# Pedestrian Detection in Indoor Environments Based on 3D LiDAR Data Segmentation

WEIMIN WANG[1] KEI HIROI[2] KATSUHIKO KAJI[3] NOBUO KAWAGUCHI[2]

**Abstract:** With advantages like independence from visible light, detecting pedestrian with range data acquired by depth sensors or LiDAR sensors is attracting more interests. Especially, LiDAR sensors have been widely utilized in many fields such as autonomous system or robot vision for obstacle detection. However, LiDAR data are usually very sparse, which is a challenge for detecting pedestrian especially when the pedestrian is occluded by another object or far from the LiDAR. As an application in indoor mapping, the traces of pedestrians need to be eliminated for quality mapping. Compared with outdoor environment, indoor environment is usually very crowded and more objects like pillars or tables whose LiDAR scan lines are similar to a pedestrian's. This increases difficulty for pedestrian detection in indoor environment. In this paper, we propose a method that detects pedestrian in 3D directly from sparse 3D data acquired in indoor environment. We utilize a Velodyne HDL-32E LiDAR sensor in this work. 3D data from LiDAR sensor are clustered into scan line segments and segmented by agglomerating these clustered line segments. Then we find out the segment that posses specific features of pedestrians. Finally, The experimental result of pedestrians detection from point cloud data of an underground shopping mall acquired by the LiDAR sensor verifies the feasibility and robustness of the method we proposed.

## 1. Introduction

LiDAR(Light Detection and Ranging) sensors detect the distance by measuring the time difference between emission and transmission of a pulse laser beam. Thus, it is insensitive to visible light. In addition to this, it possess long effective range with high accuracy in a wide viewing angle (Table.1). For these advantages, LiDAR has been widely used in many fields such as security, robot vision or intelligent vehicles. Especially, it is a key device in autonomous system for detecting objects like pedestrians or other obstacles.

Table 1: Specification of popular LiDAR sensors

| Product | Viewing Angle | Effective Range |
|---|---|---|
| Velodyne HDL-64E | $360° \times 26.8°$ | $< 120m$ |
| Velodyne HDL-32E | $360° \times 41.3°$ | $1m - 70m$ |
| HOKUYO 3D-LIDAR | $210° \times 40°$ | $0.3m - 50m$ |
| HOKUYO UTM-30LX | $270° \times 0.25°$ | $0.1 - 30m$ |

As another application developed in recent years, LiDAR is also utilized for 3D indoor reconstruction and mapping. And pedestrians are often scanned by LiDAR sensors in this application as shown in Fig.1. Registration, which combines several point cloud data generated by LiDAR sensors to obtain dense and large one, is often needed. However, as shown in Fig.2, the accumulative traces of pedestrians after the registration could be seen obviously and this severely affect the quality of 3D indoor mapping. Thus, for better mapping, it is necessary to detect the pedestrians and eliminate the effect.

[1] Graduate School of Engineering, Nagoya University
[2] Institute of Innovation for Future Society, Nagoya University
[3] Faculty of Information Science, Aichi Institute of Technology

In the work of this paper, we propose a method that detects pedestrian in 3D indoor environments directly without the aid of any other additional information like RGB images. We utilize Velodyne HDL-32E LiDAR sensor to acquire 3D data of indoor environments in this work. Similar to other LiDAR sensors, Velodyne LiDAR HDL-32E measures distances with 32 vertical aligned lasers by Time Of Flight(TOF) method. It provides a 41.3°vertical and 360° horizontal field of view by rotating the head that emits 32 lasers around its vertical axis. For every 360° spin, a frame of point cloud consisting of 70,000 3D points is generated [1].

We first cluster the 3D LiDAR data acquired by Velodyne HDL-32E into scan line segments with a method called Jump Distance Clustering(JDC). Then, segmentation of the whole LiDAR data is done by agglomerating scan line segments into different segments by comparing the distance between centroids of two line segments with a user defined threshold. Finally, we find out the segments of pedestrians with features that satisfy conditions for pedestrians .

## 2. Related Work

There has been many mature methods for pedestrian detections with optical cameras base on the features such as HOG(Histogram of Oriented Gradients)[2] or Haar-like[3]. Since optical cameras capture images based on visible light, to detect pedestrians without the need of visible light, pedestrians detection with depth sensors like Kinect of Microsoft are also developed[4] [5] [6] . Either images captured by optical cameras or depth images by depth sensors are dense, which means the pixels are densely and evenly distributed. The methods we mentioned above perform well for dense and even image data. However, it will be a problem to ap-

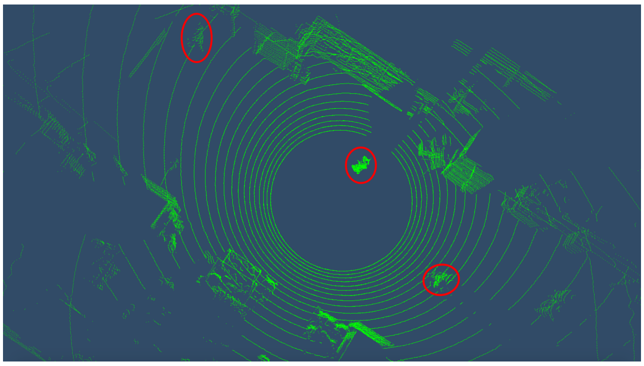


Fig. 1: A frame of raw 3D point cloud data acquired by a Velodyne HDL-32E LiDAR sensor, pedestrians are marked with red ellipse

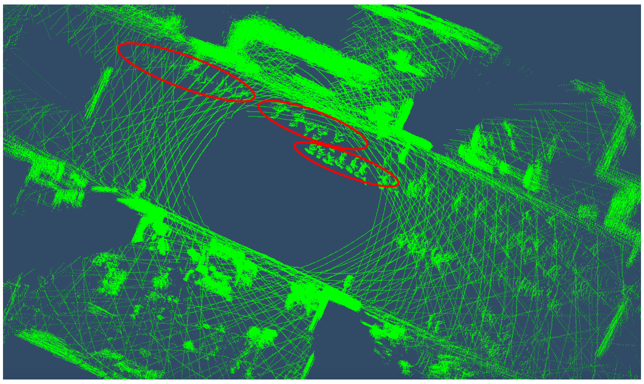


Fig. 2: A registered point cloud for indoor reconstruction and mapping, traces of pedestrians are marked with red ellipse

ply these methods to sparse 3D data generated by LiDAR sensors. For example, it is hard to calculate the histogram of oriented gradients for spare data.

A method that detects and tracks pedestrians by converting a 3D point cloud to a range image and segmenting the blocks with means shift algorithm is proposed in [7]. This method detects pedestrians in 2D rather than in 3D directly. Spinello proposed a method that detects pedestrians in 3D data by training a bank of classifiers for different scan line segments of a pedestrian in [8]. The scan line segments are generated by clustering neighboring points at the same height, which is called as JDC(Jump Distance Clustering) in [8]. For a new point cloud, all points in it are clustered as scan line segments with JDC method first, and every scan line segment is classified individually with the trained classifiers.

However, the experimental environment for the methods mentioned above is either outdoor environment or simple indoor environment [7] [8] [9]. And there is no method that detects a whole pedestrian as a 3D object in 3D directly. To detect pedestrian in crowded indoor environment, we propose the method that is able to detect pedestrians of dynamic crowded indoor environment in 3D directly without any other additional information. Although we also used the method called JDC which is as same as proposed in [8], the essential idea is different. Spinello use features of a single scan line segments while we use features of a whole 3D object consisted of several scan line segments.

The experimental result of pedestrians detection in point cloud data of an underground shopping mall acquired by the HDL-32E sensor verifies the feasibility and robustness of the method we proposed.

## 3. Data processing

This section explains the data processing to generate segments of point cloud for pedestrian detection in Section 4.

Main process of the method we propose for pedestrian detection is shown in Fig.3. For a frame of point cloud, we preprocess the point cloud by clustering it in several scan line segments for reducing computation and further processing, which is explained in Section 3.1.

The idea for segmentation is based on the observation that the distance between arithmetic centroid of scan line segments from a same object is much less than the distance between arithmetic centroid of scan line segments from different objects. And this feature is especially obvious for scan line segments from pedestrians. The implementation of segmentation is explained in Section 3.2.

Then the detection of pedestrians from segments generated in Section 3.2 is somewhat like binary classification of segments: pedestrians or not pedestrians. Conditions for classification are based on boundaries of a pedestrian segment. For example, if the height of a segment is 3 meters and then we know it is impossible for this segment to be a pedestrian. Details about detection are explained in Section 4.

In Section 5, the experimental result with this method is shown.

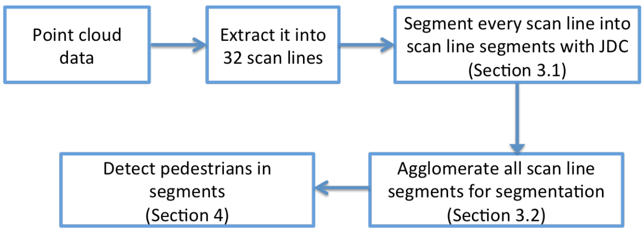



Fig. 3: Processes to detect pedestrians with the method we propose

### 3.1 Clustering

We utilize the same JDC(Jump Distance Clustering) method proposed in [8] for preprocessing. There are many ways to implement JDC. Since HDL-32E is utilized in this work, laser_id information of every point for the point cloud generated by HDL-32E is also stored, we implement JDC with the information of laser_id. Fig.4 is the point cloud that displayed according to the laser_id.

To explain the procedures of JDC simply, we declare notations as follows in this work:

- $\mathfrak{p}_{ij}$: A 3D point with the coordinate$(x_{ij}, y_{ij}, z_{ij})$, where $i$ is the laser_id and $j$ is the sequence number of the point

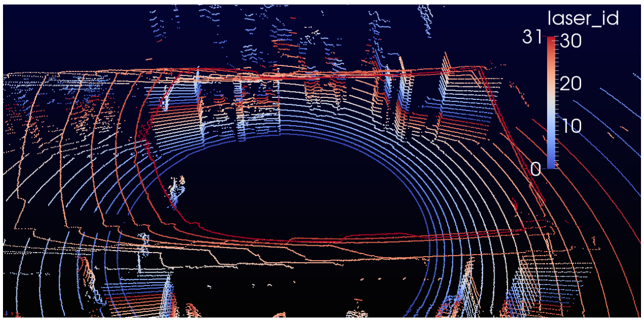



Fig. 4: Point cloud shown according to the laser_id

in the scan line of laser_id=$i$

- $\mathfrak{L}_i$: A scan line consists of all $\mathfrak{p}$ whose laser_id=$i$
- $\mathfrak{P}$: A point cloud consists of $\mathfrak{L}$. $\mathfrak{P} = \{\mathfrak{L}_i\}$, where $i$=0 to 31
- $\mathfrak{J}_{ij}$: A scan line segment after JDC, where $i$ is the laser_id and $j$ is the sequence number in the scan line segments after JDC of $\mathfrak{L}_i$

The detailed illustration is shown in Fig.5. For a point cloud $\mathfrak{P}$, we first exact it into 32 scan lines $\mathfrak{L}_{0..31}$ according to the laser_id. A $\mathfrak{J}$ is generated if the Euclid distance between two consecutive points exceeds a user defined threshold $jdc_{th}$. Illustrated in Fig.5, JDC is started from $\mathfrak{p}_{01}$, and the distance between $\mathfrak{p}_{03}$ and $\mathfrak{p}_{04}$ is greater than $jdc_{th}$, then the first scan line segment $\mathfrak{J}_{01}$ is generated as $\{\mathfrak{p}_{01}, \mathfrak{p}_{02}, \mathfrak{p}_{03}\}$. And $\mathfrak{p}_{03}$ will be the starting point of the second scan line segment $\mathfrak{J}_{02}$.

In the work, we set $jdc_{th} = 0.3m$.

Fig.6 shows the an example of point cloud after JDC. From Fig.6b, we can see the number of $\mathfrak{L}$ is much less than the number of points shown in Fig.6a. This could reduce the computation and make it easy for further processing.

Similar to [8], we also calculate some features of $\mathfrak{J}$ as following for segmentation step in Section 4 and pedestrian detection in Section 5.

- $f_1$:Centroid of $\mathfrak{J}$, the arithmetic value of all points $\mathfrak{p}$ in $\mathfrak{J}$.
- $f_2$:Total number of points in $\mathfrak{J}$
- $f_3$:Accumulative length, accumulation of distance between two adjacent points

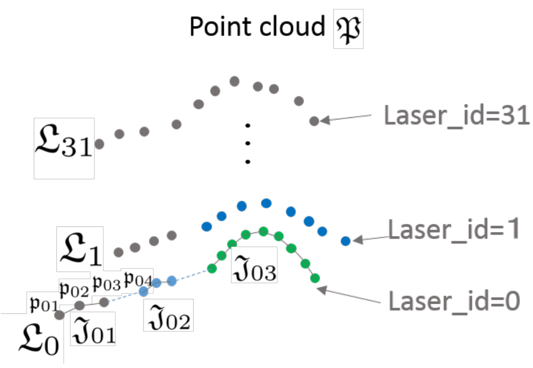
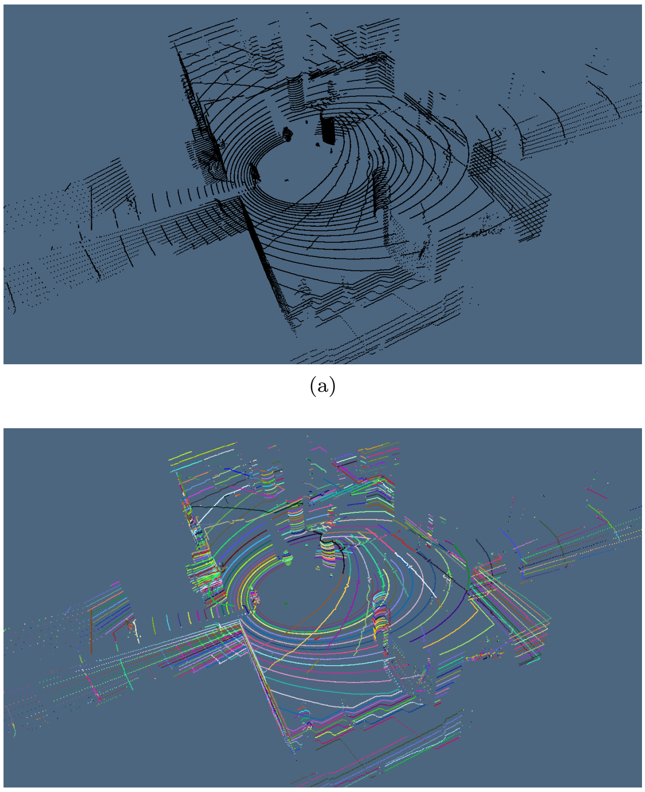



Fig. 5: Notations in this work



(a)

(b)

Fig. 6: JDC of a point cloud:(a)-A frame of raw 3D point cloud data acquired by Velodyne HDL-32E LiDAR sensor, (b)-JDC segments marked with different colors

### 3.2 Segmentation

We segment the point cloud by agglomerating $\mathfrak{J}_{ij}$ generated in Section 3.1. The main algorithm for agglomeration is shown in Algorithm 1.

For the set of all $\mathfrak{J}_{ij}$ generated in Section 3.1, we first put one $\mathfrak{J}$ into a set $\mathcal{S}$ and delete if from $\mathfrak{J}_{ij}$. Then for every $\mathfrak{J}$ in $\mathfrak{J}_{ij}$ we calculate the Euclid distance between the centroid of $\mathfrak{J}$ and the centroid of every element in $\mathcal{S}$. If there exist a scan line segmentation in $\mathcal{S}$ for $\mathfrak{J}_i$ that the distance between the centroids of them is less than a user defined threshold $aggl_{th}$, we agglomerate them as one segmentation.

In summary, we hierarchically cluster the data obtained by JDC. In this work we set $aggl_{th} = 0.5m$.

The result generated after agglomeration is a set of the set of $\mathfrak{J}_{ij}$, denoted as $\mathcal{A} = \{\mathcal{S}_n\}$ and $\mathcal{S}_n = \{\mathfrak{J}\}$, where $n$ is the sequence number of the sets after agglomeration of $\mathfrak{J}_{ij}$ shown as in Fig.7.

Every element $\mathcal{S}$ in $\mathcal{A}$ is considered a segment. The result of agglomerating the clustered data in Fig.6b once is shown in Fig.8a. From Fig.8a we can see that there are some parts(marked by red circles) are segmented separately while they are obviously supposed to be segmented as a whole. To agglomerate the parts like these, we consider iterating agglomerating Fig.6b with similar algorithm to Algorithm.1. The result after twice agglomeration is shown in 8b, and the number of segments is reduced to 620 from 646 in Fig.8a.

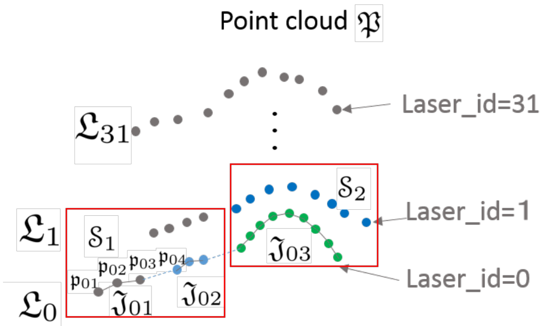



Fig. 7: Set of $\mathfrak{J}$

**Algorithm 1** Algorithm of agglomerating $\{\mathfrak{J}_{ij}\}$

**Require:** point cloud $\mathfrak{P} = \{\mathfrak{J}_{ij}\}$, $aggl_{th}$

$\mathcal{A}$ ←empty set
**while** $\{\mathfrak{J}_{ij}\}$ is not empty **do**
  $\mathcal{S}$ ← empty set
  add an element of $\{\mathfrak{J}_{ij}\}$ into $\mathcal{S}$, and delete it from $\{\mathfrak{J}_{ij}\}$
  **for all** elements $\mathfrak{J}$ in $\{\mathfrak{J}_{ij}\}$ **do**
    calculate the every Euclid distance $d$ between the centroid of $\mathfrak{J}$ and centroid of every element in $\mathcal{S}$
    **if** one distance in $d < aggl_{th}$ **then**
      add $\mathfrak{J}$ into $\mathcal{S}$
      delete $\mathfrak{J}$ from $\{\mathfrak{J}_{ij}\}$
    **end if**
  **end for**
  add $\mathcal{S}$ into $\mathcal{A}$
**end while**
**return** $\mathcal{A}$

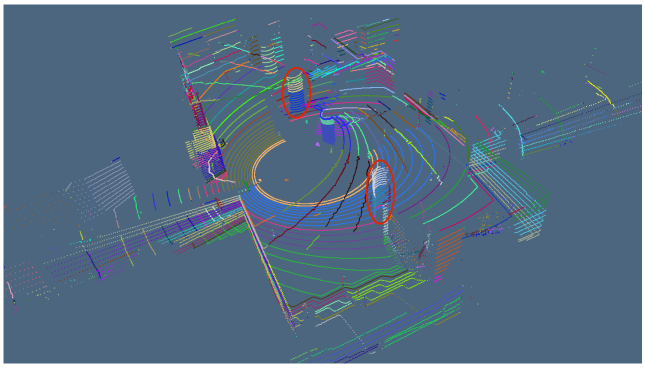
## 4. Pedestrian Detection

In Fig.8b, we can find that pedestrians have been segmented well from the surrounding environment after the agglomeration. As mentioned in Section 3, the detection of pedestrians from these segments can be implemented by finding the segments whose features satisfy the conditions for a pedestrian.

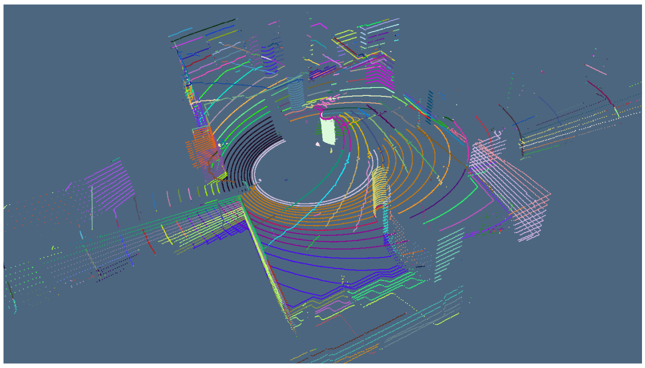
The features of a segment and the conditions for the features we used in this work is defined as follows:

- $f_1$:Bound. Bound of projection of the segment to x,y,z axis. $f_1$ should $> 2$.
- $f_2$:Maximum of accumulative length, the value of the biggest accumulative length of all scan line segments. $f_2$ should $> 50$.
- $f_3$:Total number of the points in the segment. $f_3$ should $> 50$.
- $f_4$:Number of scan line segments in the segment, $f_4$ should $> 6$

We calculated every feature defined above of every segment generated in Fig.8. The detected hypothesis marked with different colors is shown in Fig.9, while negative segments are shown with black color. We can see that segments of pedestrians are well detected and segments these are not pedestrians are considered as negative in this frame of point cloud. The result of point cloud after removing detected pedestrians is shown in Fig.10.



(a)



(b)

Fig. 8: Segmentation result after agglomerating scan line segments in Fig.6b with Algorithm.1, (a)-646 segments are generated after agglomerating once (b)-620 segments are generated after agglomerating twice

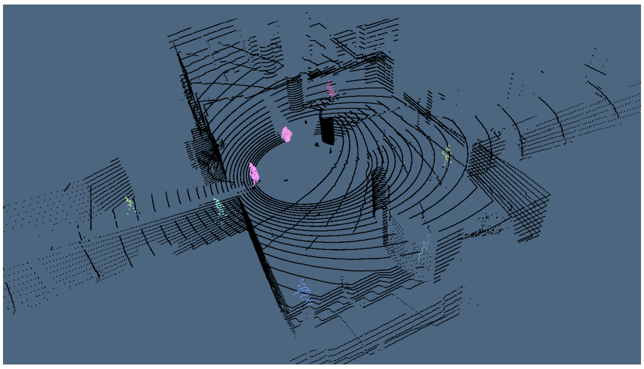


Fig. 9: Detected potential pedestrians (colored points sets)

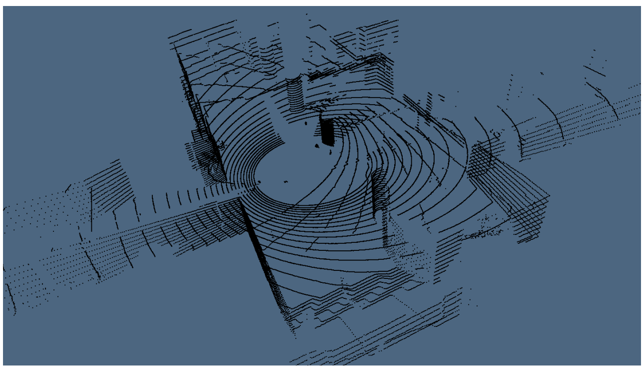


Fig. 10: Result of removing the detected pedestrians

## 5. Experiments

We evaluate the method on the 3D data set acquired with the mobile system for indoor mapping proposed in [10] in an underground shopping mall called Unimall near Nagoya station. As Unimall is near Nagoya station, it is a very bustling street and many people pass through it.

The data is acquired while a experimenter holds the Velodyne HDL-32E LiDAR sensor and walk along the underground street or rotate the inclined Velodyne HDL-32E as shown in Fig.11. Thus, the environment is dynamic and Velodyne HDL-32E is also moving.

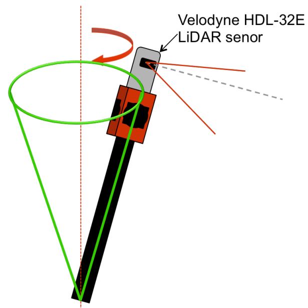
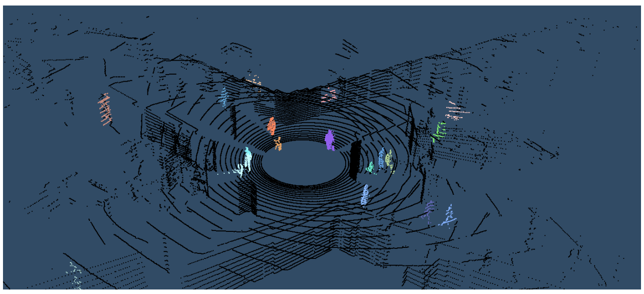
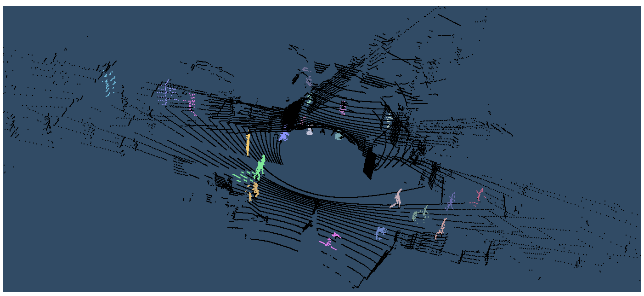



Fig. 11: Data acquisition by rotating the inclined Velodyne HDL-32E LiDAR senor

We apply the method proposed in this work to 50 frames point cloud of an intersection of the underground streets. 38 frames are acquired while Velodyne HDL-32E LiDAR sensor is moving straightly (Scenario A, Fig.12a) and 12 frames are acquired as Fig.11 (Scenario B, Fig.12b).



(a) Scenario A



(b) Scenario B

Fig. 12: Sample point cloud from ScenarioA and ScenarioB

The result is shown in Table.2. From Table.2 we can know in accuracy of Scenario B in lower than that of Scenario A. That is because pedestrians will be only scanned partly by an inclined Velodyne HDL-32E LiDAR sensor as shown in Fig.12b.

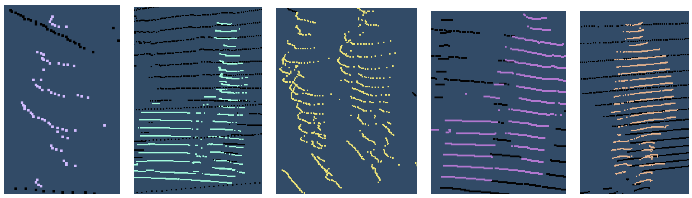
Most of failure TF cases are segments from scan lines of walls or pillars. And most of failure FP cases are the pedestrians those are occluded by other objects.

Table 2: Specification of popular LiDAR sensors

| | TP | FP | Precision | Recall |
|---|---|---|---|---|
| Scenario A | 460 | 154 | 0.723 | 0.874 |
| Scenario B | 207 | 79 | 0.693 | 0.86 |

As we intend to use this method to eliminate the affect of pedestrians' traces for 3D indoor mapping, it is more important for a high recall rate with a certain accuracy rate. Recall and accuracy can be adjusted by changing threshold values in Section 3.2 and Section 4.

Some segments of detected pedestrians in shown in Fig.13. Although it is a challenge that scanned line segments of a human change drastically if body pose changes or occlusion exists, from Fig.13 we can see that our method is robust to these aspects. As a comparison to Fig.2, we applied our algorithm to every frame that registered point cloud to eliminate traces of pedestrians. The final result is shown in Fig.14. We can see most traces of pedestrian are eliminated. This verifies the feasibility of our method for eliminating affect of pedestrians to generate quality mapping.



(a) (b) (c) (d) (e)

Fig. 13: Segments of detected pedestrians, (a)-Pedestrian far from HDL-32E b)-Pedestrian pushing a hand truck c)-Pedestrian walking closely d)-Pedestrian with a suitcase e)-Pedestrian carrying a briefcase

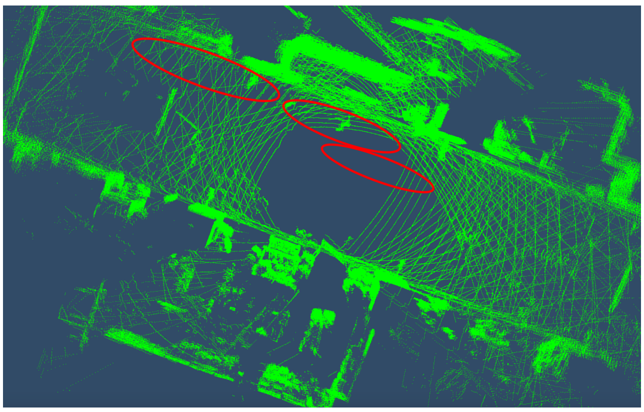


Fig. 14: A registered point cloud for indoor reconstruction and mapping after removing traces of pedestrians with the method we proposed

## 6. Conclusion and Future Work

In this work, we proposed a method that detects pedestrians for a dynamic crowded indoor environment in 3D directly without any other additional information. We achieved this by finding out the segment that possess specific features after clustering and agglomerating the point cloud data.

We evaluated our method on the experiment data that acquired in an underground shopping mall with a Velodyne HDL-32E LiDAR sensor. The experimental result of pedestrians detection verifies the feasibility and robustness of our method.

However, there are still a few traces can be seen in Fig.14. Also it is difficult to find optimum values of those thresholds for segmentation and detection. As we stated before, pedestrian detection could be considered as a classification problem. Thus, we consider train a classifier with the 3D data of various pedestrians with machine learning method to improve the recall rate and accuracy rate. And the method we proposed in this work could be applied for exacting 3D data of pedestrians as training data set for machine learning.

## Acknowledgement

A part of this research was supported by Strategic Information and Communication RD Promotion Program (SCOPE) 132306007.

### References

[1] Velodyne LiDAR, Inc., "USER'S MANUAL AND PROGRAMMING GUIDE, HDL-32E, High Definition LiDAR Sensor," 2012.

[2] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1*, pp. 886–893, 2005.

[3] R. Lienhart and J. Maydt, "An extended set of Haar-like features for rapid object detection," *Proceedings. International Conference on Image Processing*, vol. 1, pp. 0–3, 2002.

[4] U. Padova, F. Ingegneria, and P. Enrico, "People Detection and Tracking with Kinect for Mobile Platforms," 2011.

[5] L. Xia, C.-C. Chen, and J. Aggarwal, "Human detection using depth information by kinect," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2011 IEEE Computer Society Conference on*, pp. 15–22, June 2011.

[6] B. Choi, C. Mericli, J. Biswas, and M. Veloso, "Fast human detection for indoor mobile robots using depth images," August 2013.

[7] B. Kalyan, K. Lee, S. Wijesoma, D. Moratuwage, and N. Patrikalakis, "A random finite set based detection and tracking using 3D LIDAR in dynamic environments," *Systems Man and Cybernetics (SMC), 2010 IEEE International Conference on*, 2010.

[8] L. Spinello, K. Arras, R. Triebel, and R. Siegwart, "A Layered Approach to People Detection in 3D Range Data.," *Proc. of The AAAI Conference on Artificial Intelligence: Physically Grounded AI Track (AAAI)*, pp. 1625–1630, 2010.

[9] J. Shackleton, B. VanVoorst, and J. Hesch, "Tracking people with a 360-degree lidar," in *Advanced Video and Signal Based Surveillance (AVSS), 2010 Seventh IEEE International Conference on*, pp. 420–426, Aug 2010.

[10] W. Wang, K. Yamakawa, K. Hiroi, K. Kaji, and N. Kawaguchi, "A Mobile System for 3D Indoor Mapping Using LiDAR and Panoramic Camera," *IPSJ 46th UBI*, vol. 2015, no. 1, pp. 1–7, 2015.