合はその影にあうパターンがあるかどうかマッチングを行うことで検知することができる．Chung らの手法 [7] は，LRF から得られるデータを点群として扱い，その点群内に人と認識可能な点群があるかどうか調べている．点群は 3D 座標情報を持つデータ形式であり，2D の画像と比べてデータ群は軸に対して規則的には並ばず，距離は 3 次元のユークリッド距離で表現される．Chung らはクラスタリングされた点に対して，点群の特性を用いた 3 つの hand-crafted 特徴量を計算し，それを 1 クラス分類器に入力してそれが人の足であるかどうか検知している．図 **2** に示すように 3 つの特徴量は幅，胴回り，奥行きであり，座標値を持つ点 $\{P_i \mid i = 1, \ldots, m\}$ を使うことで，簡単かつ正確にそれらの特徴量を計算できる．

2.1.2 胴体の検知

胴体検知ではどちらも点群と 1 クラス分類を使った手法をとっている．Zainudin ら [6] は，LRF で取得した点群の座標データに沿った点群の分割を行い，分割したクラスタが人であるかどうかを検知するため，点群から hand-crafted 特徴量を作り出し，あらかじめ学習させた 1 クラス分類器に入力している．Jung ら [8] は胴体上部（脇付近）を検知部位としており，Chung ら [7] の特徴量に加えて角度と幅/胴回りを追加した後これらを標準化する．最後に Chung らと同じように 1 クラス分類器へ入力して人を検知して

© 2021 Information Processing Society of Japan

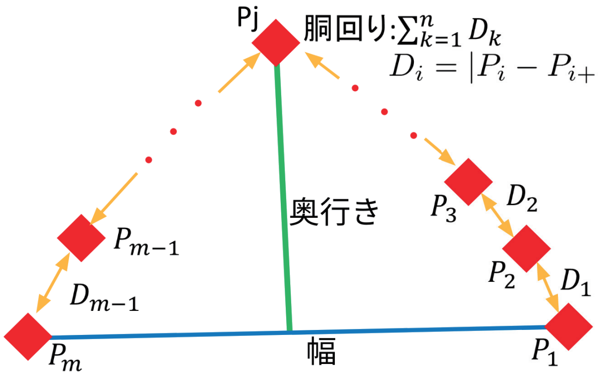

図 2　Chung らの特徴量（文献 [7] を参考に作成）
Fig. 2　Chung et al. features (created with reference to Ref. [7]).

いる．

2.2　深層学習

深層学習と単体 LRF を使った人の検知を行うにあたり，深層学習に点群を用いる．以下に hand-crafted 特徴量に対する深層学習の優位性と，点群での深層学習の活用について説明し，最後に 1 クラス分類と深層学習について言及する．

2.2.1　hand-crafted 特徴量と深層学習

2.1 節で説明したように，ほとんどの既存研究は hand-crafted 特徴量を 1 クラス分類器に入力して検出を行っている．hand-crafted 特徴量はその特徴量の設計者が自分の経験をもとに設計したものであり，またデータの細かな表現までとらえることは難しい．深層学習は訓練データからタスクに合った特徴量を自動で見つけることが可能であり，発見した特徴量は hand-crafted 特徴量よりも優れたパフォーマンスを示すことが多い．その代表的な例として，深層学習の技術を用いて作られた AlexNet [9] は画像分類において hand-crafted 特徴量を用いたモデルを超える結果を出している．

2.2.2　点群と深層学習

点群は 2 次元で軸に沿って規則的にデータ（ピクセル）が並んでいる画像と違って，不規則かつデータの順序を持たない 3 次元の表現が可能な形式である．不規則であるため，点群を画像と同じように深層学習を適応させることができない．そのため点群が存在する空間に，軸に沿ったボクセルを並べてその点群の情報をボクセルにまとめあげ，ボクセルに画像と同じような処理を施す手法をとった．この手法によって点群に深層学習を導入できた [13] が，この形式はボクセルに点群を合わせるという作業が必要なうえ，ボクセル化による細かい情報の喪失，ボクセルの解像度に依存する計算量によって手軽に扱えるものではなかった．この問題を解決したモデルが PointNet [12] である．PointNet はデータの順序の問題を解決したモデルであり，事前の準備なしにそのまま PointNet に点群を入力することができる．また Spatial Transformer Networks（STN）[14] をモデルに組み込むことで剛体変換に対応できるようになっている．PointNet の汎用性の高さは PointNet の公開後に作られた点群解析モデルに採用されていることからも分かる [15], [16]．本研究でも点群の深層学習モデルに PointNet を採用する．

2.2.3　1 クラス分類と深層学習

1 クラス分類は入力されたものが正のデータであるか負のデータであるかどうかを識別するだけの機械学習モデルである．1 クラス分類に似たものとして訓練またはテスト時に 2 つの異なる種類のデータを入力しそれを分類する 2 クラス分類があるが，1 クラス分類で扱う訓練データは正のデータだけである．そのため，本研究で扱うタスクのようにあるものだけを学習してほしい場合はこの 1 クラス分類モデルが使われる．1 クラス分類モデルとしてはサポートベクタマシンや SVDD などがあり，これらのモデルに分類したいデータの特徴量を入力することでクラスの予測を行う．深層学習モデルでは AutoEncoder [17] が 1 クラス分類に用いられることがある．AutoEncoder は入力データを低次元表現である code に変換する Encoder と code から入力データと同じものを出力する Decoder に分かれている．生成モデルとして扱われることが多いが，AutoEncoder で入力データに似た出力データを生成できるのは訓練時に AutoEncoder に訓練させたデータに近い入力データのみであるという条件があるため，この条件を生かし入力データと似ていない出力データを偽のデータであると見なす 1 クラス分類が可能となる．具体的には，訓練された AutoEncoder の Encoder から出力される code を特徴量として 1 クラス分類器に入力する [18]．本研究では，学習された AutoEncoder の Encoder を介して生成される特徴量を使い実験する．なお，近年では GAN を用いた 1 クラス分類モデル [19] や深層学習と SVDD に着目したモデル [20], AutoEncoder を発展させた Variational AutoEncoder [21] があるが，本研究では頻繁に扱われることが多く，なおかつシンプルな構造を持つ AutoEncoder を用いる．

3.　深層学習を用いた特徴量抽出モデル

本研究で使う深層学習を用いた特徴量抽出モデルには 2.2 節で述べた PointNet [12] と AutoEncoder [17] を組み合わせた PointNet-AutoEncoder [22] を使用する．PointNet-AutoEncoder を図 **3** に示す．図 3 の Encoder は n 個の点からなる入力点群から global feature と呼ばれる入力点群の特徴量を抽出するための PointNet のアーキテクチャである．global feature を点群出力へ再構築する Decoder は 3 つの全結合層であり，最後以外入力が 1024 次元で出力が 1024 次元の層となっている．最後の全結合層のみ入力が 1024 次元で出力が $n \times 3$ 次元となる．訓練時は，PointNet-AutoEncoder が出力点群を入力点群に近似する

© 2021 Information Processing Society of Japan

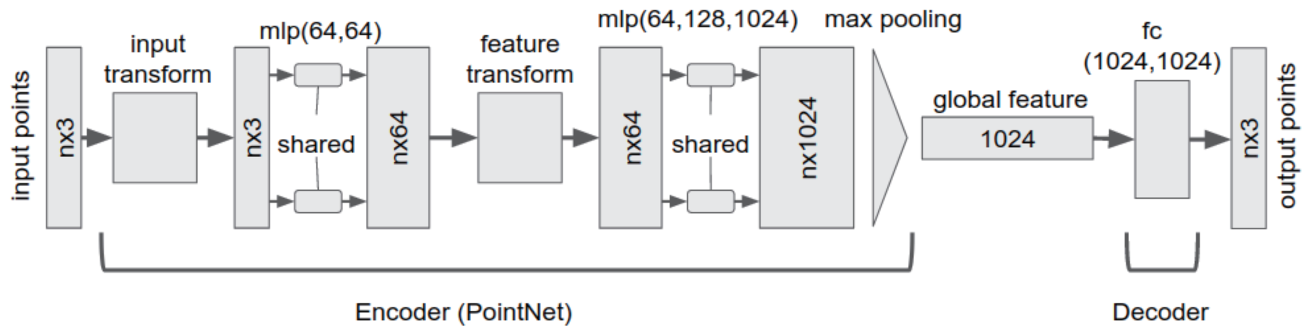

図 3　PointNet-AutoEncoder のアーキテクチャ（文献 [12] を参考に作成）
Fig. 3　PointNet-AutoEncoder Architecture (created with reference to Ref. [12]).

よう学習させる．また，2 次元画像の AutoEncoder は損失関数としてピクセルごとの差異を比べるが，点群はデータの並びに順番がないため入力した点群と出力した点群を点単位で順に直接比べることはできない．そこで，Chamfer distance と呼ばれる入力点群のある点とその点に最も近い出力点群の点の距離を扱う手法を損失関数とする．S_1 を入力点群，S_2 を出力点群，x を入力点群に含まれる点，y を出力点群に含まれる点としたとき，Chamfer distance の定義式は

$$CD(S_1, S_2) = \sum_{x \in S_1} \min_{y \in S_2} \{||x-y||_2^2\} + \sum_{x \in S_2} \min_{y \in S_1} \{||x-y||_2^2\}$$

である．

PointNet-AutoEncoder の学習が終了した後は，Encoder を特徴量抽出モデルとして，global feature を 1 クラス分類器への入力特徴量として扱う．

3.1　モデルの実装

PointNet-AutoEncoder は Chainer [23] によって実装される．実装は PointNet [12] の実装をベースにしている．PointNet-AutoEncoder の訓練時のエポック数は 250，最適化には Adam，学習率はエポック数によって段階的に変化するようになっている．学習率はエポック数が 10 になるまで 0.001，20 まで 0.003，100 まで 0.001，150 まで 0.0003，200 まで 0.0001，230 まで 0.00003，250 まで 0.00001 になるよう設定されている．

4.　単体 LRF を用いた各部位の点群データの取得手法

本研究では，単体 LRF を用いて点群データを取得し，機械学習モデルの訓練や実験に使うための足や胴体上部の点群を取得する．取得するための手順を以下に示す．

4.1　LRF の点群データに対する処理と保存

(1) LRF を中心として，前後の範囲は −1.0〜4.0 m，左右の範囲は −1.0 m〜1.0 m の範囲にある点群のみを残す．範囲を制限した理由は，LRF に映る点群は検知部位を得るための人のみである必要があり，これは人以外のオブジェクトが映った場合，そのオブジェクトが訓練またはテストデータに紛れ込むことを防ぐためである．

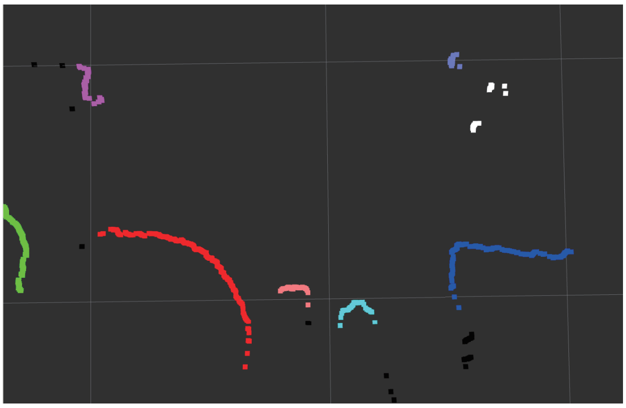

図 4　点群の分割
Fig. 4　Clustering point clouds.

(2) 次にクラスタリングを使用して点群を分割する．クラスタリングされた点群がどのように分割されるか可視化したものを図 4 に示す．図 4 では，クラスタリング結果が分かりやすいようにオブジェクトを並べている．これらの点群にクラスタリングをかける理由は，実際に追跡対象者を捕捉する際に処理される点群はクラスタリングを施した点群だからである．

(3) 分割された点群の中で最も点の数が多い塊を保存する．これは分割された点群をすべて保存してしまうと，LRF から見て足と足が重なってしまったときに，陰に隠れてしまった足の一部のみが映った点群を保存してしまうことを防ぐためである．

このアルゴリズムを行っている間，LRF に映っている人は歩行しながら検知部位まわりの形状を LRF に提供し点群を保存させる．歩行中の足などよく動く部分を読み取るときの検知部位の点群の形は変化しやすく，また，胴体上部などの楕円形に近い形状は読み取る角度によってその点

© 2021 Information Processing Society of Japan

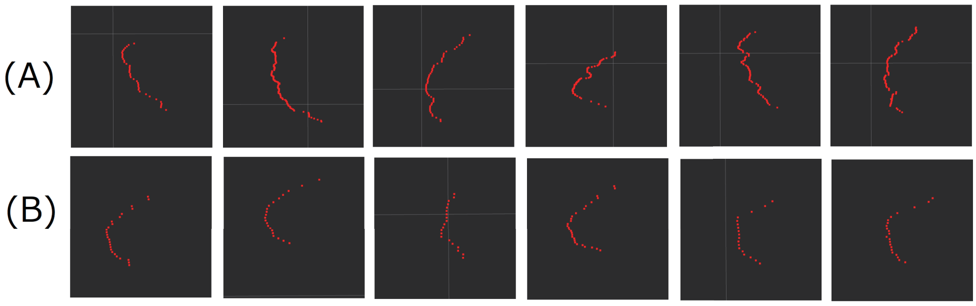

図 5 LRF から取得した点群データ
Fig. 5 Point cloud data obtained from LRF.

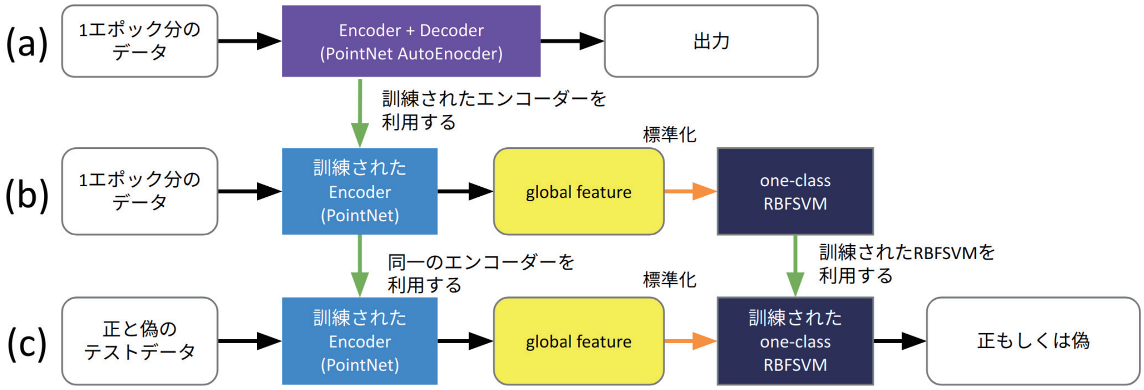

図 6 評価プロセス
Fig. 6 Evaluation process.

群の形が変わりやすいため前述した方法で多様なデータを採取する必要がある．LRF で読み取った点群を**図 5** に示す．図 5 の A が胴体上部，B が足の点群である．注意点として，図 5 の点群は視認しやすいように点群から視点までの距離と点のサイズを検知部位ごとに調節している．

4.2 アルゴリズムと実験環境

LRF で取得した点群データに対するクラスタリングには，Point Cloud Library（PCL）[24] のクラスタリング [25] を使用した．実装時のクラスタリングの設定では，点がどのクラスタに属するか決めるための基準値である tolerance 値を 0.06 m，クラスタと見なす点の数の最小値と最大値をそれぞれ 10 と 200 とした．なお，実験環境として範囲，応答時間，点数などで必要なスペックを十分に満たす LRF として，UTM-30LX [26] を使用した．

5. 実験

5.1 各部位の検知率比較

この実験では，4 章で説明した方法で検知部位の点群を取得したものを使った 1 クラス分類を行う．実験で扱う検知部位は既存の研究で採用していた胴体上部と足である．訓練データは胴体上部と足でそれぞれ 3,000 個ずつ用意され，データは対応する部位のみを映した点群を持つ．人の部位のみが映った 3,000 個の訓練データを用意し，これらを各手法で訓練に用いる．実験に用いるデータは部位ごとに用意した正のテストデータ 3,000 個と偽のテストデータ 3,000 個である．訓練データと正のテストデータには様々な向きから見た部位のデータが入っているが，データはすべて 1 人の人間から収集したデータであり，ズボンを着用している．偽のデータには各部位の高さでとれるデータを有しており，データ取得範囲には人が入っていないデータである．PointNet-AutoEncoder は**図 6** の (a) で訓練させる．PointNet-AutoEncoder のみ，3,000 個の訓練データのうち 600 個をバリデーションデータとして扱い，残りの 2,400 個を訓練データとして扱う．入力点数 n は，足では $n = 64$，胴体上部では $n = 128$ とする．点の数が足りないもしくは多い場合は，点群の点を n まで減らすもしくは点を複製するものとする．点群の値は 0 から 1 で表せるように正規化する．

訓練された後，PointNet-AutoEncoder には訓練データを 1 エポック分入力し，Encoder の出力に標準化を施したものを RBFSVM の 1 クラス分類モデルに入力し訓練させる（図 6 (b)）．テストデータで実験する際は，図 6 (c) に示すように，入力点群を Encoder に入力し Encoder の出力に標準化を施したものを訓練させた RBFSVM の 1 クラス分類モデルに入力しテストデータが正か偽か判別する．正と偽のラベルと出力される特徴量の相関性が高ければ高いほど，RBFSVM の 1 クラス分類モデルによる精度は良く

© 2021 Information Processing Society of Japan

表 1 1 クラス分類の結果

Table 1 One-class classification result.

部位	足 (%)			胴体上部 (%)		
データ	正	偽	合計	正	偽	合計
Chung らの 3 つの特徴量 [7]	89.26	84.50	86.88	93.16	68.0	80.58
Jung らの 5 つの特徴量 [8]	86.36	96.23	91.30	85.39	84.10	84.75
PointNet-AutoEncoder	83.39	98.26	90.83	90.90	92.50	91.70

表 2 点の数と精度の関係

Table 2 Relationship between number of points and accuracy.

部位	足 (%)			胴体上部 (%)		
点の数	正	偽	合計	正	偽	合計
32	77.43	97.80	87.61	88.40	88.06	88.23
64	83.39	98.26	90.83	90.83	88.56	89.70
128	75.70	99.13	87.41	90.90	92.50	91.70

なるため，ここではRBFSVMの1クラス分類モデルの精度を評価に使用する．比較として，Chungらの3つの特徴量 [7] と Jungらの5つの特徴量 [8] をRBFSVMの1クラス分類モデルで判別させたものを示す．また，検知部位ごとに n の数によって検知率の変化について示す．

結果：既存の手法による特徴量と PointNet-AutoEncoder を用いた特徴量の検出率を比較した結果について，**表 1** に示す．Chungらの3つの特徴量 [7] の手法は足を適切に検知できているものの，想定されていない胴体上部の検知に関しては，偽のデータの誤検知が目立っている．Jungらの5つの特徴量 [8] は標準化を行うことでChungらの特徴量に比べて偽のデータをうまく見分けることに成功しており，結果的に合計の検知率は上がっている．PointNet-AutoEncoderによる胴体上部の検知は，正と偽のデータに対して高い検知率を誇っており，合計値はどの手法よりも優れていた．その一方で足の正のデータを比較的正しく検知できなかった．理由として，点の数が少ないことに加えて，歩行中の衣服の動きにより，ネットワークにとってはとらえにくいランダムな形状になってしまったことが考えられる．

また，検知部位の点の数と検知率を比較したものを**表 2** に示す．胴体の点数は128，足の点数は64のとき最も精度が高くなることが分かる．点の数を任意の固定数にするために点群のサンプリングを行っており，今回は一般的なサンプリング方法であるランダムサンプリングを採用した．ランダムに点を増減させたが，たとえば全体の点の密度が等しくなるようなサンプリング方法で点の数を調節することで精度が上昇することも考えられる．

5.2 Jetson Nano を使用した処理速度の検証

この検証では，Jetson Nanoを使用して本提案手法の処理速度を計測する．計測区間は，PCLで使用できるクラスタリング [25] によって各点にクラスタのラベルが振り分けられた点群を受け取った時点から，RBFSVMの1クラス分類モデルで各クラスタの判別結果を得る時点までとする．この計測区間の処理フローは以下のとおりである．

(1) **取得**：最初に，クラスタリングによって得られるクラスタラベル付きの点群を受け取る．

(2) **前処理**：クラスタラベル付きの点群をクラスタごとのリストに変換し，各クラスタの中心座標を計算した後，各クラスタの点に対してランダムサンプリングと正規化を施す．

(3) **PointNet-AutoEncoder**：PointNet-AutoEncoderへ各クラスタの点を入力する．PointNet-AutoEncoderによって出力されたglobal featuresに対して標準化を施す．

(4) **RBFSVM の 1 クラス分類**：global features を RBFSVMに入力し，1クラス分類を行う．

この処理フローの中で，処理段階ごとに時間を取得する．取得するタイミングは図3より，前処理とPointNet-AutoEncoderの処理，RBFSVMの1クラス分類モデルの処理が終わるときとする．この処理フローの中で，GPUを使用する処理はPointNet-AutoEncoderだけである．また，この処理フローにかかる時間はクラスタの数によって変動するため，そのクラスタの数も記録する．計測結果はクラスタ数ごとにとった複数の記録の平均を，胴体をとる場合（$n = 128$）と足をとる場合（$n = 64$）に分けて示す．LRFから得られる点群は，5.1節でデータセットを取得したときの環境で得られる点群と同じであり，LRFの周りにはオブジェクトが並んでいるものとする．

結果：結果を**表 3**，**表 4** に示す．部位ごとにクラスタ数が変わっているのは，実施環境では足元にオブジェクトが多く，胴体上部にはオブジェクトが少ないためである．どのクラスタ数でもPointNet-AutoEncoderによる処理が処理時間のほとんどを占める結果となった．クラスタ数が1個の場合は40 ms台の処理が可能であるが，クラスタの数が増えるにつれて処理時間が長引くことが分かる．特に，クラスタが少ないときの前処理とRBFSVMの1クラス分類の処理時間はわずかであるが，クラスタが多い場合は処理時間が数倍になる．

また，この実験とは別にhand-crafted特徴量であるChungらの3つの特徴量 [7] とROS [27]，Raspberry pi 3 Model B+ [28] と自作のロボットを利用して実際に追跡

© 2021 Information Processing Society of Japan

表 3 足に対する処理時間計測結果
Table 3 Measurement results of processing time.

クラスタ数	11	12	13	14	15	16	17	18	19	20
前処理	6.8 ms	7.2 ms	8.0 ms	8.6 ms	9.0 ms	9.1 ms	9.4 ms	9.8 ms	10.4 ms	10.6 ms
PointNet-AutoEncoder	41.3 ms	39.6 ms	42.7 ms	42.0 ms	46.5 ms	44.4 ms	47.7 ms	46.0 ms	52.1 ms	52.3 ms
RBFSVM の 1 クラス分類	14.4 ms	15.5 ms	16.8 ms	17.9 ms	19.1 ms	20.5 ms	21.8 ms	23.1 ms	24.0 ms	25.8 ms

表 4 胴体に対する処理時間計測結果
Table 4 Measurement results of processing time.

クラスタ数	1	2	3	4	5	6	7	8	9	10
前処理	1.3 ms	1.8 ms	2.3 ms	2.8 ms	3.2 ms	3.5 ms	4.5 ms	5.2 ms	5.9 ms	6.3 ms
PointNet-AutoEncoder	38.7 ms	38.6 ms	38.5 ms	39.5 ms	40.8 ms	41.3 ms	41.1 ms	39.7 ms	43.1 ms	41.3 ms
RBFSVM の 1 クラス分類	1.8 ms	3.1 ms	4.5 ms	5.9 ms	7.2 ms	8.5 ms	9.9 ms	10.9 ms	12.3 ms	13.5 ms

対象者の検知と追跡を行った．50 ms ごとに追跡対象者の位置を更新するように動作させ，問題なく人を追跡することに成功している．さらに，中沢らの手法 [29] でも，人の歩く速度に対して LRF を用いた認識を 50 ms 以内で処理し，車椅子型のロボットを用いた実験を行っている．これらより，LRF とロボットを用いた人に対する認識と追跡タスクでも，LRF を介した認識は 50 ms 以内の処理が必要であると考えられる．PointNet-AutoEncoder を用いた手法の処理時間では 50 ms を超えてしまうケースがあるものの，それらはクラスタの数に依存するため，たとえばクラスタの数を減らすための簡易的なフィルタを通すことで 50 ms 以内の処理ができる可能性は十分にある．

6. まとめ

深層学習モデルを利用した追跡対象者の検知を行った．見る角度によって形が変化しやすい胴体上部の検知において，PointNet-AutoEncoder は的確に形状をとらえることができ，従来手法である hand-crafted 特徴量を超える検知率を出した．しかし，足の検知では，hand-crafted 特徴量の正のデータの検知率と比べて最大 5%劣るという結果も一部出している．また，Jetson Nano 上で PointNet-AutoEncoder と RBFSVM による推論における処理速度の計測を行った．点群のクラスタの数が増えるにつれて処理時間が長くなるため，この処理フローに渡すクラスタの数を絞るなどの工夫が必要になるが，ロボット上でも PointNet-AutoEncoder と RBFSVM によるリアルタイム処理ができる可能性があることが分かった．

今後の発展として，Ruff らの手法 [30] のように End-to-End な学習を行うモデルを作成することで，出力される特徴量によりフィットできるような分類器を作成できることが考えられる．また，本手法では人を検出することのみに着手したが，Doog 社のサウザー [4] のように他のロボットを検出する場合もある．そのため，マルチクラス分類などの 1 クラス分類以外のタスクを導入することも考慮した設計も必要となるだろう．

参考文献

[1] NVIDIA：モダン AI のパワーを数百万のデバイスへ—NVIDIA Jetson Nano，NVIDIA（オンライン），入手先 ⟨https://www.nvidia.com/ja-jp/autonomous-machines/embedded-systems/jetson-nano/⟩（参照 2020-03-30）．

[2] 兼時淳人，小林博明：レーザーレンジファインダを用いた自律移動ロボットによる路面障害物検知に関する研究，自動制御連合講演会講演論文集，Vol.51, pp.269–269（オンライン），DOI: 10.11511/jacc.51.0.269.0 (2008).

[3] 小澤真裕美，萬 礼応，松村哲哉，高橋正樹：レーザレンジファインダを用いた歩行計測システムの提案，日本機械学会論文集 C 編，Vol.79, No.801, pp.1550–1560（オンライン），DOI: 10.1299/kikaic.79.1550 (2013).

[4] 株式会社 Doog：株式会社 Doog—サウザー，株式会社 Doog（オンライン），入手先 ⟨http://jp.doog-inc.com/product-thouzer.html⟩（参照 2020-03-30）．

[5] 奥迫伸一，坂根茂幸：レーザレンジファインダを用いた移動ロボットによる人の追跡，日本ロボット学会誌，Vol.24, No.5, pp.605–613（オンライン），DOI: 10.7210/jrsj.24.605 (2006).

[6] Zainudin, Z., Kodagoda, S. and Dissanayake, G.: Torso detection and tracking using a 2D laser range finder, *Proc. 2010 Australasian Conference on Robotics and Automation, ACRA 2010* (2010).

[7] Chung, W., Kim, H., Yoo, Y., Moon, C. and Park, J.: The Detection and Following of Human Legs Through Inductive Approaches for a Mobile Robot With a Single Laser Range Finder, *IEEE Trans. Industrial Electronics*, Vol.59, No.8, pp.3156–3166 (2012).

[8] Jung, E., Lee, J.H., Yi, B., Park, J., Yuta, S. and Noh, S.: Development of a Laser-Range-Finder-Based Human Tracking and Control Algorithm for a Marathoner Service Robot, *IEEE/ASME Trans. Mechatronics*, Vol.19, No.6, pp.1963–1976 (2014).

[9] Krizhevsky, A., Sutskever, I. and Hinton, G.E.: ImageNet Classification with Deep Convolutional Neural Networks, *Proc. 25th International Conference on Neural Information Processing Systems–Volume 1*, Red Hook, NY, USA, Curran Associates Inc., pp.1097–1105 (2012).

[10] NVIDIA：NVIDIA の AI（人工知能）コンピューティングにおけるリーダーシップ，NVIDIA（オンライン），入手先 ⟨https://www.nvidia.com/ja-jp/⟩（参照 2020-03-30）．

[11] NVIDIA：NVIDIA JetBot パートナーの AI ロボットキット—NVIDIA，NVIDIA（オンライン），入手先 ⟨https://www.nvidia.com/ja-jp/autonomous-machines/embedded-systems/jetbot-ai-robot-kit/⟩（参照

© 2021 Information Processing Society of Japan

2020-03-30).
[12] Charles, R.Q., Su, H., Kaichun, M. and Guibas, L.J.: PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation, *2017 IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), pp.77–85 (2017).
[13] Maturana, D. and Scherer, S.: VoxNet: A 3D Convolutional Neural Network for real-time object recognition, *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems* (*IROS*), pp.922–928 (2015).
[14] Jaderberg, M., Simonyan, K., Zisserman, A. and kavukcuoglu, K.: Spatial Transformer Networks, *Advances in Neural Information Processing Systems 28*, Cortes, C., Lawrence, N.D., Lee, D.D., Sugiyama, M. and Garnett, R. (Eds.), Curran Associates, Inc., pp.2017–2025 (online), available from ⟨http://papers.nips.cc/paper/5854-spatial-transformer-networks.pdf⟩ (2015).
[15] Deng, H., Birdal, T. and Ilic, S.: PPFNet: Global Context Aware Local Features for Robust 3D Point Matching, *The IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*) (2018).
[16] Qi, C.R., Yi, L., Su, H. and Guibas, L.J.: PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space, *Advances in Neural Information Processing Systems 30*, Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S. and Garnett, R. (Eds.), Curran Associates, Inc., pp.5099–5108 (online), available from ⟨http://papers.nips.cc/paper/7095-pointnet-deep-hierarchical-feature-learning-on-point-sets-in-a-metric-space.pdf⟩ (2017).
[17] Hinton, G.E. and Salakhutdinov, R.R.: Reducing the Dimensionality of Data with Neural Networks, *Science*, Vol.313, No.5786, pp.504–507 (online), DOI: 10.1126/science.1127647 (2006).
[18] Erfani, S.M., Rajasegarar, S., Karunasekera, S. and Leckie, C.: High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning, *Pattern Recognition*, Vol.58, pp.121–134 (online), DOI: 10.1016/j.patcog.2016.03.028 (2016).
[19] Schlegl, T., Seeböck, P., Waldstein, S.M., Schmidt-Erfurth, U. and Langs, G.: *Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery* (2017).
[20] Ruff, L., Vandermeulen, R., Goernitz, N., Deecke, L., Siddiqui, S.A., Binder, A., Müller, E. and Kloft, M.: Deep One-Class Classification, *Proc. 35th International Conference on Machine Learning*, Dy, J. and Krause, A. (Eds.), Proceedings of Machine Learning Research, Vol.80, Stockholmsmässan, Stockholm Sweden, PMLR, pp.4393–4402 (online), available from ⟨http://proceedings.mlr.press/v80/ruff18a.html⟩ (2018).
[21] Kingma, D.P. and Welling, M.: Auto-Encoding Variational Bayes (2013).
[22] charlesq34: charlesq34/pointnet-autoencoder: Autoencoder for Point Clouds, github (online), available from ⟨https://github.com/charlesq34/pointnet-autoencoder⟩ (accessed 2020-03-30).
[23] Tokui, S., Okuta, R., Akiba, T., Niitani, Y., Ogawa, T., Saito, S., Suzuki, S., Uenishi, K., Vogel, B. and Yamazaki Vincent, H.: Chainer: A Deep Learning Framework for Accelerating the Research Cycle, *Proc. 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ACM, pp.2002–2011 (2019).
[24] Rusu, R.B. and Cousins, S.: 3D is here: Point Cloud Library (PCL), *IEEE International Conference on Robotics and Automation* (*ICRA*), Shanghai, China (2011).
[25] pointclouds.org: Documentation – Point Cloud Library (PCL) – Euclidean Cluster Extraction, pointclouds.org (online), available from ⟨http://www.pointclouds.org/documentation/tutorials/cluster_extraction.php⟩ (accessed 2020-03-30).
[26] 北陽株式会社：測域センサ データ出力タイプ/UTM-30LX 商品詳細—北陽電機株式会社，北陽株式会社（オンライン），入手先 ⟨https://www.hokuyo-aut.co.jp/search/single.php?serial=21⟩（参照 2020-03-30）.
[27] Foundation, O.S.R.: ja – ROS Wiki, Open Source Robotics Foundation (online), available from ⟨http://wiki.ros.org/ja⟩ (accessed 2020-03-30).
[28] Foundation, R.P.: Buy a Raspberry Pi 3 Model B+ – Raspberry Pi, Raspberry Pi Foundation (online), available from ⟨https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/⟩ (accessed 2020-03-30).
[29] 中沢 実，阿部拓真，杉下大河，佐野渉二：L0 ノルム最小化と多重解像度法を用いた室内向け自律移動車椅子の開発，情報処理学会論文誌，Vol.58, No.12, pp.2060–2069 (2017).
[30] Ruff, L., Vandermeulen, R., Goernitz, N., Deecke, L., Siddiqui, S.A., Binder, A., Müller, E. and Kloft, M.: Deep One-Class Classification, *Proc. 35th International Conference on Machine Learning*, Dy, J. and Krause, A. (Eds.), *Proc. Machine Learning Research*, Vol.80, Stockholmsmässan, Stockholm Sweden, PMLR, pp.4393–4402 (2018).

小原 裕輝（学生会員）

2019 年金沢工業大学工学部情報工学科卒業．同年同大学大学院工学研究科情報工学専攻博士前期課程入学，現在在学中．

中沢 実（正会員）

1993 年金沢工業大学大学院工学研究科情報工学専攻博士前期課程修了．同年富士通研究所入社．1996 年金沢工業大学助手．2011 年同大学教授となり，現在に至る．博士（工学）．自律分散システム，ロボティクス，信号処理等に関する研究に従事．2016 年情報処理学会山下記念研究賞受賞．電子情報通信学会，IEEE 各会員．

© 2021 Information Processing Society of Japan