

1次元畳み込みニューラルネットワーク (1D-CNN) による全方位 LIDAR からの歩行者検出

國貞祐貴† 山下隆義† 藤吉弘亘†
† 中部大学

E-mail: kuni@mprg.cs.chubu.ac.jp

Abstract

自動運転支援システムにおいて、歩行者検出は歩行者と自動車の接触事故を軽減するための重要な技術である。従来の全方位 LIDAR による歩行者検出は全ての点群を取得後に、クラスタリング処理を行い、各クラスが歩行者か否かを判定する。そのため、処理結果が得られた時点で歩行者は近接している場合があり、制御が間に合わない可能性がある。そこで本研究では、撮影から処理結果の出力までを遅延なく直接扱うために、全方位 LIDAR の波形データを直接扱う 1次元畳み込みニューラルネットワーク (1D-CNN) による歩行者検出手法を提案する。提案手法は全方位 LIDAR の波形データの一部分を逐次 1D-CNN に入力し、各点群が歩行者か否かを判定し、クラスタリングを行う。これにより、全方位 LIDAR の回転とともに歩行者検出ができるため、歩行者の検出した位置と実際の位置のずれを小さくすることが可能となる。

1 はじめに

自動運転の研究は非常に注目されており [9][10]、自動運転に向けて、周辺に存在する物体認識や、歩行者と自動車の接触事故を軽減するための歩行者検出は、重要な技術である。歩行者検出手法として、全方位の 3次元情報を取得できる Light Detection and Ranging(LIDAR) を用いる方法がある [1]-[8]。LIDAR は、レーザを照射して物体の反射光を感知するまでの時間を計測して、3次元情報や反射強度を取得する技術である。レーザレーダは主に障害物を精度良く計測するために使用されている [11]。さらに、可視光ベースの歩行者検出 [12][13] や、Convolutional Neural Network(CNN)[14] を用いた歩行者検出では検出が困難な夜間においても、LIDAR は輝度や形状の変化に頑健なため歩行者検出が可能である。全方位 LIDAR を用いた歩行者検出では、3次元情報に路面除去と立体物のクラスタリングを行い、各立体物を識別器により歩行者を判定する方法が一般的である。しかし、歩行者が他の物体に近接して存在する場合や歩行者から得られる点群数が少ない場合に、立体物のクラスタリングの際に同一物体として判定され

たり、立体物として判定されずクラスタリングに失敗することがあり、歩行者の検出が困難となるケースがある。また、検出する際に全方位 LIDAR から 3次元情報を抽出してから検出処理を行うため、検出した位置と実際の位置にずれが生じる問題がある。

本研究では、LIDAR から取得した 3次元情報の距離値を 1次元波形データとみなし、直接畳み込み処理を行う 1 Dimensional-Convolutional Neural Network(1D-CNN) を提案する。1次元波形データを 1D-CNN に入力し、クラスタリング処理の前に歩行者の点群を検出することで、歩行者検出の高精度化と同時に処理の高速化を図る。

2 関連研究

全方位 LIDAR から取得した 3次元情報を用いた物体検出では様々な手法が提案されている。Support Vector Machine(SVM) では、3次元情報から複数の特徴量を取得して歩行者を検出する Kidono らの手法がある [2]。また最近では、3次元情報を 2次元の点群マップに変換し、Fully Convolutional Network(FCN) によって検出する手法 [3] や、3次元情報を直接畳み込み層に入力する手法 [4][5]、3次元情報をボクセルに分割した特徴を畳み込み層に入力する手法 [6][7] などが存在する。さらに、鳥瞰図、正面図、RGB 画像を用いて各入力に対して畳み込み処理をする Multi-View[8] がある。本章では LIDAR データを用いた関連手法について述べる。

2.1 SVM による歩行者検出

Kidono らの手法 [2] では、LIDAR から取得した 3次元情報を立体物と路面の 2クラスに分割する。立体物の点群に対して、一定距離内に存在する 3次元情報をグループ化して、複数の歩行者候補を抽出する。歩行者候補に対して、Premebida らの特徴量 [15]、Navarro-Serment らの特徴量 [16]、Kidono らが提案した歩行者の凸凹形状特徴であるスライス特徴、照射した物体の反射特性 [17] を用いた反射強度分布を含めた合計 9つの特徴量を用いて、SVM より識別を行っている。

2.2 FCN による物体検出

FCN を用いた手法 [3] では、3次元情報を変換して FCN に入力することで検出を行う。まず、LIDAR か

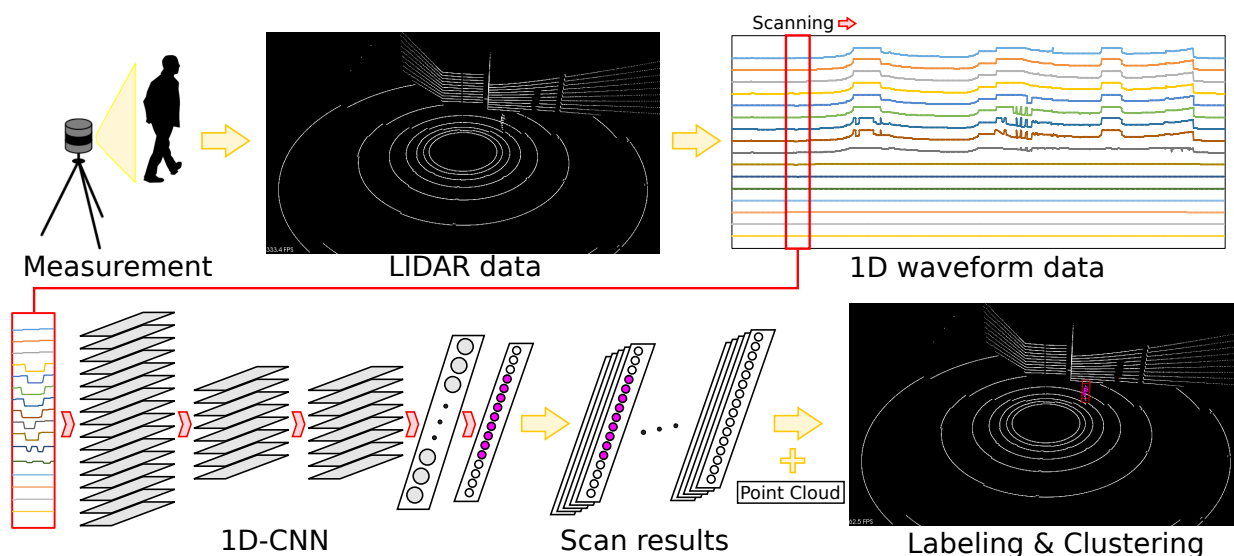


図1 提案手法の流れ

ら取得した3次元情報を2次元の点群マップに変換する。そして、点群マップに対して、FCNによりオブジェクトと3次元境界を同時に検出する。

2.3 PointNet

PointNet[4]では、3次元情報を直接畳み込み層に入力することでクラス分類やセグメンテーションを行うネットワークである。PointNetは、Spatial Transformer Network(STN)、Classification Network、Segmentation Networkの3つから構成される。まず、STNにより入力点群に対してノイズを軽減する。次に、Classification Networkにより畳み込み処理から各点群ごとの特徴抽出を行い、Max poolingから全体の特徴抽出することでクラス分類を行う。Segmentation Networkは、Classification Networkを拡張したものであり、各点群ごとの特徴と全体の特徴を結合したものを入力として各点群に対してセグメンテーションを行う。

2.4 VoxelNet

VoxelNet[6]では、3次元情報をボクセルに分割し、畳み込み処理によって3次元物体検出するネットワークである。VoxelNetは、Feature Learning Network(FLN)、Convolutional Middle Layers、Region Proposal Network(RPN)の3つの構成になっている。まず、FLNにより3次元情報を等間隔の3次元ボクセルに分解し、各ボクセルの内の形状情報を取得する。次に、Convolutional Middle Layersによってボクセル単位の特徴を集約する。最後に、RPNの畳み込み処理によって3次元物体検出を行う。

2.5 Multi-View

Multi-View[8]では、鳥瞰図と正面図、RGB画像を用いて各入力に対して畳み込み処理を行いクラス分類と3次元境界を検出するネットワークである。Multi-Viewは、3D Proposal Network、Region-based Fusion

Networkの2つの構成になっている。まず、3D Proposal Networkから鳥瞰図、正面図、RGB画像の各入力に対して畳み込み処理を行う。そして、Region-based Fusion Networkによって、ROI Poolingによる各特徴のサイズ合わせと、各特徴を組み合わせることでクラス分類と3次元境界を求める。

2.6 従来手法の問題点

従来手法の問題点として、歩行者付近に他の物体が存在した場合や点密度が低い歩行者は、クラスタリング処理を行う際に他の物体が歩行者と同一物体として判定されたり、クラスタリング処理が失敗したりする可能性があるため、検出が困難となる。また、検出する際に全方位LIDARから3次元情報を抽出してから検出処理を行うため、自動車などに搭載して歩行者を検出する場合、常に移動していることにより、検出した位置と実際の位置にずれが生じる問題がある。

3 提案手法

図1に提案手法の流れを示す。全方位LIDARの各レーザから順に取得した距離値を1次元波形データとみなし、1D-CNNより、各点に対して歩行者か否かを判定する。歩行者と判定した点群に対してクラスタリング処理を行い、歩行者領域(3次元長方体)を検出する。これにより、クラスタリング処理の際に歩行者の点群に対してのみクラスタリング処理を行うため、歩行者付近に他の物体が存在しても検出が可能となり、クラスタリング処理の高精度化も期待できる。さらに、1D-CNNの判定領域を走査することで、全方位LIDARの回転と共に歩行者の識別が可能となるため、検出した位置と実際の位置のずれを小さくすることが期待できる。以下に各処理の詳細を述べる。

3.1 1次元波形データの生成

本研究で使用する全方位 LIDAR は、Velodyne 社製の VLP-16 を用いる。全方位 LIDAR の仕様を表 1、点群の可視化結果を図 2 に示す。

表 1 Specification of LIDAR

Item	Specification
Horizontal field of view	360°
Irradiation interval(Horizontal)	0.2°
Vertical field of view	30°(-15° ~ 15°)
Irradiation interval(Vertical)	2°
Measured distance	100[m]
Scanning rate	10[scans/s]

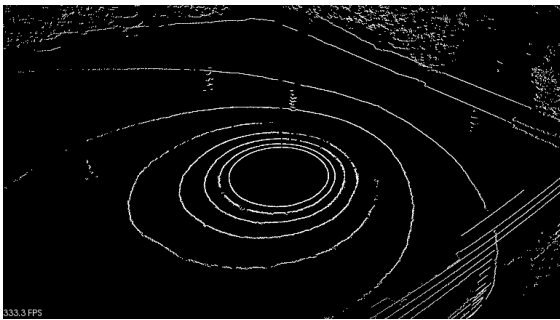


図 2 VLP-16 で撮影したデータの可視化

VLP-16 は、縦に 16 本のレーザを搭載し、全方位の 3 次元情報を取得する。0.2 度の間隔でレーザを全方位に照射するため、縦 16 ライン、横 1800 点の合計 28,800 点の 3 次元情報を取得することができる。全方位 LIDAR から取得した 3 次元情報の距離値を各レーザ ID に繋げることで、レーザ ID ごとの 1 次元波形データを生成する。1 次元波形データ例を図 3 に示す。

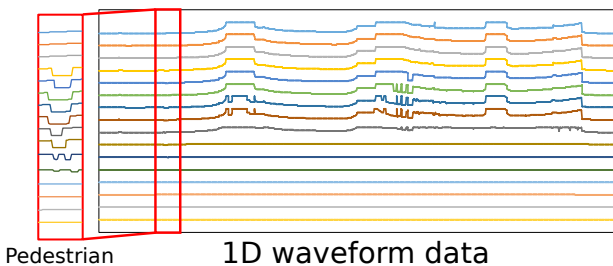


図 3 1次元波形データ

赤枠で囲んだ範囲は、歩行者に対する波形データであり、歩行者領域の距離値は LIDAR に近くなるため、1 次元波形データが歩行者の形状に合わせて凹んでいることがわかる。

3.2 外れ値の内挿

LIDAR の 3 次元情報には、空や鏡面体など反射光が取得困難な場所がある。もし、反射光が取得できなかつ

た場合、その場所の距離値を取得できず、外れ値となる。外れ値が含まれる場合、1 次元波形データの分布が従来の歩行者と大きく異なる。そのため、外れ値付近の距離値を内挿して補間する。内挿方法として 2 パターン検討する。1 つ目の方法は、外れ値の周りの中央値を利用して補間する。2 つ目の方法は、表 1 に示す LIDAR の仕様より、0° 以上のレーザ ID に対する外れ値には、LIDAR の照射上限距離である 100[m]、0° 未満のレーザ ID には照射角度と設置高さから算出した値を用いて補間する。

3.3 1D-CNN の構造

1D-CNN の構造を図 4 に示す。1D-CNN は、3 層の畳み込み層と 1 層の全結合層から構成されている。全ての畳み込み層において、1 次元重みフィルタを横軸方向のみに移動して畳み込み処理を行う。それにより、各レーザ ID ごとの特徴マップから各レーザ ID の歩行者を識別する。

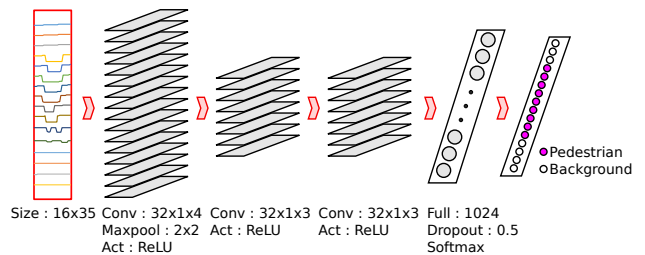


図 4 1D-CNN の構造

LIDAR で取得した 1 次元波形データの一定範囲 (判定領域) に着目して 1D-CNN に入力する。1 層目の畳み込み層では、入力データに対して 1 次元重みフィルタを畳み込み、マックスプーリングを行い、特徴マップを得る。2 層目の畳み込み層では、1 層目で取得した特徴マップに対して、1 次元重みフィルタを畳み込み、新たな特徴マップを得る。3 層目の畳み込み層では、2 層目で取得した特徴マップに対して、1 次元重みフィルタを畳み込み、新たな特徴マップを得る。そして、3 層目で得た特徴マップを特徴ベクトルに変換し、全結合層に与える。出力層で、各レーザ ID に対する歩行者と背景の確率を softmax 関数を用いて算出する。判定領域を走査して処理を繰り返すことで各点に対して歩行者か背景かの識別を行う。これにより、LIDAR の全方位に存在する歩行者を検出することが可能となる。

ネットワークの更新にはミニバッチ学習法を用いる。ミニバッチ学習法は 1 度に複数の学習サンプルを用いてパラメータの更新量を算出する方法である。バッチサイズを M 、教師ラベル T 、出力 O とした場合、誤差関数 E は式 (1) になる。

$$E = -\frac{1}{M} \sum_{m=1}^M \left(T_m \log O_m + (1-T_m) \log(1-O_m) \right) \quad (1)$$

3.4 クラスタリング

1D-CNNにより得られた歩行者検出結果と3次元点群を用いて、ラベリング処理とクラスタリング処理を行う。歩行者としてラベリングされた3次元点群に対して、クラスタリング処理を行い複数の歩行者候補を生成する。点群間のユークリッド距離が閾値以内の時に点群が5つ以上存在した場合、同一物体とする。点群間の距離の閾値は、3次元上でのユークリッド距離と x,y 軸の2次元上でのユークリッド距離の2パターン行う。3次元上でのユークリッド距離では、点群間の距離が $0.6[m]$ 以下の場合で行う。 x,y 軸の2次元上でのユークリッド距離は $0.25[m]$ 以下の場合で行う。3次元上でのユークリッド距離の場合、閾値が大きいため歩行者付近に誤検出した他の物体が存在した場合、同一物体となってしまう検出ができない可能性がある。そのため、点群が疎である z 軸を考慮せずにクラスタリング処理を行うことで、閾値を小さくすることが可能となり、3次元上でのクラスタリングでは検出困難な歩行者を検出できるようになる。次に、クラスタリング処理から生成された歩行者候補のサイズに基づいて歩行者を検出し、指定したサイズ条件に入る歩行者候補に対して3次元長方体を当てはめる。歩行者候補のサイズ条件を式(2)に示す。 w, l, h は、それぞれ幅、奥行き、高さを表している。

$$\begin{aligned} 0.1[m] < h < 2.0[m] \\ w < 1.0[m] \\ l < 1.0[m] \end{aligned} \quad (2)$$

サイズ条件の基準として身長 $2m$ 以下の歩行者を対象としている。1D-CNNにより歩行者の点群は検出できているので高さの下限を小さくすることが可能である。しかし、地面に対して誤検出が出る可能性があるため下限を $0.1m$ と設定している。

4 評価実験

1D-CNNによる歩行者検出の有効性を調査するために評価実験を行う。本実験では、1D-CNNによる各点群の識別結果に対する検出精度とクラスタリング処理による歩行者の検出精度の2つの評価と、関連手法との比較、実データでの評価を行う。1D-CNNによる各点群の識別結果に対する検出精度では、1次元波形データの外れ値の内挿方法と学習係数による比較を行う。クラスタリング処理による歩行者の検出精度では、クラ

スタリング時の距離の閾値を、3次元上でのユークリッド距離と x,y 軸の2次元上でのユークリッド距離にした場合で比較を行う。関連手法との比較では、Kidonoらの手法[2]と比較する。学習設定は、更新回数を500000回、ミニバッチサイズを20で学習を行い、歩行者か背景を判定するスコアの閾値を0.05から0.95までの0.05刻みで変えて評価する。データセットと評価結果を以下に述べる。

4.1 データセット

使用するデータセットは、CGの屋外データからVelo-dyne VLP-16と同じ設定で撮影して生成したデータセットを使用する。使用するサンプルデータ数は表5、サンプルデータ例を図5に示す。

表2 データセット枚数

Training data	Pedestrian	27889
	Background	172111
Evaluating data	LIDAR data	3000[Frame]

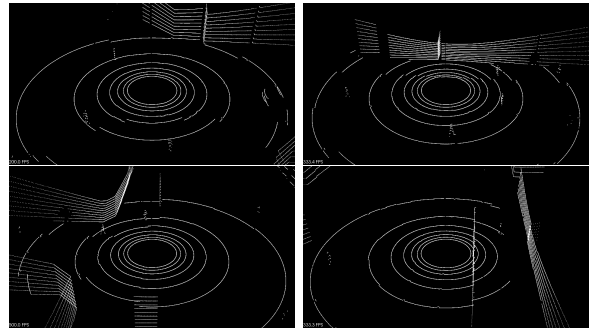


図5 サンプルデータの可視化

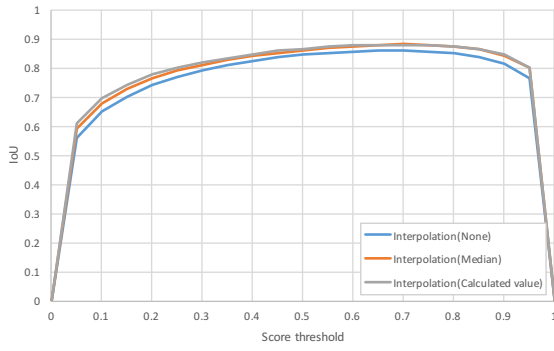
生成したデータセットには、建物とランダムに配置された歩行者と自動車が存在する。歩行者の設置位置はLIDARの設置位置から $20m$ 以内、人数は1人から10人、LIDARの設置高さは $0.8m$ 、LIDARの設置位置は車道上にランダムとする。障害物である自動車は、車道上にランダムに設置される。撮影場所によってはガードレールや標識などがあり、歩行者と自動車、建物以外の垂直物体も含まれる。

4.2 1D-CNNによる評価

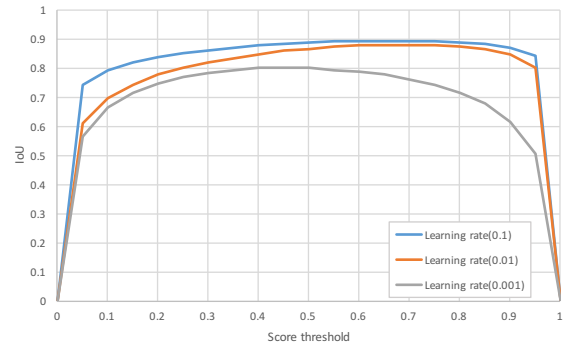
評価指標として、重なり率(IoU)を用いる。各点群に対して歩行者と背景でラベリングした結果と各点群の正解ラベルが正解しているかで評価を行う。 IoU は、True Positive(TP)、False Positive(FP)、False Negative(FN)を用いて算出する。 IoU の算出方法を式(3)に示す。

$$IoU = \frac{TP}{TP + FP + FN} \quad (3)$$

評価結果を図6に示す。縦軸が IoU の精度、横軸が

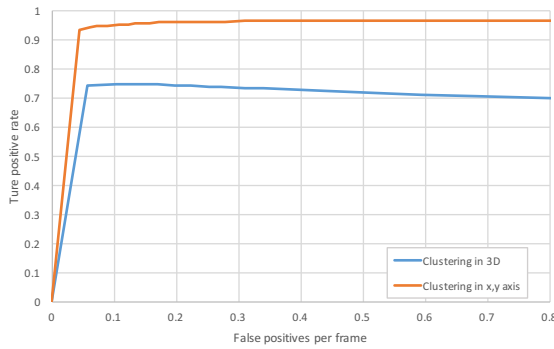


(a) 外れ値補間による評価結果

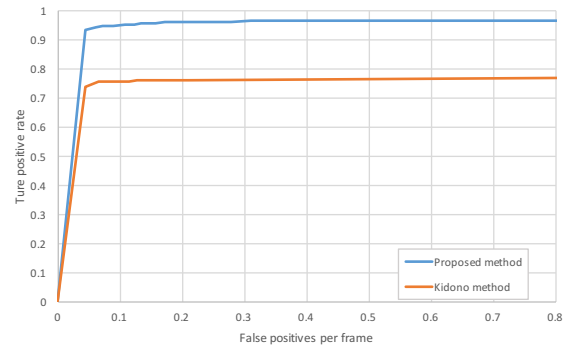


(b) 学習係数変更による評価結果

図6 IoU による評価結果



(a) クラスタリング変更による評価結果



(b) 比較手法との評価結果

図7 ROC 曲線による評価結果

1D-CNNにより歩行者か背景を判定するスコアの閾値である。この閾値が小さくなると歩行者の検出が多くなるが、誤検出も多くなる。一方で、閾値を大きくすると歩行者の検出数が少なくなり、誤検出も減る。

学習係数を0.01として、1次元波形データの外れ値の補間方法に、内挿なし、中央値での内挿、LIDAR仕様による内挿の3パターンを比較する。図6(a)から、閾値が0.7の時、中央値とLIDAR仕様の内挿を用いた場合、IoUが約88%となった。内挿なしの結果と内挿ありの結果を比較して約2%向上した。1次元波形データの外れ値に対して内挿した方が精度が向上することが確認できる。

次に、1次元波形データの補間方法をLIDAR仕様の内挿で、学習係数の変更について比較する。図6(b)から、学習係数を0.1と0.01、0.001で比較した場合、閾値が0.7の時、学習係数0.1とするとIoUが約89%が最も高くなった。

4.3 クラスタリング処理による評価

クラスタリングの評価指標として、ROC(Receiver Operating Characteristic) 曲線を用いる。検出した歩行者の中心と正解ラベルとのユークリッド距離を算出し、0.2[m]以内であれば検出成功とする。評価結果

を図7に示す。ROC 曲線は、グラフの縦軸は検出率 (*True positive rate*)、横軸は1フレームあたりの誤検出数 (*False positive per frame*) を示しており、性能曲線が左上に位置するほど精度が高いことを示している。*True positive rate* と *False positive per frame* の算出方法を式(4)に示す。

$$\begin{aligned} \text{True positive rate} &= \frac{\text{Positive detection count}}{\text{Pedestrians}} \\ \text{False positive per frame} &= \frac{\text{False detection count}}{\text{Frames}} \end{aligned} \quad (4)$$

1次元波形データの補間方法をLIDAR仕様による内挿とし、学習係数0.1とした場合の、3次元上でのユークリッド距離とx,y軸の2次元上でのユークリッド距離を用いてクラスタリングした精度について比較する。図7(a)から、1フレームあたりの誤検出数が約0.1個の時、3次元上でのユークリッド距離は、検出率が約75%、x,y軸上でのユークリッド距離は約94%の精度である。歩行者検出例を図8に示す。オレンジ色の点群は1D-CNNにより歩行者と判定された点群、紫色の点群はクラスタリング処理により同一物体として判定された点群、赤の直方体は歩行者領域を確定した場所であ

表 3 相違の比較結果

Method	CNN processing time per detection	Clustering processing time	Time from discovery to detection	Difference
Proposed method	0.00064[s]	0.0013[s]	0.0019[s]	0.021[m]
Kidono[2]	N/A	0.0852[s]	0.852[s] ~ 1.852[s]	0.946[m] ~ 2.057[m]

る。3次元上でのユークリッド距離によるクラスタリングでは、歩行者付近に他の歩行者が存在した場合、両歩行者を同一物体として判定しているが、 x,y 軸上でのユークリッド距離によるクラスタリングでは、閾値が小さくなっているため、歩行者付近に他の歩行者が存在しても個別に検出できていることが確認できる。さらに、遠い場所に存在する点群数の少ない歩行者も検出できていることが確認できる。

4.4 精度比較結果

1D-CNNによる提案手法とKidonoらの手法[2]の精度をROC曲線で比較する。評価結果を図7(b)に示す。

提案手法はKidonoらの手法[2]に比べて約20%検出精度を向上させることができている。Kidonoらの手法は歩行者付近に他の物体が存在するとクラスタリングの際に他の物体と同一物体にされてしまう。一方、提案手法は、先に歩行者点群を検出してからクラスタリング処理を行うため、点群数の少ない歩行者や、歩行者付近に他の物体が存在する場合でも検出することが可能である。

また、LIDARを40km/hで移動している自動車に搭載した場合、LIDARの測定を開始してから検出するまでの時間の比較した結果を表3示す。クラスタリング処理での処理時間はKidonoらの手法は全ての立体物に対して行うのに対して、提案手法では1D-CNNより検出した歩行者の点群に対してのみクラスタリング処理を行うためクラスタリング処理を高速に行える。Kidonoらの手法は全方位LIDARの3次元情報を取得してから処理を行うため、全方位の3次元情報を取得するのに0.1[s]かかる。そのため、3次元情報を取得した時点で歩行者は大きく移動しており、約0.8から0.18秒遅れて検出することになる。一方、提案手法では、LIDARの回転とともに判定領域を操作することで3次元情報の取得と同時に歩行者の検出が可能となる。そのため、3次元情報を取得後、歩行者を検出するまでの時間は約0.002[s]で行うことができる。自動車が40km/hで移動して正面に歩行者が存在した場合、Kidonoらの手法で検出した場合、歩行者が約2[m]移動する。一方、提案手法では、歩行者の移動は0.02[m]しか移動しないため、検出した位置と実際の位置のずれを限りなく小さくすることが可能であることがわかる。

4.5 実データでの検出結果

実環境における提案手法による歩行者検出の有効性を調査する。1次元波形データの補間方法をLIDAR仕

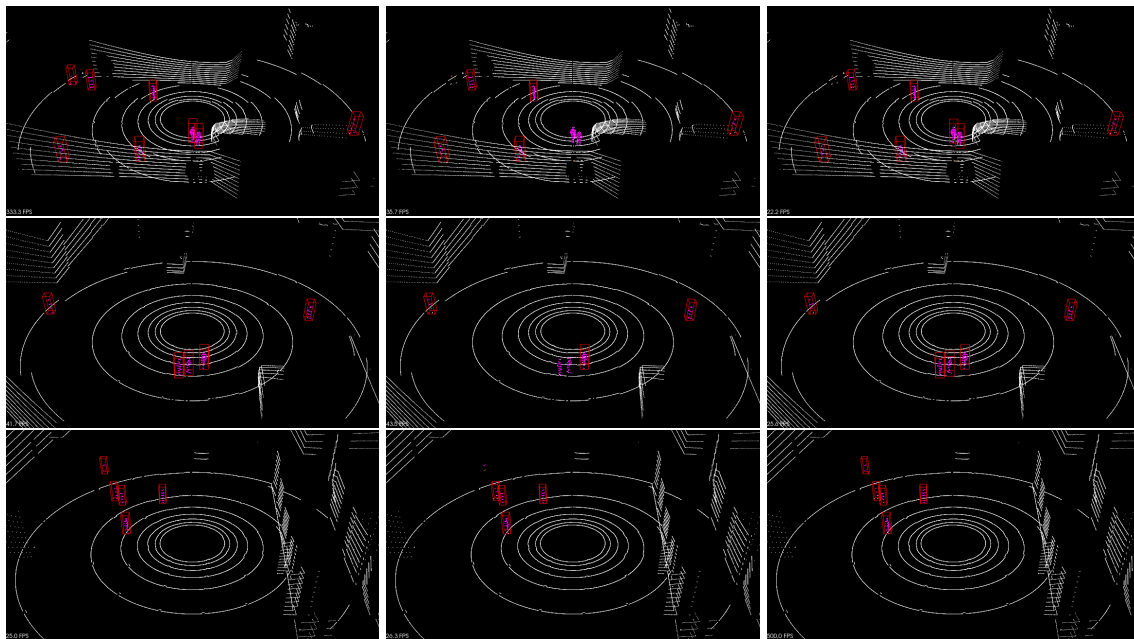
様による内挿、学習係数を0.1、クラスタリング方法を x,y 軸の2次元上でのユークリッド距離とする。実データでの歩行者検出結果を図9に示す。これにより、実環境における歩行者の検出にも適応可能である。

5 おわりに

本稿では、全方位LIDARの各レーダから取得した距離値を1次元波形データと、逐次1D-CNNに入力する歩行者検出法を提案した。全方位LIDARに適した1D-CNNにより先に点群に対して歩行者検出を行い、その後クラスタリング処理を行うことで、従来手法が困難な歩行者を検出可能とした。また、全方位LIDARの回転とともに判定領域を走査することで点群と属性を同時に取得できるため、検出した位置と実際の位置のずれを小さくした。今後は、1D-CNNの構造の検討を行い、更なる精度向上と高速化を目的とする。

参考文献

- [1] L. Spinello, M. Luber, and K. O. Arras, "Trackig People in 3D using a Bottom- Up Top-Down Detector", Proc.2011 IEEE Int.Conf. on Robotics and Autmation, pp.1304-1310, 2011.
- [2] K. Kidono, T. Miyasaka, A. Watanabe, T. Naito, and J. Miura, "Pedestrian Recognition Using High-definition LIDAR", IEEE Intelligent Vehicles Symposium, pp.405-410, 2011.
- [3] Bo. Li, Tianlei. Zhang, Tian. Xia, "Vehicle Detection from 3D Lidar Using Fully Convolutional Network", arXiv preprint arXiv:1608.07916, 2016.
- [4] Charles. R. Qi, Hao. Su, Kaichun Mo, Leonidas. J. Guibas, "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation", arXiv preprint arXiv:1612.00593, 2016.
- [5] Charles. R. Qi, Li. Yi, Hao. Su, Leonidas. J. Guibas, "PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space", arXiv preprint arXiv:1706.02413, 2017.
- [6] Yin Zhou, Oncel Tuzel, "VoxelNet: End to End Learning for Point Cloud Based 3D Object Detection", arXiv preprint arXiv:1612.06396, 2017.
- [7] Daniel Maturana, Sebastian Scherer, "VoxNet: A 3D Convolutional Neural Network for real-time



(a) アノテーションデータ (b) 3次元上でのユークリッド距離 (c) x,y 軸上でのユークリッド距離

図8 提案手法による歩行者検出例

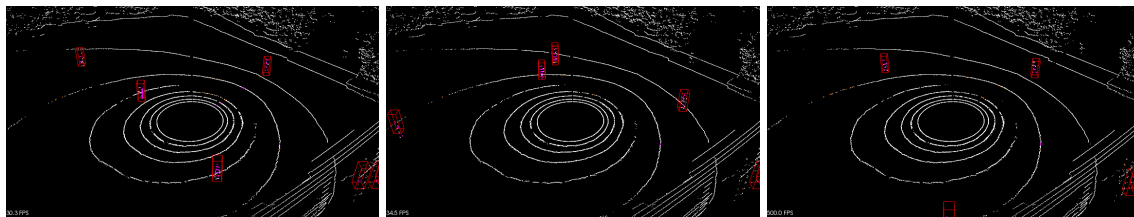


図9 実データによる歩行者検出例

- object recognition”, Intelligent Robots and Systems, 2015.
- [8] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, Tian Xia, “Multi-View 3D Object Detection Network for Autonomous Driving”, arXiv preprint arXiv:1611.07759, 2017.
- [9] A. Broggi, L. Bombini, S. Cattani, P. Cerri and R. I. Fedriga, “Sensing Requirements for a 13,000 km Intercontinental Autonomous Drive”, Proceedings of 2010 IEEE Intelligent Vehicles Symposium, pp.500-505, 2010.
- [10] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner et al., “End to End Learning for Self-Driving Cars”, arXiv preprint arXiv:1604.07316, 2016.
- [11] 大前学, 菅沼直樹, 清水浩, “レーザーレーダーを用いた自動車のインテリジェント化と自動運転”, Laser Review, vol.38, no.8, pp.565-570, 2010.
- [12] N. Dalal and B. Triggs, “Histograms of Oriented Gradients for Human Detection”, CVPR, pp.886-893, 2005.
- [13] Constantine Papageorgiou, Tomaso Poggio, “A Trainable System for Object Detection”, International Journal of Computer Vision, vol.38, no.1, pp.15-33, 2000.
- [14] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, “Backpropagation applied to handwritten zip code recognition”, Neural Computation, vol.1, pp.541-551, 1989.
- [15] C. Premebida, O. Ludwig and U. Nünse, “Exploiting LIDAR based Features on Pedestrian Detection in Urban Scenarios”, Proceedings of 12th Int. IEEE Conf. on Intelligence Transportation Systems, pp.18-23, 2009.
- [16] L. E. Navarro-Serment, C. Mertz, and M. Hebert, “Pedestrian Detection and Tracking Using Three-Dimensional LADAR Data”, Int. Journal of Robotics Research, Vol.29, No.12, pp.1516-1528, 2010.
- [17] D. Williams “Methods of Experimental Physics”, Academic Press, vol.13, 1976.