IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 3, 2020, accepted March 12, 2020, date of publication March 23, 2020, date of current version April 17, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2982681

Object Recognition Based Interpolation With 3D LIDAR and Vision for Autonomous Driving of an Intelligent Vehicle

IHN-SIK WEON[1], SOON-GEUL LEE[2], AND JAE-KWAN RYU[1]
[1]Unmanned Systems, LIG Nex1 Co., Ltd., Seongnam 13488, South Korea
[2]Department of Mechanical Engineering, Kyung Hee University, Yongin 17104, South Korea

Corresponding author: Soon-Geul Lee (sglee@khu.ac.kr)

This work was supported in part by the National Research Foundation of Korea, funded by the Ministry of Education under Grant 2019R1A2C2010195, in part by the LIG Nex1 Co., Ltd., through the Development of Multi-Platform Interlocking Control Model and Control Algorithm Program of the Agency for Defense Development (ADD) of Korea under Grant 20192610, and in part by the Ministry of Science and ICT of Korea, through the ICT Consilience Creative Program supervised by the Institute for Information & communications Technology Planning & Evaluation (IITP) under Grant IITP-2019–2015–0-00742.

ABSTRACT An algorithm has been developed for fusing 3D LIDAR (Light Detection and Ranging) systems that receive objects detected in deep learning-based image sensors and object data in the form of 3D point clouds. 3D LIDAR represents 3D point data in a planar rectangular coordinate system with a 360° representation of the detected object surface, including the front face. However, only the direction and distance data of the object can be obtained, and point cloud data cannot be used to create a specific definition of the object. Therefore, only the movement of the point cloud data can be tracked using probability and classification algorithms based on image processing. To overcome this limitation, the study matches 3D LIDAR data with 2D image data through the fusion of hybrid level multi-sensors. First, because 3D LIDAR data represents all objects in the sensor's detection range as dots, all unnecessary data, including ground data, is filtered out. The 3D Random Sample Consensus (RANSAC) algorithm enables the extraction of ground data perpendicular to the reference estimation 3D plane and data at both ends through ground estimation. Classified environmental data facilitates the labeling of all objects within the viewing angle of 3D LIDAR based on the presence or absence of movement. The path of motion of the platform can be established by detecting whether objects within the region of interest are movable or static. Because LIDAR is based on 8- and 16-channel rotation mechanisms, real-time data cannot be used to define objects. Instead, point clouds can be used to detect obstacles in the image through deep learning in the preliminary processing phase of the classification algorithm. By matching the labeling information of defined objects with the classified object cloud data obtained using 3D LIDAR, the exact dynamic trajectory and position of the defined objects can be calculated. Consequently, to process the acquired object data efficiently, we devised an active-region-of-interest technique to ensure a fast processing speed while maintaining a high detection rate.

INDEX TERMS 3D LIDAR, 2D vision, interpolation, object recognition, intelligent vehicles, big data.

I. INTRODUCTION

Autonomous vehicles detect and recognize various types of fixed and moving objects, such as roads, vehicles, obstacles, and pedestrians, using data obtained from sensors such as radar, LIDAR (Light Detection and Ranging), and cameras installed in vehicles. An autonomous vehicle can judge the driving situation around itself, plan the driving route, and safely arrive at the destination independently, thereby realizing automatic driving while minimizing driver input [1]. The operation of autonomous vehicles depends on the sequential processing of recognition, judgment, and control systems [2].

The driving environment-recognition system consists of sensors that identify the location of the vehicle and recognize static and moving objects and obstacles in the driving space, including roads. The driving judgment system receives

The associate editor coordinating the review of this manuscript and approving it for publication was J. D. Zhao.

the output of the driving environment-recognition system to determine the driving conditions with the surroundings and on the road and establish a safe and efficient driving strategy. The control system controls the horizontal and vertical movement of the vehicle including acceleration, deceleration, and steering according to the driving strategy and the route plan set by the driving situation determination system and actuators coupled to the control system [3].

The resolution of 3D LIDAR is lower than that of the image sensor, but 3D LIDAR can accurately measure depth information about the 3D environment [4]. Therefore, many object-recognition studies have used point cloud data obtained using 3D LIDAR. LIDAR scans the environment by dividing the vertical field of view (FOV) by at least one rotation and n angles, where n is the number of LIDAR channels. The performance of LIDAR varies depending on the number of channels and the installation position of the LIDAR [5]. The data of a 64-channel LIDAR are denser than that of a 16-channel LIDAR. LIDAR systems with 32 or more channels are referred to as high-channel LIDARs, and LIDARs with fewer than 32 channels are referred to as low-channel LIDARs. High-channel LIDARs are suitable for object recognition because they offer high resolution in which object features appear clearly. Object recognition using a high-channel LIDAR includes 3D [6]. Although it is advantageous to use a high-channel LIDAR to improve object-recognition performance, high-channel LIDARs are expensive and require considerable computing resources to process the data acquired through multiple channels. To overcome these problems, many recent studies have used low-cost low-channel LIDARs. The data acquired using a low-channel LIDAR should be processed because they contain insufficient object features owing to the low resolution of such LIDARs.

Furthermore, the data acquired using a 3D LIDAR are sometimes added to the movement trajectory of the platform to detect and discern objects using overlapped data over time. Accurate object positioning becomes difficult in this case because the errors in the data obtained from the global positioning system and inertial measurement unit (IMU) are added to the errors in the data obtained from 3D LIDAR. Data fusion by means of simple unification of only data coordinates causes dispersion of points. Therefore, some algorithms define objects by learning the characteristics of data obtained from LIDAR. Representative examples include VoxelNet, PointNet, and fully convolutional networks. Previous studies used 32-channel LIDAR, but PointNet defines objects using 64- or higher-channel LIDARs. Therefore, effective obstacle detection is enabled by fusing the features of sensors to ensure that they complement each other. However, environment-recognition data need to be unified to integrate and interoperate the sensors with diverse features. Moreover, the accurate matching of unified data and fusion over time is required.

Recently, Caltagirone acquired information through sensor-level processing of the data obtained using 3D LIDAR and image sensors separately, managed a track list of 3D LIDAR and image sensors, and finally fused the two sets of data through the unification of pixel coordinate systems [7]. However, this method invariably requires calibration before initiating the process. Furthermore, fast fusion is impossible because of the heavy system load imposed by the management of the processes of individual sensors. To solve the sensor-level and computing resource problems that occur in the fusion of multiple sensors [8].

The frequency modulated continuous wave (FMCW) radar, which is primarily used as a vehicle radar, cannot obtain the size and classified states of detected objects but only the positions of objects that are included in the radar frequency region and reflected. Consequently, it cannot define objects and cannot detect obstacles beyond the vertical detection distance of the sensor. 3D LIDARs have been used in all recent autonomous vehicles and platforms because they offer the advantage of expressing the reflected light of all objects as points, thus making the images appear as if they contain depth information.

3D LIDARs are classified into high-channel and low-channel based on the number of light-emitting parts, but for accurate object detection, a high-channel 3D LIDAR must be used in most cases. High-channel 3D LIDARs require vast computing resources to process significant amounts of point cloud data, which makes them expensive. In contrast, a low-channel 3D LIDAR can be implemented at a low cost, but its accuracy is low given that it must detect obstacles with feature points obtained from limited data because it obtains fewer data points in 3D environments.

In this paper, we have proposed an effective obstacle detection method that fuses the features of multiple sensors to ensure that they complement each other. However, the environment-recognition data of all sensors must be unified to integrate and interoperate the sensors with diverse features, and accurate matching of unified data and fusion over time are required. We have devised and proposed an active viewing angle adjustment technology to solve the processing speed problem caused by using various sensor data. The convergence of radar and 3D rider and image data is important, but the range of interest (ROI) can be actively changed in proportion to the detected object's position and the moving position to solve this based on the vehicle's moving speed. Thus, superiority was verified in comparison with other studies on object detection—processing speed increased by approximately 30–40%, and a map was obtained.

II. GEOMETRIC MATCHING BETWEEN 3D LIDAR AND IMAGES

Instead of using 3D LIDAR data directly to detect objects, transforming them into 2D image coordinates can help reduce the amount of computation required. Two methods are available to transform the coordinates of 3D LIDAR data into 2D image coordinates: top-view and polar-view.

The top-view method involves transforming 3D LIDAR data into image coordinates by multiplying the (x, y) axis with a simple constant as shown in Fig. 1. This method can help achieve clear object recognition because the point data on

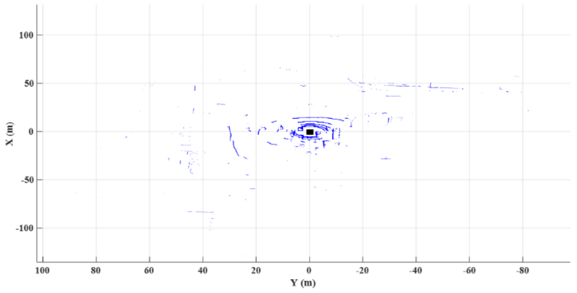


FIGURE 1. 3D LIDAR data (top-view).

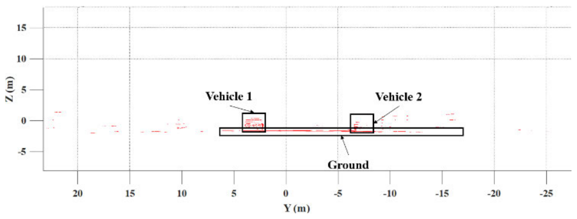


FIGURE 2. 3D LIDAR data (polar-view).

the z-axis, which is the direction of progress, do not overlap. However, the data become meaningless if they cannot be acquired using a multi-channel LIDAR.

The polar-view method of Fig. 2 uses the spherical coordinate system (r, θ, ϕ). Compared to the top-view method, the detectable area is determined by θ, and if multiple data are located at the same θ, the objects may appear to be overlapped. However, clear object recognition is possible even with minimal data. To combine with image sensors that have a fixed FOV, matching must be performed using 2D image frame coordinates. The polar-view method is more appropriate because it can set the LIDAR region for data acquisition by performing detection in all directions.

A. 3D LIDAR ROTATIONAL TRANSFORMATION OF COORDINATES

Sensor information can be unified around the vehicle coordinate system if the behavior information of the vehicle is acquired as. A rotational transformation must be performed to update the rotational information of the three axes in the sensor data. The rotational transformation to be applied can be determined using the following equations:

$$\mathbf{R}_x(\theta_x) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_x & -\sin\theta_x \\ 0 & \sin\theta_x & \cos\theta_x \end{bmatrix} \tag{1}$$

$$\mathbf{R}_y(\theta_y) = \begin{bmatrix} \cos\theta_y & 0 & \sin\theta_y \\ 0 & 1 & 0 \\ -\sin\theta_y & 0 & \cos\theta_y \end{bmatrix} \tag{2}$$

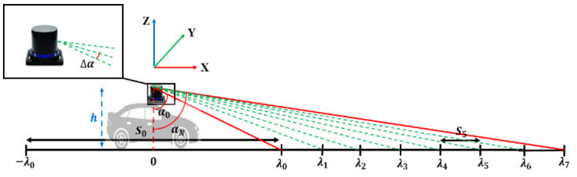


FIGURE 3. Ground estimation expressed in terms of each channel of 3D LIDAR.

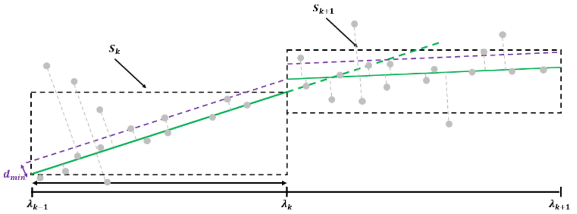


FIGURE 4. Expression of 3D plane fitting using RANSAC.

$$\mathbf{R}_z(\theta_z) = \begin{bmatrix} \cos\theta_z & -\sin\theta_z & 0 \\ \sin\theta_z & \cos\theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3}$$

Eqs. (1–3) can be combined counterclockwise from each axis using the basic translation method to express them in the form of a random 3D rotation matrix as follows:

$$\mathbf{R} = \mathbf{R}_z(\theta_z)\mathbf{R}_y(\theta_y)\mathbf{R}_x(\theta_x) \tag{4}$$

Furthermore, the rotation matrix obtained with a rotation angle θ along a random unit vector $\mathbf{u} = (u_x, u_y, u_z)$ as the rotation axis, as opposed to the coordinate axis, is given by Eq. (5), as shown at the bottom of this page. This can be expressed in the linear translation form using 3D parallel movement and $\mathbf{t} = [t_x\ t_y\ t_z]^T$, as in Eq. (6). [9], (5) as shown at the bottom of this page, where $c_\theta = \cos\theta$ and $s_\theta = \sin\theta$.

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \mathbf{R}_u \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \tag{6}$$

B. GROUND SEGMENTATION USING 3D RANSAC

The ground and obstacles in the data acquired using 3D LIDAR are estimated using the Random Sample Consensus (RANSAC) algorithm as shown in Fig. 3 and 4. The ground and obstacles exhibit strong data dependencies because they are located on the surfaces of roads and streets. In most studies, object detection is performed under the assumption that the ground is flat, and all objects are standing [10]. The error of the recommended lane is declined by the decision of the lane type, and it is beneficial for making comparative decisions when performing rapid computations in high-speed

$$\mathbf{R}_u(\theta) = \begin{bmatrix} c_\theta + u_x^2(1-c_\theta) & u_xu_y(1-c_\theta) - u_zs_\theta & u_xu_z(1-c_\theta) + u_ys_\theta \\ u_yu_x(1-c_\theta) + u_zs_\theta & c_\theta + u_y^2(1-c_\theta) & u_yu_z(1-c_\theta) - u_xs_\theta \\ u_zu_x(1-c_\theta) - u_ys_\theta & u_zu_y(1-c_\theta) + u_xs_\theta & c_\theta + u_z^2(1-c_\theta) \end{bmatrix}, \tag{5}$$

situations [13]. Standard filtering produces inaccurate results because most actual driving environments are not constant. Thus, in this study, we estimate ground types, such as uphill and downhill, in an actual road environment—by applying 3D RANSAC—and perform advance filtering for clear obstacle detection.

The RANSAC algorithm does not re-predict the model from the sum of the current errors but instead creates an approximate model directly from the predicted model. This enables high-speed processing because it reduces the time required for comparing the errors related to the model predicted from a small number of observation points. However, the algorithm is sensitive to noise because the errors are not compensated. [11] Unlike the least-squares method, which uses the entire dataset, the RANSAC method estimates the model parameters by applying different observation values to each subset of the data. For the subsampled data, the region of each 3D LIDAR channel is divided by λ_k, and $\lambda_0 = 2m$ is defined. Furthermore, λ_k can be calculated using Eq. (7). $a_0 = \tan^{-1}(\lambda_k/h)$ and $h = h_{offset} + 400$ mm; thus, $h = 1,750\ mm$. The number of regions N is calculated using Eq. (8), and $\Delta a = 1.2°$.

$$\lambda_k = h \cdot \tan(a_0 + k \cdot \eta \cdot \Delta a), \quad \{k : 1, \dots, N\} \tag{7}$$

$$N = \frac{a_N - a_0}{\eta \cdot \Delta a}, \quad \eta = 2 \tag{8}$$

Using this method, the regions of S_k, etc., are divided, as depicted in Fig. 5. Furthermore, a random 3D plane model is created using the expression $a_k x + b_k y + c_k z + d_k = 0$. In this plane model, the attitude of the autonomous vehicle $(\theta_x, \theta_y, \theta_z)$ and the rotation matrix given by Eq. (5) are calculated using Eq. (6) and then combined. [14]

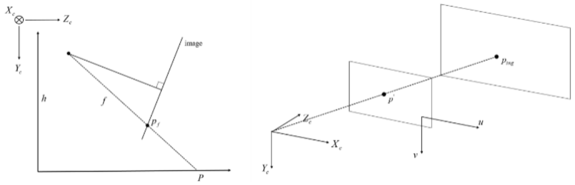


FIGURE 5. Coordinates of image sensor.

C. 3D LIDAR MATCHING THROUGH INTERPOLATION OF IMAGE COORDINATES

The 3D LIDAR information of a 2D image can be matched by transforming the coordinates expressed as 3D LIDAR data from Fig. 6 and then matching them with the FOV of the image sensor. The image frame is configured as depicted in Fig. 5. $\mathbf{P} = (X, Y, Z)$ becomes a random object point on the global coordinate system. For the camera, the coordinates of $\mathbf{P}_c = (X_c, Y_c, Z_c)$ can be set, as depicted above. Furthermore, the pixel coordinates in the image frame become $\mathbf{p}_{img} = (x, y)$. These coordinates represent the image seen by the naked eye. With the left top corner of $\mathbf{p}_{img}$ in Fig. 4 as the origin, the coordinates increase along the y-axis toward the left and the x-axis toward the right. [15]

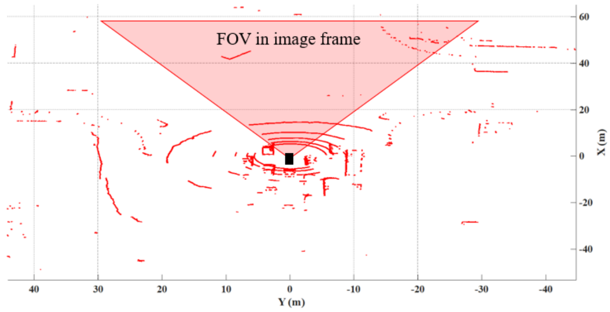


FIGURE 6. Matching with 3D LIDAR and vision FOV.

To solve this problem geometrically, a random point P in 3D space is created to pass through the focus of the camera and projected onto the image plane coordinate point $\mathbf{p}_{img}$. Finally, a virtual image coordinate system is defined and expressed as a normal coordinate system by removing the sum of the internal parameters of the camera. The origin of the normal coordinate system is its intersection with the optical axis Z_c, which is the midpoint of the normal image plane, and its position differs from that of the pixel coordinate system. If the image coordinate system is assumed as $\mathbf{p} = (u, v)$, a transformation between the pixel coordinate system and the normal coordinate system is possible using Eq. (9).

In the Eq. (9), f_x and f_y are focal distances, and c_x and c_y are the coordinate values for the intersection between the optical axis and the pixel plane, referred to as camera matrix $\mathbf{C}$. Furthermore, $\mathbf{K}$ is a 3 × 3 matrix consisting of the camera internal parameters, as in Eq. (10). Then, a matrix transformation through camera calibration is performed as follows:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \tag{9}$$

$$\mathbf{p}_{img} = \mathbf{K}\mathbf{p}' \tag{10}$$

$$\mathbf{C} = [\mathbf{K}|\mathbf{t}_k] \tag{11}$$

$$\mathbf{t}_e = \mathbf{K}^{-1}\mathbf{t}_k \tag{12}$$

$$\mathbf{C} = \mathbf{K}[\mathbf{I}|\mathbf{t}_e] \tag{13}$$

Furthermore, the matrix consisting of the rotation matrix $\mathbf{R}$ and the translational motion matrix $\mathbf{t}$ is generally referred to as an external matrix because it represents the external information of a camera. The external matrix can be considered a translation matrix that transforms the global coordinate system into the camera coordinate system. Assuming the points in the 3D space are projected onto the image plane through $\mathbf{K}$, the location to which a point in the 3D space is projected onto the image plane can be determined using Eq. (10).

In the case of 3D LIDAR, a 3D map is obtained with the LIDAR at the origin. However, for 3D-2D matching, not all 3D information can be used because the information in the image is used, and the 3D information is projected onto the image plane. Therefore, the reference 3D coordinates of the camera can be obtained if there is no $\mathbf{K}$ in Eq. (12).

The values obtained by multiplying **K** represent the 2D coordinates in the image of the corresponding 3D coordinates. By combining these values, a depth image that has the same type of 3D coordinate information can be created [16]. However, this depth image does not contain a considerable amount of the initial 3D information owing to differences in the viewing angle between the camera and LIDAR sensor. Because information is distributed, it is likely that there is no 3D information at the location of a detected feature. Object detection may be impossible if this gap in 3D information is significant.

The data may exist as uneven, irregular forms of the 2D space in the frame in which 3D LIDAR data obtained by projecting the above 3D LIDAR data are matched with the image. Thus, they are aligned using the interpolation method [17]. Dilation interpolation involves the interpolation of sparse information using the dilation operation, which a fundamental image processing technique. To perform dilation, the image to be dilated and the corresponding structuring elements are required. The result of dilation interpolation for an image matched with 3 × 3 structuring elements is depicted in Fig. 7. First, the edge features of uniform objects in the front are determined from the LIDAR point data. To interpolate between the detected data and the pixel coordinates of the image, the data are placed in a grid map, as depicted in Fig. 7. $\mathbf{P}_i^k$ denotes the 3D coordinates of the i-th position in the k-th frame. The 3D information can be interpolated by executing the dilation operation according to a predefined mask, and the overlapping parts can be replaced with the vector average values of the 3D coordinates [18].

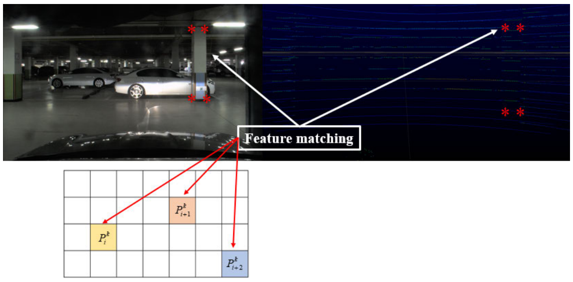


FIGURE 7. Interpolation using feature matching and dilation operation.

III. OBJECT-RECOGNITION ALGORITHM

A. OBJECT RECOGNITION BASED ON VISION WITH YOLO NETWORK

We used the YOLO network in this study to perform object recognition and classification. The YOLO network simultaneously locates and classifies bound boxes from the final output of the network. One network extracts features, creates bound boxes, and performs classification simultaneously. As depicted in Fig. 7, two data points in the center are created from the raw input image using the algorithm. Encoded in these data is information about the classes in the cells of the corresponding grid when multiple bound boxes and images are divided into *nxn* grids.

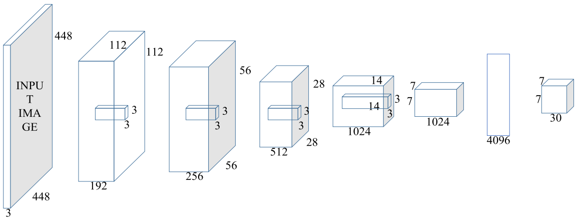


FIGURE 8. Basic structure of faster YOLO network.

The network in Fig. 8 classifies images into 7 × 7 grids. In each grid, two bound boxes of different sizes with their centers inside the grid are created; 98 bound boxes are created because there are 49 grid cells. The bound boxes are indicated using thicker lines because the probability that these boxes contain an object is higher. When the objects are selected using the non-maximum suppression (NMS) algorithm for the remaining candidate bound boxes, we obtain the final image, as depicted in Fig. 8.

$$\begin{aligned}
&\lambda_{coord}\sum_{i=0}^{s^2}\sum_{j=0}^{B}1_{ij}^{obj}[(x_i-\hat{x}_i)^2+(y_i-\hat{y}_i)^2]\\
&+\lambda_{coord}\sum_{i=0}^{s^2}\sum_{j=0}^{B}1_{ij}^{obj}[(\sqrt{w_i}-\sqrt{\hat{w}_i})^2+(\sqrt{h_i}-\sqrt{\hat{h}_i})^2]\\
&+\sum_{i=0}^{s^2}\sum_{j=0}^{B}1_{ij}^{obj}(C_i-\hat{C}_i)^2+\lambda_{coord}\sum_{i=0}^{s^2}\sum_{j=0}^{B}1_{ij}^{noobj}(C_i-\hat{C}_i)^2\\
&+\sum_{i=0}^{s^2}1_{i}^{obj}\sum_{c\in classes}(p_i(c)-\hat{p}_i(c))^2 \qquad (14)
\end{aligned}$$

To derive 7 × 7 × 30, which is the prediction result depicted in Fig. 3, four convolution layers and two full-connection layers are formed in the input image. The bound boxes determined in this manner are divided into classes. Multiple classes are filtered using the defined threshold for the values expressed as probability values. However, if the object size is large, the number of classes with a high probability of housing an actual object increases. The classes are filtered using the NMS algorithm to determine the final object bound box, as in Eq. (14), where S is the number of grids and B is the predicted number of bound boxes in each grid cell.

The corresponding loss function includes a sun-squared error for optimization. λ_{coord} is the trained gain when there is an object, and λ_{noobj} is the gain when there is no object. Furthermore, in Eq. (14), 1^{obj} is a cell that contains an object, 1^{noobj} is a cell that does not contain any object, and 1_{ij}^{obj} is a value predicted using the j-th bound box that exists in the i-th grid cell. Hence, loss occurs only in the i-th grid cell in which there is an object, and in the case of the grid cells that do not contain any object, the loss is not calculated. The loss can be calculated using one of five methods:

1: The (x, y) coordinate of the bound box is trained in such a manner that when an object exists in the i-th grid cell, the j-th bound box in the j-th grid cell is identical to the object data that have been extracted. Furthermore, the two bound boxes ($j = 0$, $i = 0$) predicted using the gird cells are induced to be matched.

2: The width and height of the bound box are trained. This is the same as in the first method, except that the root is added. For a large bound box, the area increases greatly even when the width and height increase only marginally, and the value of the derivative is large. For a small bound box, the area does not change considerably even when the width and height are increased significantly, and the value of the derivative is small. The root is used because the use of the sum-squared error generates a large difference in derivatives between large and small bound boxes.

3: The class is predicted when there is an object in the i-th grid cell.

4: The class can be predicted when there is no object in the i-th grid cell. Because the weighted value is not large compared to method 3, the grid is searched more reliably when an object exists, as in method 3, due to training.

5: There is no sum of B in the equation. The bound box in the i-th grid cell predicts $B = 2$, but the sum is not calculated because the c of 20 class probabilities is shared.

Next, the NMS algorithm is executed. First, the box with the highest detection probability score is selected from among the current set of detection boxes of interest. Then, the intersection over union (IOU) between the selected box and the ground truth box is determined; if it is smaller than the specified threshold, the selected box is removed from the list. Furthermore, the IOU, which is the ratio of union and intersection, approaches 1 as the area of overlap between the selected box and the ground truth box increases. This process is repeated until there is no case in which the detection probability of the selected box is smaller than X. After this, the multiple detection regions in which one object is captured are reduced to a single region, as depicted in Fig. 8.

The obstacles must be detected by setting the actual ROI in the converted (x_r, y_r, z_r). As depicted in Fig. 9, the data of $\mathbf{p}_0 \sim \mathbf{p}_n$, detected as point cloud data for one object, must be grouped and defined as $\mathbf{P}_m$. $\mathbf{P}_m$ denotes cloud data when the number of obstacles classified in the extracted frames per hour is m. Particle filtering is performed continuously between the images and the objects in the calibrated images through $\mathbf{P}_m$. The objects that are lost during driving are predicted and compensated for using the results of particle filtering.

Many methods are available for classifying cloud data by finding regularity among multiple point data, such as a support vector machine and the k-means algorithm [12]. However, the processing speeds of these methods decrease as the number of samples increases, and their accuracy varies depending on the number of repetitions. Consequently, considerable optimization is required to classify and track obstacles in 3D LIDAR data when driving on an actual road. Therefore, in this study, we perform high-speed object detection using an object detection-method for image sensors based on a deep learning technique, as depicted in Fig. 9, and set the object-detection coordinates in the images as the ROI.

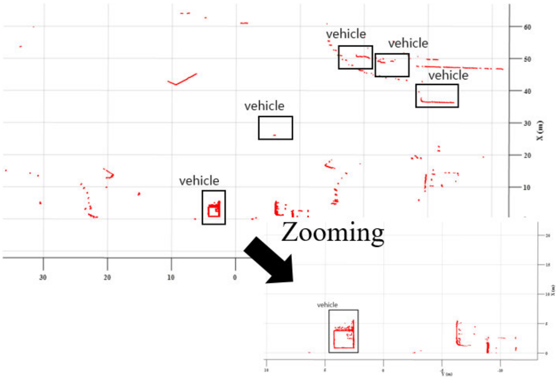


FIGURE 9. Result of object detection by fusing data.

TABLE 1. Configuration of vision sensor.

Training image resolution	4K, 4,096 × 2,160 px
Camera resolution	1,280 × 960 px
Camera FPS	30 FPS
Camera interface	USB 3.0

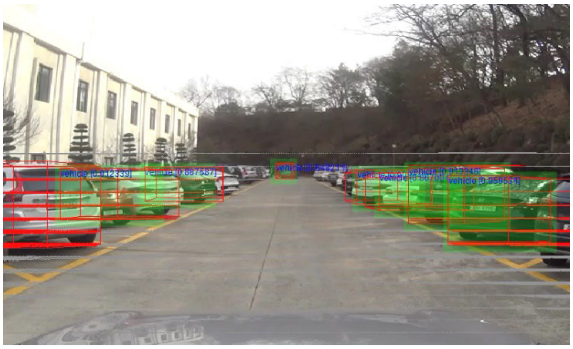


FIGURE 10. Result of multiple-object detection with 3D LIDAR and vision.

As summarized in Table 1, the obstacle detection coordinate system through image data is represented by pixels corresponding to the resolution of the image frame. Furthermore, the class ID for prior training is included as a header. Thus, for matching with point data in the rectangular coordinate system, calibration was performed according to the sensor positions. The actual camera view angle of the autonomous vehicle is 68°, and the 3D LIDAR sensor is attached in front at d_{offset} from the camera. Therefore, in Fig. 10, both sides of the image frame corresponding to the coordinates 0–1280 of y_f are removed because they are lost zones. Furthermore, 3D LIDAR data (x_r, y_r, z_r) for h_{offset} are added to prevent distortions when the ROI is set in 3D LIDAR coordinates based on the image-detection result. Then, an arbitrary grid

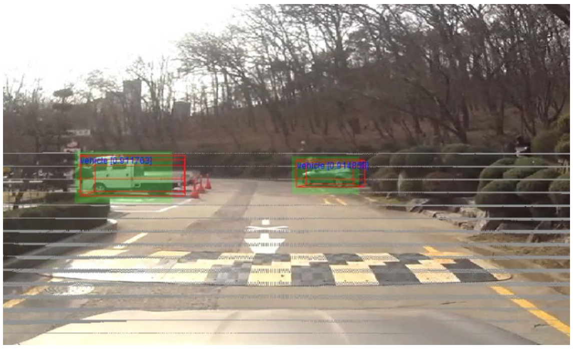


FIGURE 11. Result of object detection with 3D LIDAR and vision.

map is created in the object coordinate system of the 3D LIDAR sensor. Thereafter, to match the lines of the grid map with the coordinate points in the pixels acquired from the image, the coordinates in the pixel are arranged within a fixed range in S_i. Furthermore, the ROI is set in the lines considering that the detection distance of 3D LIDAR is 24 m.

The result is calculated for matching with the objects detected in the image after the midpoint of the object. In this way, the prediction is generated through the particle filter with no loss from the continuous images of the detected objects. First, the direction x_t, position r_t, and state vector s_t at time t are defined for the $\mathbf{P}_m$ of the objects. Then, the predicted locations of the detected objects are determined using Eq. (15). When the objects move from the predicted positions to new positions, the predictions are generated using Eq. (16).

$$B(x_t) = p(x_t|s_t, r_t, s_{t-1}, r_{t-1}, \ldots, s_0, r_0) \tag{15}$$

$$B^{-}(x_t) = \int p(x_t|x_{t-1}, r_t)B(x_{t-1})dx_{t-1} \tag{16}$$

$$B(x_t) = \eta_t p(s_t|x_t)B^{-}(x_t) \tag{17}$$

$$P_{m,t} = \{(x_t^i, w_t^i)|i = 1, \ldots\ldots, n\} \tag{18}$$

B. IMAGE SEGMENTATION FOR ADAPTIVE ROI APPLICATION IN IMAGE FRAME

In the process of matching the three-dimensional and two-dimensional data and interpolating it in one frame and then detecting obstacles based on the interpolated data, significant process time is required. Therefore, in this study, the area corresponding to the predicted section of the route to which the vehicle is coming or going is searched for first, thereby reducing the unnecessary process load, and increasing the speed increases the range of the autonomous vehicle.

In this chapter, the active area of interest technique is used to efficiently solve the system latency problem of sensor fusion technology. For the active area of interest scheme, the overall system flow diagram is structured as depicted in Fig. 12. First, when the system is initiated, the objects are detected and tracked according to the fused result of the image and the 3D ray. The centerline of the horizontal axis, according to the position and number of the object, is determined through the subsequent results. The area of interest is determined as a product of a value proportional to the vehicle speed along the defined baseline of the horizontal axis.

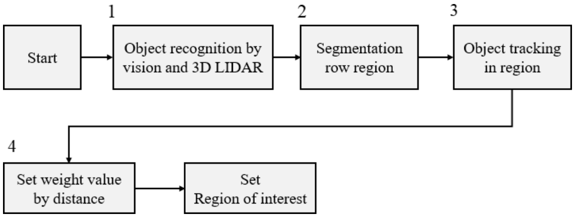

Start
1
Object recognition by vision and 3D LIDAR
2
Segmentation row region
3
Object tracking in region
4
Set weight value by distance
Set Region of interest

FIGURE 12. Workflow of adaptive ROI process.

The detected object in the determined area can calculate position using the object tracking technique. The distance between the vehicle and the object can be determined through the calculated position value, and a weight value proportional to the distance to the object can be derived. The region of interest in the x-axis region can be subdivided through the weights, and the closer the distance to the vehicle, the higher the interest of the object.

Based on this study, it is possible to use the object detection system more efficiently by dividing the region of interest based on the risk of the object without examining the entire image frame.

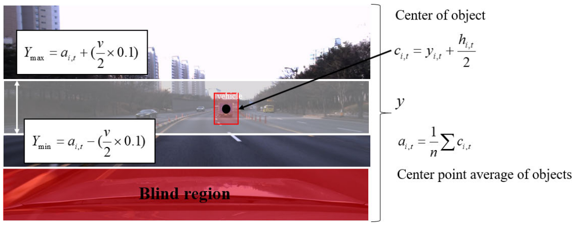

$Y_{\max} = a_{i,t} + (\frac{v}{2} \times 0.1)$
$Y_{\min} = a_{i,t} - (\frac{v}{2} \times 0.1)$
vehicle
Blind region
Center of object
$c_{i,t} = y_{i,t} + \frac{h_{i,t}}{2}$
y
$a_{i,t} = \frac{1}{n}\sum c_{i,t}$
Center point average of objects

FIGURE 13. Using a centroid of object for segmentation a row.

First, to execute this process, the position of the object detected in the image frame is calculated based on the image resolution, as depicted in Fig. 13. If there are many objects to be detected in the image, calculate the average of the object's center point, as depicted in Eq. 19, and define it as $a_{i,t}$, where $a_{i,t}$ is the y-axis baseline of the computable area in the image frame. The range to be detected must be set through the y-axis reference line. The range can be calculated in conjunction with the vehicle speed.

As the speed increases, the section where the object must be examined during driving becomes narrower. Therefore, in this study, the proportional value based on the vehicle speed can be defined by Eq. 20, and the y-axis area can be specified in the image frame. The detection area applied in the first step can be applied to the detection of the object through the fusion between the 3D sensor and the image sensor applied based on the results of the position tracking study.

The object detected in the y-axis region of interest can be computed as a result of the previously-studied object and can

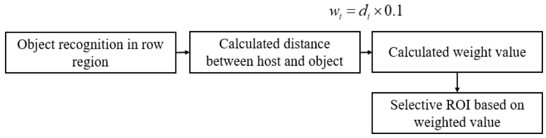

$w_i = d_i \times 0.1$
Object recognition in row region
Calculated distance between host and object
Calculated weight value
Selective ROI based on weighted value

FIGURE 14. Process of Adaptive ROI based on object weight.

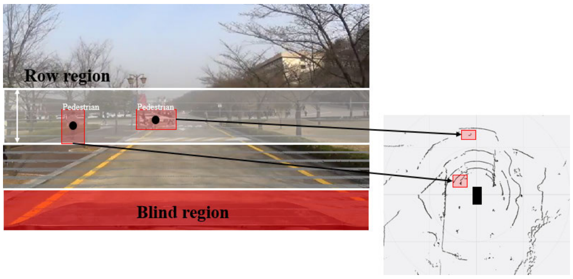


FIGURE 15. Result of calculation weight value from object.

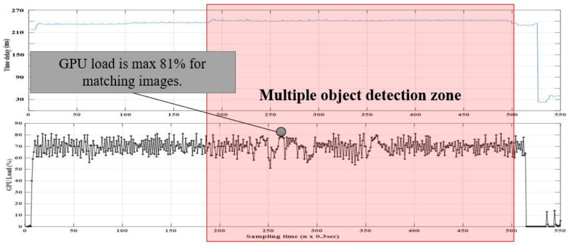


FIGURE 16. GPU utilization with 3D LIDAR and vision.

be calculated as a continuous trajectory through the particle filter. In this study, the risk is calculated as the risk weight between the vehicle and the obstacle. Then, the distances are weighted by distances for detected obstacles, as depicted in Figs. 10 and 11—the closer the distance to the vehicle, the more continuity is obtained for detection.

$$c_{i,t} = y_{i,t} + \frac{h_{i,t}}{2} \tag{19}$$

$$a_{i,t} = \frac{1}{n}\sum c_{i,t} \tag{20}$$

IV. RESULTS AND DISCUSSION

The object detection system developed by fusing a 3D LIDAR and an image sensor achieved a detection rate of 78.3%. Furthermore, the proposed system alleviates to a considerable extent the disadvantages of the object-detection method using deep learning. In the case of the image sensor, normal detection is difficult when many objects appear simultaneously, but the proposed system that fuses an image sensor with a 3D LIDAR can detect all objects encountered by the system.

A single sensor has a long delay, as depicted in Tables 2–5, and the delay time of the image sensor is approximately 70 ms. However, the delay time of the proposed system is at least 200 ms. Furthermore, at 81%, the GPU utilization of the proposed system is considerably higher.

TABLE 2. Result of object detection rate (vehicles).

-	Fusing	FPS	Time delay	GPU Load	Recall	mAP
Vision only	X	13	78ms	65%	0.75	57.9
Vision + Radar	O	9	115ms	70%	0.88	56.2
Vision+3D LIDAR	O	4	230ms	81%	0.85	78.3
Vision+3D LIDAR (Adaptive ROI)	O	12	93ms	82%	0.85	78.3

TABLE 3. Result of object detection rate (pedestrian).

-	Fusing	FPS	Time delay	GPU Load	Recall	mAP
Vision only	X	13	78ms	65%	0.58	43.1
Vision + Radar	O	9	115ms	70%	0.58	43.9
Vision+3D LIDAR	O	4	230ms	81%	0.61	55.4
Vision+3D LIDAR (Adaptive ROI)	O	12	93ms	82%	0.62	55.4

TABLE 4. Comparison of detection rate with other research (vehicles).

Approach	Fus ing	Sensors	Time delay	Rec all	mAP
MV3D[110]	O	3D LIDAR+RGB	0.36	0.75	68.49
OH[111]	O	3D LIDAR+RGB	1.3	0.62	81.4
OURS	O	3D LIDAR+RGB	0.2	0.85	78.3
OURS	O	3D LIDAR+RGB (Adaptive ROI)	0.093	0.85	78.3

TABLE 5. Comparison of detection rate with other research (pedestrian).

Approach	Fus ing	Sensors	Time delay	Reca ll	mA P
F-PC-CNN[112]	O	3D LIDAR+RGB	0.5	0.51	45. 22
MV3D[110]	O	3D LIDAR+RGB	0.36	0.58	55. 12
OURS	O	3D LIDAR+RGB (Adaptive ROI)	0.093	0.62	55. 4

The purpose of this study is to investigate an object detection system with a faster processing speed through the active area of interest control. The experimental environment was conducted on a campus where many moving objects and stationary objects could be detected irregularly. As depicted

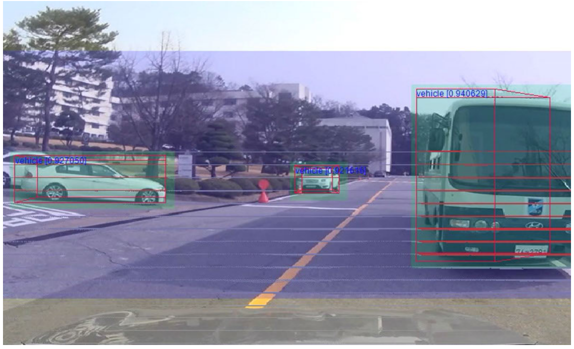


FIGURE 17. Result of object detection with 3D LIDAR and vision.

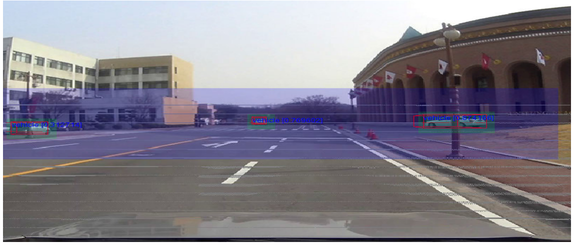


FIGURE 18. Result of vehicles detection by LIDAR and vision.

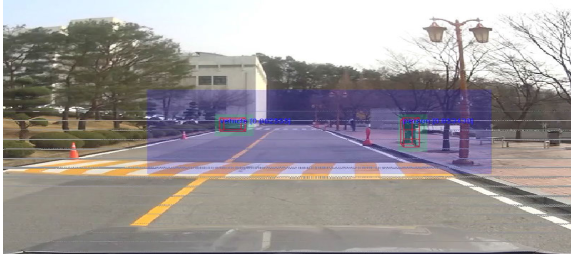


FIGURE 19. Result of pedestrian detection with 3D LIDAR and vision.

in Fig. 17 ∼ 19, the detection accuracy is high in the fused frame between the 3D image and the image for the object detected in the driving environment. Furthermore, even if a large number of objects are detected, the object is detected and tracked accurately. The object in the detection box of Fig. 17 ∼ 19 are continuously detected, while the object in the other area is not influenced by the processing load.

Furthermore, as depicted in Fig. 17 ∼ 19, the row detects the entire area to detect abrupt pedestrians that may appear in front of the vehicle, and the y-axis area is controlled, such that it is possible to cope with the unexpected situation along the road. To reduce process load due to the detection of the entire x-axis region, the load factor is significantly reduced by dividing the region of interest around the detected object.

Considering that it requires more than 200 ms in process delay time to detect the whole area through fusion between the existing 3D LIDAR and the image, the delay time of the proposed system is approximately 100–120 ms, which is a 50% reduction.

However, because it is necessary to detect the entire area of the x-axis in a section where a large number of objects is detected, a delay time of approximately 130 ms is required. Nonetheless, the results illustrate that the delay time reduction effect is approximately 35%, and the detection rate is not significantly affected.

V. CONCLUSION

In this study, we developed an object-detection system that matches 3D LIDAR data with image frames. First, to remove noise from 3D LIDAR data, segmentation between the ground and the objects was performed using the 3D RANSAC method. For the accurate definition of object data in the segmented 3D data, matching between the 3D images and the 3D data was performed using interpolation. To reduce the inherent inaccuracy of position information of the 3D images and facilitate accurate object positioning, the positions of the objects detected in the images were matched with the 3D data points. Fast processing speed was achieved by implementing dilation interpolation, with the associated drawbacks alleviated by removing the preprocessed 3D noise data.

The result of object detection using a 2D image-based object detector. The five vehicles depicted could be detected discontinuously depending on the image condition, and their exact positions cannot be known even if they are detected. However, Fig. 17 ∼ 19 illustrates the matched image, in which the data containing the depth information of each object are matched to each feature point. We used a single image sensor, and the other sensors supplemented the continuity of object tracking and recognition based on the object results recognized through the image sensor. Moreover, to address the disadvantages of the technologies, we proposed an optimization technique.

As depicted in Fig. 18, using the image sensor, the delay time based on object detection is 50 ms, and even if many objects are recognized, thee processing time is 78 ms. However, the object detection rate was 57.9%. When used in other studies, the image detection technique used in this study—with a high detection rate of 70–90%—exhibited a low detection rate because it recognized and detected not only the car but also the pedestrian; the detection rate according to the recall calculated within the currently obtainable frame was confirmed without setting the limit distance. The purpose of this study is to investigate an object detection technique to enable autonomous navigation in a real-world environment, so no other constraints are made, and the results are not produced.

The object detection and follow-up study through convergence between RADAR and image sensor illustrates the same result as the detection rate of the previous research. Because object detection using only image sensors is a single-stage-based recognition technology for high-speed object recognition, the continuity of object recognition is greatly reduced if the relative speed is high or the environment is affected by the image. To compensate for this, we use a radar that can acquire the existence of objects in two-dimensional space points.

The resulting detection rate is 56.2%. The object detection and tracking depend on the result of the image sensor, so the detection rate was similar. However, as depicted in Fig.16, it is possible to produce the exact result once detected.

The convergence between 3D LIDAR and the image sensor enabled more accurate object detection and tracking due to the large quantity of 3D point data, including accurate distance values. As depicted in Table 4, the detection rate was as high as 78.3%, which is a result of comparison with other studies. Moreover, the high data of 3D LIDAR was able to be controlled only by a system load of 81% of the GPU. However, we have conducted research to improve the high latency time of 230 ms.

To address these disadvantages, the concept of adaptive ROI was proposed, and low latency of 93 ms was achieved without a reduction in the detection rate. Based on the object tracking, all obstacles in the obtainable frames are detected, and the ROI is actively-controlled based on the risk to the result, and the delay time reduction achieved is at least 50%. The system can detect a pedestrian and the car simultaneously, and can detect the situation even when the relative speed is great than 60 km/h, as depicted in Fig. 18. As presented in Table 5, most studies detect an object with a delay time of at least 300 ms and a detection rate of 55% or less when simultaneous detection of a pedestrian and a car is performed. However, the proposed study has a detection rate of 78.3% and a delay time of less than 100 ms.

REFERENCES

[1] J.-C. Son and J.-D. Choi, "Autonomous driving car system technology direction and task," *J. Korean Ins. Commun. Sci.*, vol. 35, no. 5, pp. 21–27, 2018.

[2] I.-S. Weon and S.-G. Lee, "Environment recognition based on multi-sensor fusion for autonomous driving vehicles," *J. Inst. Control, Robot. Syst.*, vol. 25, no. 2, pp. 125–131, Feb. 2019.

[3] N. H. Amer, H. Zamzuri, K. Hudha, and Z. A. Kadir, "Modelling and control strategies in path tracking control for autonomous ground vehicles: A review of state of the art and challenges," *J. Intell. Robotic Syst.*, vol. 86, no. 2, pp. 225–254, May 2017.

[4] I.-S. Weon, S.-K. Kim, H.-R. Kim, S.-G. Lee, Y.-J. Kim, and S.-H. Woo, "Development of KHUV (Kyung Hee unmanned vehicle) for autonomous driving," in *Proc. 33rd ICROS Annu. Conf. (ICROS)*, 2018, pp. 188–189.

[5] S.-C. Moon, M.-W. Kim, D.-N. Joo, and S.-G. Lee, "GPS-RTK solution using WAVE communication for vehicle," in *Proc. 20th ITS World Congr.*, 2013, p. 10.

[6] I.-S. Weon and S.-G. Lee, "Lane departure algorithm based on classification roadway type using DGPS/GIS," *Int. J. Smart Home*, vol. 13, no. 1, pp. 1–6, Apr. 2019.

[7] C.-Y. Chen and H.-J. Chien, "Geometric calibration of a multi-layer LiDAR system and image sensors using plane-based implicit laser parameters for textured 3-D depth reconstruction," *J. Vis. Commun. Image Represent.*, vol. 25, no. 4, pp. 659–669, May 2014.

[8] I. S. Weon and S. G. Lee, "Velocity-based object detection in dynamic environment using YOLO-based deep learning algorithm," *Int. J. Multimedia Ubiquitous Eng.*, vol. 14, no. 1, pp. 7–12, 2019.

[9] Z. Liu, S. Yu, X. Wang, and N. Zheng, "Detecting drivable area for self-driving cars: An unsupervised approach," 2017, *arXiv:1705.00451*. [Online]. Available: http://arxiv.org/abs/1705.00451

[10] O. Chum, J. Matas, and J. Kittler, "Locally optimized RANSAC," in *Proc. Joint Pattern Recognit. Symp.* Berlin, Germany: Springer, 2003, pp. 236–243.

[11] I.-S. Weon, S.-G. Lee, J.-Y. Jeong, and S.-C. Moon, "A study of gait intension by the IMU based on pedestrian yaw characteristic," in *Proc. KSPE Autumn Conf.*, 2016, pp. 89–90.

[12] I.-S. Weon, S.-G. Lee, and J.-K. Ryu, "Virtual bubble filtering based on heading angle and velocity for unmanned surface vehicle (USV)," in *Proc. 17th Int. Conf. Control, Autom. Syst. (ICCAS)*, Oct. 2017, pp. 1954–1958.

[13] S. Bai, F. Chen, and B. Englot, "Toward autonomous mapping and exploration for mobile robots through deep supervised learning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, vol. 24, no. 28, pp. 2379–2384.

[14] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P.-J. Nordlund, "Particle filters for positioning, navigation, and tracking," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 425–437, 2002.

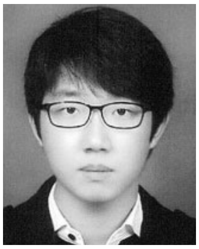
[15] F. M. Mirzaei, D. G. Kottas, and S. I. Roumeliotis, "3D LIDAR–camera intrinsic and extrinsic calibration: Identifiability and analytical least-squares-based initialization," *Int. J. Robot. Res.*, vol. 31, no. 4, pp. 452–467, Apr. 2012.

[16] R. Hashim, S.-I. Mohammad, and S. MigwaN, "Mosque tracking on mobile GPS and prayer times synchronization for unfamiliar area," *Int. J. Future Gener. Commun. Netw.*, vol. 4, no. 2, pp. 37–48, 2011.

[17] J. Zhang and S. Singh, "LOAM: Lidar odometry and mapping in real-time," *Robot., Sci. Syst.*, vol. 2, no. 9, pp. 1–9, 2014.

[18] S. Sengupta, D. C. Sarkar, S. Biswas, and P. Sarkar, "A novel approach to identify a fraud Website using Android smartphone under the collaborative frameworks of QR codes and GPS and motion parameters of the user," *Int. J. Secur. Appl.*, vol. 8, no. 5, pp. 161–184, Sep. 2014.

[19] I.-S. Weon, "Obstacle tracking based on deep learning and multi-sensor fusing for autonomous vehicle in the road," Ph.D. dissertation, School Eng., Kyung Hee Univ., Yongin, South Korea, 2019.



IHN-SIK WEON received the B.S. degree in mechanical engineering from the Engineering College, Korea University, South Korea, in 2014, and the M.S. and Ph.D. degrees in mechanical engineering from the Engineering College, Kyung Hee University. Since 2019, he has been with the Unmanned System, LIG Nex1 Co., Ltd., where he is currently an Assistance Researcher. His main research interests are the autonomous vehicle's control algorithm, unmanned surface vessel, and intelligent assistance robot.

SOON-GEUL LEE received the B.E. degree in mechanical engineering from Seoul National University, Seoul, South Korea, in 1983, the M.S. degree in production engineering from KAIST, Seoul, in 1985, and the Ph.D. degree in mechanical engineering from the University of Michigan, in 1993. Since 1996, he has been with the Department of Mechanical Engineering, Kyung Hee University, Yongin, South Korea, where he is currently a Professor. His research interests include robotics and automation, mechatronics, intelligent control, and biomechanics.

JAE-KWAN RYU received the B.E. degree in nuclear engineering and the M.S. degree in mechanical engineering from Kyung Hee University, South Korea, and the Ph.D. degree in robotic engineering from JAIST, Japan. Since 2009, he has been with the Unmanned System, LIG Nex1 Co., Ltd. His research interests include robotics and automation, mechatronics, intelligent control, and biomechanics.