Development of Real-time Environment Recognition System using LiDAR for Autonomous Driving[1]

Naruaki TOKUDOME, Shuhei AYUKAWA, Shun Ninomiya, Shuichi ENOKIDA, Takeshi NISHIDA

Kyushu Institute of Technology
Fukuoka, Japan
Email:nishida@cntl.kyutech.ac.jp

***Abstract*—Light detection and ranging (LiDAR) sensors are able to measure the shape, distance, and infrared refection intensity of objects within the range of 100 m, and the performance deterioration is small against the change of the natural environment. In this research, we developed a novel real-time environment-recognition system for autonomous driving using a LiDAR sensor. Moreover, we developed an improved method of long-distance recognition precision of LiDAR by superimposing the measurement data in time-series. In this paper, we describe the developed systems and methods. Furthermore, their effectiveness, such as in recognition precision and processing time as confirmed by several experiments using a car installed with the developed LiDAR system and employed methods, is described.**

Keywords—Autonomous Driving, Real-Time Measurement, LiDAR

I. INTRODUCTION

LiDAR is widely used on autonomous systems to detect information regarding the surrounding environment. Also, using LiDAR, collating with digital maps, and technology to recognize moving objects on the road has been eagerly studied. Pedestrian recognition, in particular, has been actively studied [1, 2]. For distant objects, recognition accuracy decreases with decreasing measurement points. It has been noted that challenges for safety of automatic driving arise at speeds of about 60 km/h, even if the LiDAR sensor is of the highest standard [3]. To address this limitation, a method to dramatically improve pedestrian recognition accuracy by superimposition of distance measurement data in time series by LiDAR has been proposed [4, 5].

We consider the requirement for a processing speed that can cope with a sudden jump-out of a pedestrian or movement of an oncoming vehicle, which are real-world requirements for the environment recognition systems relied on for autonomous driving. Thus, in this research, we assumed that the surrounding environment is sampled at 20 Hz, and we developed the environment system that realizes, in real-time, a method proposed in previous research [4, 5]. This system completes the process of data acquisition to person recognition within an average of 50 ms. The algorithm and configuration of this system are described below.

II. ENVIRONMENT RECOGNITION SYSTEM

A. Recognition Processing Flow

The processing flow of the environment recognition system using the measurement data of LiDAR is shown in Fig. 1. First, the algorithm completes ground removal, clustering, and rectangle fitting processing on point cloud data measured by LiDAR, thereby detecting objects around the vehicle. Next, the point cloud data presumed to measure pedestrians (pedestrian candidate cluster) is extracted from detected objects, and is judged by a classifier. A classifier identifies distant pedestrian candidate clusters after superimposing time-series point cloud data and increasing data. Each process is described in detail in the following sections.

B. Ground Removal

For object detection around the vehicle, it is necessary to remove from the measurement point cloud data any data identified as ground. In this research, we project point cloud data acquired by LiDAR onto a two-dimensional grid map, wherein the ground is removed based on the height information in each grid. In the experiment described in Section IV, the grid size of the grid map is set to 1.5 m and the height threshold is set to 0.3 m.

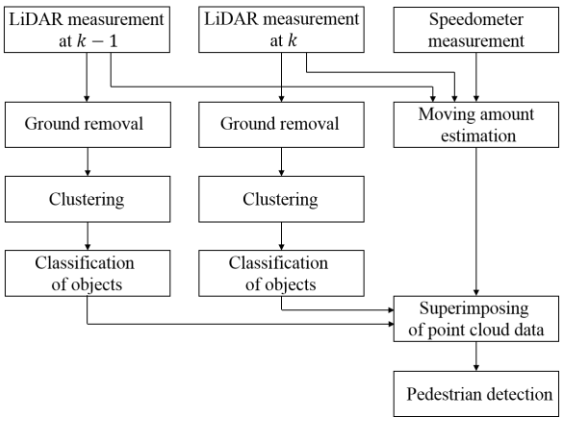


Fig. 1. Recognition processing flow of LiDAR data

[1]This content of this paper was summarized from the proceedings of the *33rd Fuzzy System Symposium* on Sept. 15, 2017 and translated from Japanese.

C. Clustering

Clustering of the measurement point cloud data is performed for data that are within a certain distance of the same object. In this research, Euclidean Clustering of PCL (Point Cloud Library) [6] was used for this process.

D. Rectangle Fitting and Classification by Rectangle Size

Next, rectangle fitting is performed on the created cluster. An axis-aligned bounding box (AABB) is used as a rectangle. After the rectangular fitting process, a classification of the cluster is defined per the size of the AABB. The lengths of the dimensions in the x-axis, y-axis, and z-axis of the AABB are represented as w, d, and h, respectively. The conditions for a cluster of measurements to be classified as a pedestrian candidate cluster are defined as:

$$\begin{cases} 0.05\ \text{m} \le w \le 1.2\ \text{m} \\ d \le 1.2\ \text{m} \\ 0.45\ \text{m} \le h \le 2.0\ \text{m} \end{cases} \quad (1)$$

The conditions for a cluster of measurements to be classified as a vehicle candidate cluster are defined as:

$$\begin{cases} 1.0\ \text{m} \le w \le 5.0\ \text{m} \\ d \le 4.0\ \text{m} \\ 1.0\ \text{m} \le h \le 2.5\ \text{m} \end{cases} \quad (2)$$

E. Pedestrian Recognition

To perform recognition processing on the pedestrian candidate cluster, we calculate the three-dimensional feature quantity. First, the pedestrian candidate cluster is represented as $C^{(i)} = \left\{ \boldsymbol{x}_k^{(i)} \middle| k = 1, \cdots, K \right\}$ where i is the index of the pedestrian candidate cluster and k is the index of the point cloud data included in the cluster. Next, we calculate the two feature vectors $\boldsymbol{f}_1^{(i)}$ and $\boldsymbol{f}_2^{(i)}$ from $C^{(i)}$ and generate $\boldsymbol{f}^{(i)} \triangleq \left[\boldsymbol{f}_1^{(i)} \quad \boldsymbol{f}_2^{(i)} \right]^T$ using them. The methods for calculating each feature vector are defined below.

a. Elements of the three-dimensional covariance matrix: $\boldsymbol{f}_1^{(i)} \in \mathbb{R}^6$

The dimensional covariance matrix of the point cloud data in the cluster is expressed by the following equation.

$$\boldsymbol{\Sigma}^{(i)} = \frac{1}{n-1} \sum_{\boldsymbol{x}_k^{(i)} \in C^{(i)}} \boldsymbol{x}_k^{(i)} \boldsymbol{x}_k^{(i)T} = \begin{bmatrix} f_{11}^{(i)} & f_{12}^{(i)} & f_{13}^{(i)} \\ f_{12}^{(i)} & f_{22}^{(i)} & f_{23}^{(i)} \\ f_{13}^{(i)} & f_{23}^{(i)} & f_{33}^{(i)} \end{bmatrix} \quad (3)$$

Here, $\boldsymbol{x}_k^{(i)} \triangleq \left[x_k^{(i)} \quad y_k^{(i)} \quad z_k^{(i)} \right]^T$ represents the coordinates of the three-dimensional point expressed in a reference frame with its origin at the center of the point cloud data. Because $\boldsymbol{\Sigma}^{(i)}$ is a symmetric matrix, we use six elements, without duplication of the mirrored off-diagonal matrix elements, as feature quantities and construct $\boldsymbol{f}_1^{(i)} \triangleq \left[f_{11}^{(i)} \ \cdots \ f_{16}^{(i)} \right]^T$.

b. Three-dimensional features calculated based on eigenvalues: $\boldsymbol{f}_2^{(i)} \in \mathbb{R}^7$

TABLE I. 3D FEATURES OF POINTS

$\boldsymbol{f}_{21}^{(i)}$	Linearity	$(\boldsymbol{\lambda_1 - \lambda_2})/\boldsymbol{\lambda_1}$
$f_{22}^{(i)}$	Planarity	$(\boldsymbol{\lambda_2 - \lambda_3})/\boldsymbol{\lambda_1}$
$f_{23}^{(i)}$	Scattering	λ_3/λ_1
$f_{24}^{(i)}$	Omnivariance	$\sqrt[3]{\lambda_1 \lambda_2 \lambda_3}$
$f_{25}^{(i)}$	Anisotropy	$(\lambda_1 - \lambda_3)/\lambda_1$
$f_{26}^{(i)}$	Eigenentropy	$-\Sigma_{i=1}^{3} \lambda_i \ln \lambda_i$
$f_{27}^{(i)}$	Change in curvature	$\lambda_3/(\lambda_1 + \lambda_2 + \lambda_3)$

The eigenvalues of $\boldsymbol{\Sigma}^{(i)}$ are represented as $\lambda_1, \lambda_2, \lambda_3$ where $\lambda_1 > \lambda_2 > \lambda_3 > 0$. From these eigenvalues, we calculate the seven values (See TABLE I.) proposed by Liu at al. [7], and construct $\boldsymbol{f}_2^{(i)} \triangleq \left[f_{21}^{(i)} \ \cdots \ f_{27}^{(i)} \right]^T$ from them.

Using Support Vector Machine (SVM) as classifier, the learning data of SVM was prepared from measurement data. The point cloud measurement data resulted in both correct and incorrect answer data. Pedestrians were correctly identified as answer data, but the point cloud data incorrectly identified objects other than pedestrians, such as trees and streetlights. The number of data used for learning was 1800 for correct answers and 2780 for incorrect answers.

F. Moving Amount Estimation of the Vehicle

Because measurements from the vehicle speedometer contain errors, sensor data from LiDAR is included to estimate the amount of movement of the vehicle. The movement amount is obtained by estimating the rotation matrix $\boldsymbol{R}_k$ and the translation vector $\boldsymbol{t}_k$ that associate each measurement. At time $k-1$ and time k, $\boldsymbol{R}_k$ and $\boldsymbol{t}_k$ are estimated by collating $P_{k-1} = \{\boldsymbol{q}_1, \boldsymbol{q}_2, \cdots, \boldsymbol{q}_M\}$ and $P_k = \{\boldsymbol{p}_1, \boldsymbol{p}_2, \cdots, \boldsymbol{p}_M\}$ point cloud data measured from LiDAR. In this research, normal distribution transform (NDT) [8] was used for matching point cloud data.

G. Superimposing of Point Cloud Data

In this research, to identify pedestrians at far distances, where the number of lasers irradiated from LiDAR decreases, a method of superimposing measurement data in time series [4, 5] was adopted. An overview of time-series data superimposition is shown in Fig. 2. To perform the superimposing of point cloud data of the pedestrian, it is necessary to search for the correspondence between the pedestrian candidate cluster extracted at time $k-1$ and the pedestrian candidate cluster extracted at time k. In this method, the location of the center of gravity and the size of each side of the AABB were used for identity classification. The fluctuation of the LiDAR measurements is not constant; the number of laser spots on the same pedestrian changes for each frame. Therefore, the size of AABB in each frames changes. Thus, allowable errors are set to compare voxels as follows:

$$h_{k-1}^{(j)}(1-\alpha_1) \le h_k^{(j)} \le h_{k-1}^{(j)}(1+\alpha_2) \quad (4)$$

$$w_{k-1}^{(j)}(1-\beta_1) \le w_k^{(j)} \le w_{k-1}^{(j)}(1+\beta_2) \quad (5)$$

where j is the index of AABB of the pedestrian candidate cluster, $\alpha_1 = 0.3$, $\alpha_2 = 0.4$, $\beta_1 = 0.55$ and $\beta_2 = 1.2$. These parameters were defined through trial and error.

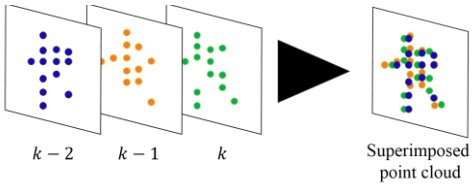


Fig. 2. An overview of time-series data superimposition

By assessing the range of pedestrian movement between samples, the allowable distance is also included in the error parameters. Voxels determined to represent the same pedestrian are superimposed by coordinate transformation to align their centroids. Next, the determination of the identity of the voxels in consecutive frames is defined by:

$$\left\| \boldsymbol{x}_{k-1,c} - \boldsymbol{x}_{k,c} - \boldsymbol{l}_k \right\| \leq r \qquad (6)$$

where $\boldsymbol{l}_k \triangleq [l_{k,x} \quad l_{k,y}]^T$ is the estimated vehicle movement, $\boldsymbol{x}_{k,c} \triangleq [x_{k,c} \quad y_{k,c}]^T$ is the center of gravity coordinates of a AABB at k, and r is the range of movement of the pedestrian and the measurement error. In the experiment described in Section IV, r is defined as 1.0 m. When AABBs satisfy Eqs. (4), (5) and (6), the two clusters are regarded as identical.

III. Environment Recognition System

The function defined in the previous section was implemented on a car as a real-time processing system. The required specifications of the system are: (i) LiDAR and vehicle speedometer measurement data must be time-stamped,, as simultaneous measurements are required by various processes, (ii) processing time from the acquisition of measured data to object recognition around the vehicle must be less than 50 ms, and (iii) the processing time of movement amount estimation for superimposing time-series data must be within 50 ms. Further, in order to realize real-time processing, we rely on a distributed processing system using numerous computer processors. An overview of the distributed processing system is shown in Fig. 3. A schematic of processing sharing of each computer is shown in Fig. 4.

IV. Experimental Details

A. Configuration of the Experimental Vehicle

In this research, a LiDAR sensor, HDL-64E S2 (Velodyne) was used. HDL-64E is able to measure the horizontal omnidirectional; however, in this experiment, it was used to focus on the measurement range of forward $\pm 30°$. Computers for data processing and network connection device were stored in the trunk of the experimental vehicle.

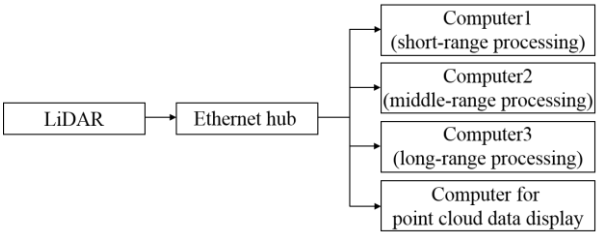


Fig. 3. An overview of the constructed processing system

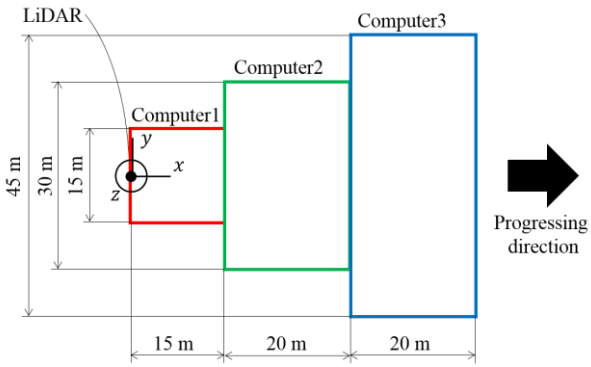


Fig. 4. Outline of the range for which each computer is responsible. For the movement amount estimation, the measurement data for the range of computer1 is used. Pedestrian recognition in the area handled by computer3 performs superimposition processing.

B. Experimental Overview

To verify recognition accuracy for distant pedestrians using the superimposing process, we placed a person approximately 50 m ahead from the experimental vehicle and drove the experimental vehicle towards the person at a speed of 20 km/h. Meanwhile, a person crossed the path in front of the experimental vehicle. Finally, we installed the developed environment recognition system on the experimental vehicle, drove on campus at approximately 15 km/h, and measured the processing time of each process.

V. Experimental Results

The results of the sharing of processing areas by the three computers are shown in Fig. 5. In addition, the progress of each process for pedestrian recognition is shown in Fig. 6.

A. Recognition Experiment of Pedestrian

The results of the recognition experiment for the pedestrian crossing the front of the vehicle (distance: 45–50 m) are presented in TABLE II. In this experiment, the number of superimposing frames f_n was increased from 0 to 4, and each recognition experiment was conducted 60 times. As summarized in TABLE II it was confirmed that superimposition processing is effective for recognizing distant pedestrians.

TABLE II. Accuracy of Pedestrian Recognition with Respect to the Number f_n of Superimposed Frames

f_n	0	1	2	3	4
Recognition rate [%]	18.8	49.2	68.9	77.0	67.2

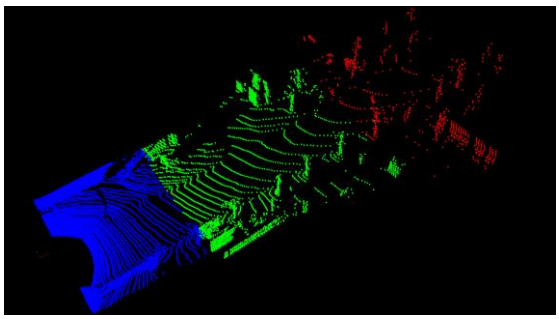
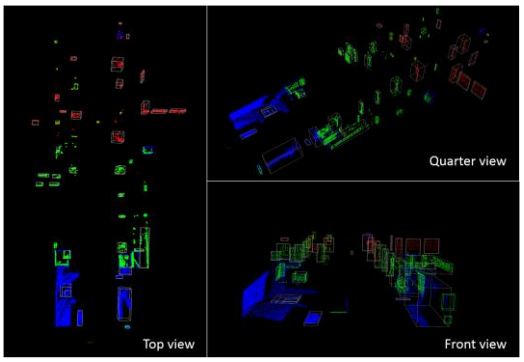
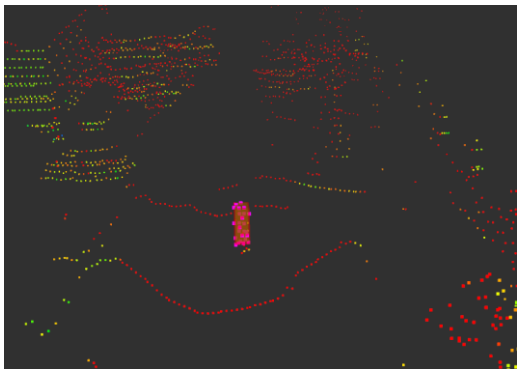


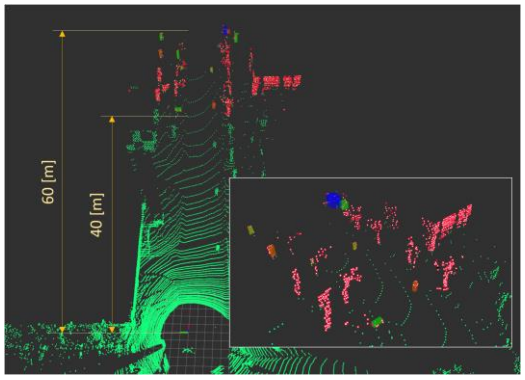
Fig. 5. Data acqusition from LiDAR and processing sharing



(a) Ground removal, clustering and rectangle fitting



(b) Pedestrian recognition



(c) Superimposition processing

Fig. 6. Progress of each process for pedestrian recognition

TABLE III. RESULT OF VERIFICATION OF PROCESSING TIME

Processing type	Average processing time [ms]	Maximum processing time [ms]
Short-range processing	55.71	158.2
Middle-range processing	36.77	71.68
Long-range processing	12.12	27.74
Moving amount estimation	37.96	60.86

B. Verification of Processing Time

The resulting processing times are shown in Table III. The movement amount estimation processing, including down sampling, was executed within 50 ms on average. In rare cases, processing time exceeded 50 ms. Short-range processing was not completed within this research's goal of 50 ms. One suspected cause is that there are many measurement data in this area. In middle-range processing, processing time sometimes exceeded 50 ms. In long-range processing, processing cloud was completed within 50 ms. Considering each processing case, the largest time was consumed for ground removal processing and clustering processing in short-range and medium-range processing. The total of these processing times accounts for about 74% (short-range) and 61% (medium-range) of the total processing time. These experimental results suggest that it is necessary to expedite these processes, further divide the responsible distance ranges, and/or increase the number of parallel processors.

VI. CONCLUSION

In this research, we developed a real-time environment-recognition system using LiDAR for automatic operation. In addition, we improved the recognition accuracy of the distant pedestrian by using the superimposition process. We increased the processing speed by performing distributed processing on multiple computers. We confirmed that recognition accuracy improves from 18.8% to 77.0% by superimposing measured data for distant pedestrians. However, it was confirmed that recognition processing could not be executed within 50 ms for each sample. Improvements to be implemented in forward work include reducing the processing times, likely to be accomplished by efficiency improvements such as further subdividing the designated distance ranges to be handled by each processor, and increasing the number of parallel processing processes.

REFERENCES

[1] S. Enokida, T. Kamishina, and M. Jono, "Pedestrian Detection Based on The Shape Feature and The Reflection Intensity Distribution Using LiDAR," Journal of Fuzzy System Symposium of Japan, vol. 32, pp. 141–146, 2016.

[2] K. Kidono, A. Watanabe, T. Naito, J. Miura, "Pedestrian Recognition Using High-definition LiDAR," Journal of Robotic Society of Japan, vol. 29, No. 10, pp. 963–970, 2011.

[3] T. Nishida, S. Enokida, S. Ayukawa, N. Tokoudome, "Investigation of Specification of LiDAR for Autonomous Driving," Journal of Fuzzy System Symposium of Japan, vol. 32, pp. 147–150, 2016.

[4] S. Ayukawa, N. Tokudome, S. Enokida, T. Nishida, "Time-series LiDAR Data Superimposition for Autonomous Driving," Proc. of ICT-ROBOT, ThBT3.3, Pusan, Sep. 8, 2016.

[5] N. Tokudome, S. Ayukawa, T. Nishida, "Pedestrian Recognition Using The Superimposition of LiDAR Data," technical paper of the SICE Kyushu branch annual meeting, pp. 23–26, 2016.

[6] "Point Cloud Library," http://pointclouds.org/ (accessed 2017–09–06).

[7] K. Liu, et al: "Classification of Big Point Cloud Data Using Cloud Computing," ISPRS, XL-3/W, pp. 553–557, 2015.

[8] M. Martin, et al: "Scan Registration for Autonomous Mining Vehicle Using 3D-NDT," Journal of Field Robotics, Vol. 24, pp. 803–827, 2007.