Proceedings of the SICE Annual Conference 2021
September 8-10, 2021, Tokyo, Japan

3D PCD map creation for LiDAR localization using a spherical camera

Yusuke Isozumi[1†] Satoshi Ito[2,3] Ryosuke Morita[2,3], Yuki Funabora[4]

[1]Graduate school of Natural Science and Technology, Gifu University,
Tokai National Higher Education and Research System, Gifu, Japan
(Tel: +81-58-293-2540; E-mail: a4525007@edu.gifu-u.ac.jp)
[2]Faculty of Mechanical Engineering, Gifu University,
Tokai National Higher Education and Research System, Gifu, Japan
[3]Intelligent Production Technology Research & Development Center for Aerospace (IPTeCA)
Tokai National Higher Education and Research System, Nagoya, Japan
[4]Graduate school of Engineering, Nagoya University,
Tokai National Higher Education and Research System, Nagoya, Japan

Abstract: This paper proposes a method for creating a 3D PCD map using the spherical camera to utilize it for the LiDAR self-localization of the AMR. Usually, the LiDAR on the AMR is utilized for the map creation, but it requires some process with moving the big AMR itself. This paper attempts to simplify this process by replacing the scanning device with a small spherical camera. Finally, a map creating method is established using OpenVSLAM, and automated driving is demonstrated based on the map obtained from this method.

Keywords: Autonomous Mobile Robot (AMR), localization, Light Detection, and Ranging (LiDAR), Point Cloud Data Map (PCD Map), spherical camera, map creation

1. INTRODUCTION

Manufacturing industries such as the automobile industry and the aircraft industry are thriving in the Central region of Japan including our prefecture, Aichi and Gifu. In particular, the aerospace industry in Gifu Prefecture is the second largest in Japan in terms of the number of business establishments and employees and boasts a high concentration so that it occupies the third-largest in the shipment value of manufactured goods. Taking advantage of its regional characteristics, Gifu area has been designated as one of the governmental projects named, 'Human resource development and research project on production technology aiming to form Japan's aerospace industry cluster'.

One of the aims of this project is the manufacturing innovation of the aerospace industry towards the cyber-physical factory, in which automatization is one of the key technology realizing the cost-saving high-performance with accuracy and rapidness. Among the automatization, we are tackling the transportation achieving the unmanned delivery between or within the factories.

To reflect the characteristics of factories in the aircraft industry, we are proceeding to develop the following functions:

1. **Supports frequent changes of factory layout.**
   Since the products, namely, aircrafts are large, their position is rather fixed than flowed on the line: parts and tools are often transported and then temporally installed for manufacturing. This implies that the position of the manufacturing machines has to be modified to cope with the frequent changes in the production site layout.
2. **Supports a wide variety of transportation routes.**
   The large aircraft factory contains many buildings treating numerous numbers of parts and assemblies. Each part has a different start point and endpoint for transportation, and even the same part is sometimes used in a different place, Also, the destinations of manufactured parts are also diverse. These facts indicate that the transport route is always changed frequently and various transportation routes are required.
3. **Supports the co-existence of both human and vehicles**
   The transportation route in the factory has to be often set to pass the area where the human workers or other vehicles are operating or moving. In particular, automobiles and bicycles are passing the outside of the building. Therefore, it is necessary to avoid not only stationary obstacles but also moving objects.

To meet these three requirements, our research group containing companies had designed an original AMR (Automatic Mobile Robot) prototype machine. Then, the most crucial and basic function was how to acquire the environmental status of AMR and how to grasp the current position based on it, that is, the method of self-localization. To realize the self-localization, we decided to install, at the prototype machine, the 3D LiDAR (Light Detection And Ranging) which is also applied in the automatic driving technology of automobiles.

However, the self-localization using LiDAR requires an environmental map in advance before the AMR works there. This map is called 3D PCD map (Point Cloud Data Map) composing of a tremendous number of 3D positional data. At the localization, the LiDAR-scanning data is compared to this map and the best matching point is selected as the estimate of the current position.

† Yusuke Isozumi is the presenter of this paper.

This PCD map is usually created by the AMR itself to be used because it already has an expensive LiDAR. Furthermore, the data from the AMR to be used are preferable for the matching; the data scanned by another AMR produces a bit different 3D PCD map due to the effect of the LiDAR height, which is not advantageous for the accurate self-localization.

However, the following problems were pointed out from the shop-floor side against this normal map-creating method.

1. Factory layout frequent changes and the map should be updated at the same time. That is to say, the situation of the updating map is too frequent.
2. The prototype machine will be operated remotely, but the remote operation requires some skills and familiarization.
3. The education of the shop-floor workers is needed to master a series of map creation processes.

Based on these problems, this paper aims at proposing a creation method of the 3D map that does not use LiDAR or the prototype AMR itself. Instead of 3D LiDAR on the AMR, we are going to introduce an inexpensive commercialized camera for the map creation. The envisioned situation is that a human walks around the factory with a camera attached to the top of his or her helmet or something similar. While walking around, the camera records the environment as a video. This video is analyzed to calculate the location information through the image processing. Based on this analysis, the 3D map is generated. In Section 2, we establish a method as a series of operations to obtain a PCD map. In Section 3, we evaluate the accuracy of this map and its applicability to AMR self-localization using LiDAR through experiments. Section 4 concludes this paper.

2. PCD MAP CREATION METHOD USING A SPHERICAL CAMERA

2.1. Overview

In this research, we use a spherical camera to capture a wide 360-degree area at once. Based on the spherical camera images, we will create a 3D map using OpenVSLAM. OpenVSLAM can detect such feature points as the places with large color changes or corners of a window frame from the spherical camera video, and output their position as a PCD map. However, point clouds created from camera images are inherently different in their number and location compared to those that LiDAR produces. Therefore, trial-and-error experiments were required to obtain a PCD map that is applicable to the LiDAR self-localization.

In this section, we will establish a procedure for creating a PCD map from a spherical camera by testing various preparations, operations, situations, and parameters to be set.

2.2. Proposed method

The following is a series of procedures we propose.

1. Camera calibration
2. OpenVSLAM parameter settings
3. Feature point extraction by OpenVSLAM
4. Map creation
5. Scale conversion

The rest of this subsection explains these processes in this order.

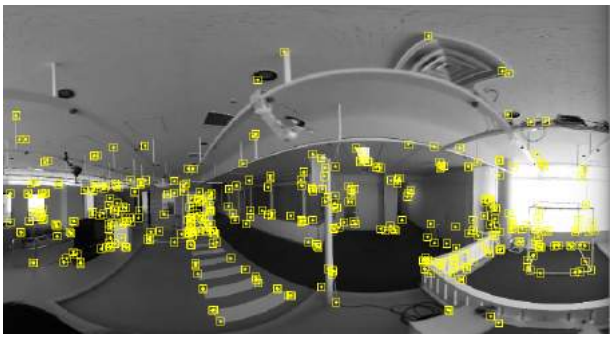

Fig. 1 Feature point extraction on OpenVSLAM

2.2.1. Camera calibration

Camera calibration is a method of estimating the internal parameters of a camera from the known-size photos and videos taken by the camera itself. It is also possible to correct lens distortion from the estimated parameters. Several checkerboard photos were taken and used to estimate the internal parameters.

2.2.2. Setting parameters for OpenVSLAM

OpenVSLAM requires some parameters of the camera to extract the feature points. The important parameters are summarized as follows:

Feature.max_num_keypoints : Maximum number of feature points in one video frame.

Feature.scale_factor: This parameter is used to estimate the 3D distance of an object from the superimposing photos of different scales internally created by the OpenVSLAM.It determines the distance between photos of different scales.

Feature.num_levels: It determines the number of photos of different scales.

Feature.ini_fast_threshold,Feature.min_fast_threshold :Brightness threshold of one video frame.

These values are set in the file, camera_config.yaml.

Among them, some parameters such as camera fps and resolution are determined by the camera specification. Other parameters were determined by trial and error.

2.2.3. Feature point extraction using OpenVSLAM

OpenVSLAM automatically extracts feature points such as the corners of a window frame or ends of a pillar, and computes their coordinates in 3D space.

Fig. 1 shows how the feature points are extracted on OpenVSLAM. Many small yellow rectangles denote the extracted feature points.

2.2.4. Map creation

The feature points extracted by OpenVSLAM were converted to PCD files. For the conversion, a python

Table 1 The specification of the spherical camera utilized in this paper.

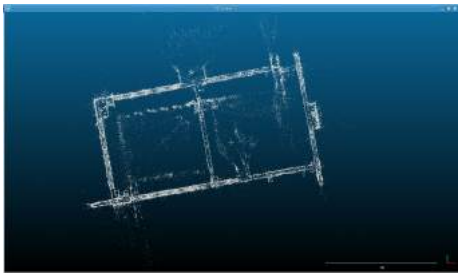

Fig. 2 Point cloud data map

code is provided on the Internet[1]. The PCD file can be checked by the PCD editing software 'CloudCompare'[2]. Fig. 2 shows a visualization result of the PCD file using CloudCompare.

2.2.5. Scale conversion

The scale of the map that OpenVSLAM produced is often not the same as that of the real space. In such a case, the scale should be manually adjusted using CloudCompare.

3. EXPERIMENTS

3.1. Purposes and methods

Using the map followed by the procedure in section 2, we aim at realizing the self-localization and the automatic driving by the AMR. Then, an automatic driving software, "Autoware (ver. 1.1.12.0)" was utilized. Autoware uses a technique called scan matching to achieve the self-localization from the created PCD map and LiDAR-scanning data[3]. We conducted experiments in the following three steps.

1. Check first if Autoware can read the map created by the procedure in section 2.
2. Confirm next whether the map the Autoware has read is available to the self-localization of Autoware.
3. Evaluate the accuracy of self-localization between the map created from the proposed method and the one generated by the LiDAR (normal method).

3.2. Experimental environment and preliminary experiments

The spherical camera, Insta360 ONE X2 (Insta360), was used in all experiments. Its specifications are summarized in Table 2.

Photos of the LiDAR and AMR used in the experiments are shown in Fig. 3 and Fig. 4 respectively. The specifications of the LiDAR and the AMR are listed up in Table 3 and Table 4, respectively.

As the result of the camera calibration, we obtained the parameter values in Table 5. These parameter values are f_x and f_y for the focal length, c_x and c_y for the optical center, and k_1, k_2, k_3, and k_4 for the distortion of the captured image.

Fig. 3 LiDAR [Velodyne VLP-16]

Table 2 Specification of spherical camera

Product name	Insta360 ONE X2
Manufacturer name	Insta360
Display size	diameter 2.54 cm
Resolution	5.7 K
Focal length	7.2 mm
Pixel count	18 MP
Maximum ISO	3200
Maximum shutter speed	1/8000

Table 3 Specification of 3D LiDAR, Velodyne VLP-16

Maximal measurement distance	100 m
Horizontal field of view angle	360°
Vertical View Angle	30°
Number of measurement points	300000 points per second
Rotation speed	5–20 Hz
Accuracy	±3 cm

Fig. 4 Autonomous Mobile Robot (AMR) utilized in the experiments.

As a preliminary experiment, we recorded a video with walking along the corridors in the building of the Faculty of Engineering at Gifu University, which is approximately 40 m x 80 m in size. From this video, we created a point cloud using OpenVSLAM. Fig. 5 shows a PCD map generated from the extracted feature points in the spherical camera images.

Table 4 Specification of AMR

Item	Performance
Weight of the desired load	50 kg
Body weight	70 kg
Charge time	5 hour
Maximum speed	6 km/h
Overcomeable steps	5 cm
Climbing ability	10°
Mileage	16 km
Minimum turning radius	760 mm

Table 5 Camera parameters of Insta360 ONE X2

Internal parameter	
f_x	1596.963
f_y	1789.927
c_x	2275.826
c_y	177.4559
k_1	0.093676
k_2	-0.0384
k_3	-0.12637
k_4	0.00026

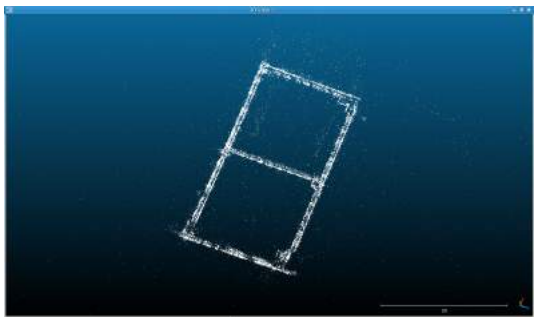

Fig. 5 PCD map from Insta360 ONE X2

3.3. Experiment 1: Availability of the created PCD map

The purpose of this experiment is to confirm whether the 3D PCD map created from the procedure in Section 2 is available for the LiDAR self-localization. The conditions for generating point clouds are very different between LiDAR scanned images and camera images. LiDAR can generate point clouds anywhere light is reflected, while the point clouds from the camera are limited to areas with specific characteristics, such as extremely bright areas within the image or corners of a window frame. So, as a preliminary experiment, we check that a 3D PCD map from the camera images is certainly available for the LiDAR self-localization.

As expected, Autoware encountered an error; it could not even read the map generated by OpenVSLAM: needless to say, it was far from utilizing for the LiDAR self-localization either.

We here inferred that one of the crucial reasons exists in the low density of the generated 3D PCD map. Therefore, we recorded the building several times to increase the number of point clouds. Then, to shorten the recording time, we reduced the recorded area to half.

Table 6 Parameter of camera_config.yaml and the number of points.

	Paremeter value			
Parameter name	1	2	3	4
feature.max_num_keypoints	2000	10000	10000	15000
feature_scale_factor	1.2	1.5	1.5	1.5
feature_num_levels	8	6	5	5
feature_ini_fast_threshold	20	20	20	20
feature_min_fast_threshold	4	4	7	7
Total number of point clouds	30911	51871	61580	72260
Matching	NG	NG	OK	NG

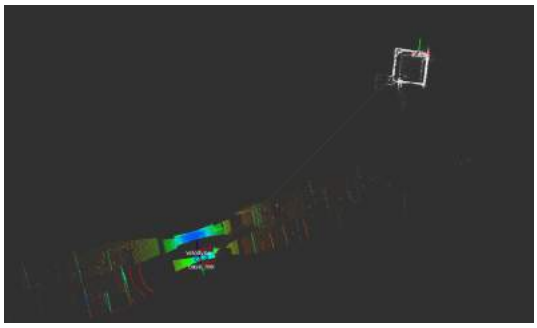

Fig. 6 Matching between PCD map and scan data

To see the increasing effect of the recording number, we changed the values of the parameters in camera_config.yaml with fixing the number of laps to record at three, and compared the total number of point clouds in the created map. The results are shown in the table 6.

The total number of point clouds increases with the maximum value of feature points that can be extracted in one video frame.

To evaluate what number of point clouds is sufficient for self-localization, we attempted to estimate the self-localization by LiDAR based on the resulting map.

Fig. 6 shows the matching between the created map and the scan data. The possibility of self-localization is summarized in the last row of Table 6, implying that the matching fails even when the total number of point clouds is large. Table 7 shows the matching result for the third map in Table 6 that was the only case where the positive result was obtained.

In short, although the total number of point clouds increased, the self-localization were sometimes failed. This result brought us the following idea: the corridor may have been unsuitable for feature point extraction due to the similar camera images from its uniform background.

3.4. Experiment 2 : Creating a map at a preferable location for feature point extraction

3.4.1. Experimental environment

Experiment 1 indicated that the preferable location for the feature point extraction from the camera images are:

1. an area including many feature point candidates such as wall borders and window frames.
2. a compact area that enables us to record in many times, repeatedly.

Table 7 Parameter of camera_config.yaml and the number of points.

	Parameter value							
Parameter name	1	2	3	4	5	6	7	8
feature.max_num_keypoints	20000	30000	30000	20000	20000	30000	30000	20000
feature_scale_factor	1.4	1.5	1.5	1.2	1.4	1.45	1.3	1.4
feature_num_levels	8	7	8	7	8	8	8	8
feature_ini_fast_threshold	20	20	20	20	20	20	20	20
feature_min_fast_threshold	7	7	7	7	7	7	7	7
Total number of point clouds	11036	10984	8662	16802	20712	18890	22930	20390
Matching	OK	OK	NG	NG	OK	OK	OK	OK

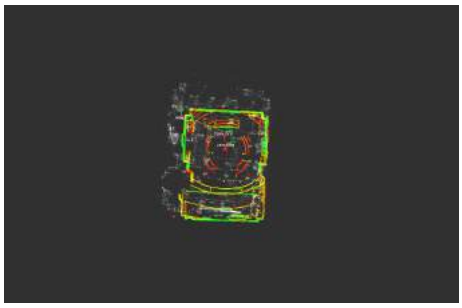

Fig. 7 Comparison of the PCD map with LiDAR scan data

Thus, the next experiment was conducted in such a location, a student laboratory in the Faculty of Engineering Building, Gifu University

In this experiment, we recorded the video in 10 laps.

As is the same as in Experiment 1, the number of point clouds and the availability of position estimation were investigated for the maps created from the different parameter values in camera_config.yaml. Table 7 shows the values of the parameters and the total number of point clouds on the map. The matching results to the LiDAR scan data were also described.

Fig. 7 shows a part of the overlaid scan data on the successfully matched map.

Extract feature points, we noticed that the total number of point clouds and the scale of the map were different between No. 5 and No. 8 of Table 7. Therefore, we transformed the map scale to fit the actual size of the room.

Fig. 8 shows the scale-adjusted map, and Fig. 9 shows the self-localization using the scaled 3D map and LiDAR scan data on Autoware.

As we can see, the 3D PCD map and the scan data from the LiDAR are overlapped well. Because the LiDAR scan data and the 3D map were matched well even at different positions, we expected that this map would be available for self-localization. The simulation based on this map and the LiDAR scan data demonstrated automatic driving as shown in Fig. 10.

From the above, the map created from OpenVSLAM needs to be adjusted in scale, and if this rescaling is done properly, the resulting map works well for self-localization using LiDAR.

3.5. Experiment 3: Evaluating the Accuracy of Self-localization

In the last experiment, we compare the accuracy of self-localization using Autoware between two maps, one created with a spherical camera (proposed method) and

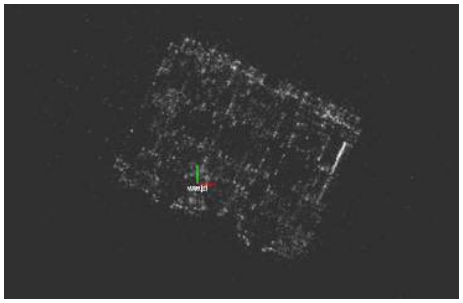

Fig. 8 Scaled PCD map.

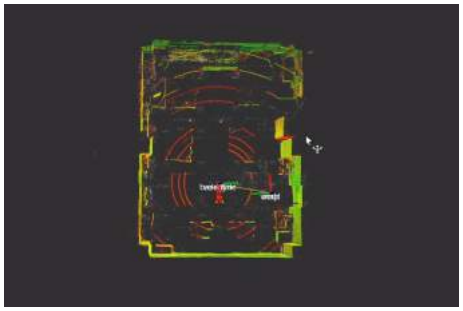

Fig. 9 Succeeded self-localization of AMR

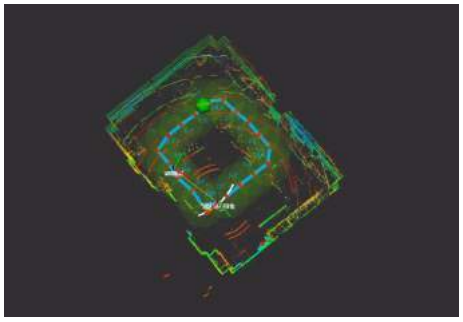

Fig. 10 Autonomous driving of AMR

the other created with LiDAR (conventional method).

We here evaluated how much the result of self-localization against the target route differs depending on the map utilized during automatic driving. The experiment is conducted in the same student laboratory as in the previous section.

At first, the AMR is manually moved along the path set for the automatic driving, as shown in Fig. 11. Namely, this data is used as the target route for automatic driving. Autoware generates the actual reference trajectory to smoothly connected to the points in this target route. The reference trajectory becomes the same even if the map for self-localization is different.

Table 8 describes one result of automatic driving comparing the positions between the target path and the actual point in the experiment using two kinds of maps.

Fig. 12 is a graphical representation of the result of Table 8.

In THIS experiment, the map created by OpenVSLAM achieved the same accuracy as the LiDAR map.

Fig. 11 Path in the experiments.

Table 8 Automatic driving results

Travel path		LiDAR map		OpenVSLAM map	
x[m]	y[m]	x[m]	y[m]	x[m]	y[m]
0.3943	0.1485	0.5891	-0.1584	0.2165	-0.2119
1.394	0.2582	1.3936	0.4696	1.1922	0.1768
2.3951	0.3221	2.3848	0.2876	2.1969	0.2998
3.4129	0.3425	3.4572	0.4342	3.1947	0.2202
4.2284	0.9459	4.2623	1.0307	4.1011	0.8491
4.1829	1.9603	4.2499	2.1183	4.2991	1.84
4.032	2.9825	3.7142	2.9646	4.1492	2.9124
3.0361	2.7652	2.6034	2.8825	3.1261	2.9886
1.9961	2.7434	1.6891	2.8325	2.0668	2.9926
0.9648	2.7575				
0.557	1.8184				
0.3318	0.8042				

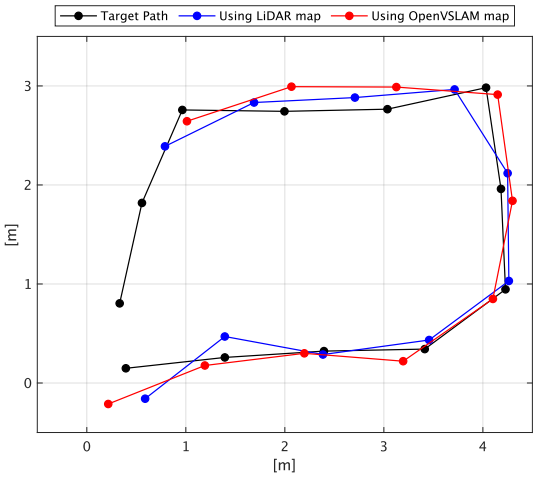

Fig. 12 Comparing Autonomous driving

4. CONCLUSION

In this study, we introduced a spherical camera to create a 3D map of the environment for AMR self-localization instead of the 3D LiDAR. OpenVSLAM extracts feature points such as wall boundaries and window frames from the video of the environment captured by the spherical camera. The extracted feature points were then converted into a 3D PCD map. Unfortunately, this 3D map is not always created on the same scale as the actual environment. Therefore, the scaling of the map is required at this stage using the application software. This scale fitting is an important process to utilize the OpenVSLAM map for the LiDAR self-localization. Through several experiments, we have confirmed that the scaled maps have sufficient information for self-localization and have finally achieved automated driving.

However, this success is limited to cases with preferable conditions, such as narrow areas where many edges can be detected easily. In the future, we will tackle clarifying the conditions and methods to increase the number of feature points that can be extracted by OpenVSLAM, and to extend the method so that it can be applied to cases with unfavorable conditions.

ACKNOWLEDGE

This work is supported by Human resource development and research project on production technology for aerospace industry: Subsidy from Gifu Prefecture.

REFERENCES

[1] Accessing or extracting point cloud data directly or via a written database file, https://github.com/xdspacelab/openvslam/issues/

[2] CloudCompare 3D point cloud and mesh processing software Open Source Project, http://cloudcompare.org/

[3] T. Azumi, D. Hukutomi, S. Tokunaga, S. Seiya “Autoware Introduction to Automated Driving Software”, Rick Telecom Inc., 2019

[4] Conducted a study on AGV/transfer robot market (in Japanese), https://www.yano.co.jp/press-release/show/press_id/2507, 2020

[5] Collaborative Robot BLOG So much has changed! The New Common Sense of Premise Transport No.2: Difference between AGV and AMR (in Japanese), https://www.kyodo-robot.com/blog_amr/202004-amr2

[6] K. Sakurada, “OpenVSLAM: A Versatile Visual-SLAM Framework”, Proceedings of the 27th ACM International Conference on Multimedia 2019 Pages 2292-2295

[7] A. Hosaka, K. Aoki, S. Tsugawa, “Automatic driving System configuration and elemental technologies”, Morikita Publishing Co., 2015

[8] S. Kato, S. Tokunaga, Y. Maruyama, S. Maeda, M. Hirabayashi, Y. Kitsukawa, A. Monrroy, T. Ando, Y. Fujii, and T. Azumi,“Autoware on Board: Enabling Autonomous Vehicles with Embedded Systems,” In Proceedings of the 9th ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS2018), pp. 287-296, 2018.

[9] S. Kato, E. Takeuchi, Y. Ishiguro, Y. Ninomiya, K. Takeda, and T. Hamada. “An Open Approach to Autonomous Vehicles,” IEEE Micro, Vol. 35, No. 6, pp. 60-69, 2015.