IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 10 November 2022, accepted 30 November 2022, date of publication 19 December 2022,
date of current version 23 December 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3230463

RESEARCH ARTICLE

Efficient Camera–LiDAR Calibration Using Accumulated LiDAR Frames

DONGKYU LEE[1] AND SEOK-CHEOL KEE[2], (Member, IEEE)
[1]Department of Smart Car Engineering, Chungbuk National University, Cheongju 28644, South Korea
[2]Department of Intelligent Systems and Robotics, Chungbuk National University, Cheongju 28644, South Korea

Corresponding author: Seok-Cheol Kee (sckee@chungbuk.ac.kr)

This work was supported in part by the National Research Foundation of Korea (NRF) Grant through the Korean Government, Ministry of Science and ICT (MSIT), South Korea, under Grant 2022R1A5A8026986; and in part by the MSIT under the Grand Information Technology Research Center Support Program, supervised by the Institute for Information and Communications Technology Planning and Evaluation (IITP), under Grant IITP-2022-2020-0-01462.

ABSTRACT In autonomous driving, the camera and LiDAR are complementary to each other through their strengths, and many autonomous vehicles and robot recognition systems utilize cameras and LiDAR. In order to effectively fuse the data of the camera and LiDAR attached to different positions and angles, we must perform camera–LiDAR extrinsic calibration. Most existing camera–LiDAR calibration methods infer the results by constructing a camera–LiDAR feature pair using features acquired from a single frame. It has the disadvantage that it is challenging to draw results. In this paper, we used sequential LiDAR data to extract features by accumulating LiDAR frames effectively. By using the location information between the accumulated frame (global) and the single frame (local) to convert the features detected from the global to the local, the shortcomings of LiDAR's single frame with few features were supplemented. Methods for detecting feature points in LiDAR are described step-by-step, and the advantages and excellence of this camera–LiDAR calibration system are shown through quantitative/qualitative evaluation methods. As a result, the proposed system outputs superior performance compared to other systems.

INDEX TERMS Extrinsic calibration, frame accumulation, foot of perpendicular, plane modeling, image plane projection, clustering, C-track, tolerance compensation room, checkerboard.

I. INTRODUCTION

Interest in autonomous vehicles is steadily increasing, with city competition [1], [2], [3] occurring for autonomous vehicles and the operation of demonstration services [4]. It is growing, and it can be said that the future of the autonomous driving market is very bright. Autonomous vehicles and robots inevitably require sensors that recognize the surrounding environment, and cameras and LiDAR, as well as representative cognitive sensors, are widely used. The camera has the advantage of containing various texture information and the disadvantage of it being difficult to estimate the distance. LiDAR has the disadvantage of including relatively few textures, although it is easy to extract distances. When the camera and LiDAR are used together, they are used most commonly in the autonomous driving market, because they can complement the opposite's advantages/disadvantages.

In order to use the camera and LiDAR together, camera–LiDAR extrinsic calibration is an essential process. The resulting extrinsic parameters are information, including the relative position/angle between the camera and LiDAR in 3D space. Furthermore, users can convert each other's data into the camera and LiDAR coordinates based on this information. [5].

The process of acquiring extrinsic parameters can be expressed as a collection of numerous functions used in autonomous driving recognition. Functions such as feature extraction/matching share content, widely used in tasks such as object detection/tracking, are frequently encountered in autonomous driving research. Therefore, the extrinsic calibration system has a characteristic that requires many references to many existing autonomous driving studies and technologies.

The associate editor coordinating the review of this manuscript and approving it for publication was Joewono Widjaja.

In order to perform extrinsic calibration, several feature pairs extracted from both camera–LiDAR sensors are required. Unlike a texture-rich camera, LiDAR uses small-sized features (e.g., checkerboard, ArUco [6], and AprilTag [7], [8]), which have the disadvantage of being difficult to extract, and this becomes more severe with low-channel LiDAR. In low-channel LiDAR, the data between each channel becomes sparse, and the features between each channel are lost, resulting in the inability to extract the feature itself, and difficulty in forming a feature pair with the camera, which causes inefficient results. In addition, when the vehicle physically interferes with the view range of the sensors, such as when the LiDAR is located in front of the autonomous vehicle, and the camera is located in the rear of the autonomous vehicle (the view ranges of the sensors do not overlap at all or they overlap each other little bit), there is a case where it is not possible to create a camera–LiDAR feature pair. We studied a system that supplements such difficult cases through the accumulation of LiDAR frames and it extracts better results based on the same data by increasing the camera–LiDAR pairs.

Section 2 deals with the calibration methods used in autonomous driving, and the mathematical solutions used in the process. Section 3.A describes LiDAR frame accumulation and its process. In section 3.B, the method of selecting a good ROI (Region Of Interest) using clustering [9] from the accumulated frame and removing outliers through plane modeling [10] and the mathematical solution used in this process are discussed. Section 3.C describes how to extract features from the selected cluster. In section 3.D, we combine the features extracted from the accumulated map with the LiDAR localization data, and perform the projection into a single frame. Section 4 quantitatively/qualitatively evaluates and analyzes this calibration system. Finally, in section 5, this system is summarized, and the contents of the system's application plan and future research directions are dictated.

In this paper, we present some contribution points, and the list of contributions is as follows.

- Novel calibration method in an environment where camera-LiDAR calibration is difficult. LiDAR accumulation and localization overcome the environment where it is difficult to find matching feature points between the camera and LiDAR, such as low-channel or non-overlapped sensors.
- Filtering method to effectively extract feature points on accumulated LiDAR. The method of increasing the texture by accumulating several frames causes noise. To compensate for this, we propose a model-based filtering and foot-of-perpendicular filtering method to help extract key points.
- Proposed solution by projecting 3D data into 2D by taking advantage of the characteristics of a checkerboard modeled as a plane. When 2D image processing is enabled, general algorithms can be used and data can be easily utilized.

II. BACKGROUND

Various types and calibration methods exist within the field of autonomous vehicles or robots. It may be a calibration for distortion or coordinates issued due to a factor inside the sensor, a calibration between the sensor and another sensor, or a calibration between the sensor and the surrounding environment. Calibration using factors inside the sensor is called intrinsic calibration, and calibration between sensors and calibration between the sensor and the surrounding environment is called extrinsic calibration.

In the case of the camera and LiDAR, which are most commonly used in autonomous driving recognition, intrinsic calibration is essential because the light source emitted from the outside or the inside passes through a specific material. In the case of LiDAR, since the manufacturer performs intrinsic calibration [11] and outputs 3D sensor data based on the data, most users do not perform additional intrinsic calibration, and trust and use the output data of LiDAR itself. In the case of cameras, although many types are provided with intrinsic parameters through intrinsic calibration, the user should confirm whether the data are good, and some manufacturers provide only the camera itself. Especially, the camera's output is 2D image plane data, not 3D, and so most developers and researchers perform additional camera intrinsic calibration.

In conclusion, extrinsic calibration estimates the position and pose of the attached sensor. In order to acquire data on the real world in autonomous driving or in a system using sensors, we essentially perform extrinsic calibration. The type may be a calibration of the sensor and vehicle or the sensor and ground; or calibration between the same or different types of sensors. Typically, in the case of LKAS (Lane Keeping Assist System) [12], which is often installed in ordinary vehicles, extrinsic calibration between the camera and the ground where the vehicle is located is essential. The actual distance of the lane features extracted from the camera is estimated. Furthermore, calibration between sensors of the same type can be typically described as AVM (Around View Monitoring) [13] using BEV (Bird's Eye View) [14] with four wide-angle cameras (generally 180 degrees or more, of a field of view). Moreover, research using AVM [15], [16], [17] is being actively conducted.

The most widely used extrinsic calibration in autonomous driving systems is camera–LiDAR calibration. It can be used as a "LiDAR to camera projection" using high-texture camera features and exact 3D LiDAR distance, and it is used in applications such as the "3D reconstruction" of data [18], [19].

This section describes camera intrinsic calibration and extrinsic calibration for camera–LiDAR and AVM.

A. INTRINSIC CALIBRATION

Camera intrinsic calibration can be divided into two main categories. The first involves correcting the lens distortion of the camera, and the second involves obtaining a camera matrix, including focal length and principal length. Both

of these processes are for the correction of distortion and the coordinate system caused by factors inside the camera, and include the types of camera lenses and errors in the manufacturing process.

The camera matrix is in Eq. (1), and includes the focal length, principal length, and skew coefficient. Nowadays, the camera manufacturing process has a higher level of precision and completeness compared to the past, so that it can assume that the skew coefficient representing the asymmetry of the camera image sensor is 0. Look at the camera matrix except for skew. It is a transformation matrix that enlarges/reduces the homogenous coordinates of the two-dimensional plane point and moves them in parallel. It can be seen that the camera matrix is a transformation matrix between the image plane originating from the left-top and the normalized plane originating from the center of the camera image.

The camera utilizes an external light source (sunlight, ambient light, etc.). The object's light is projected onto an image sensor through a lens. This can adjust the camera's angle of view using the lens, which also causes lens distortion. We must correct the lens distortion in order to use the real-world data. In order to eliminate lens distortion, users use a method to mathematically model and utilize a lens according to the shape of the lens. This section shows three lens distortion models based on the Brown–Conrady model [20], [21], perspective, equidistance [22], and catadioptric [23], [24], and we use these three models in testing this paper. (It was used with OpenCV API [25], [26], [27].) The perspective model is the camera distortion model that we use the most, and is mainly used for narrow camera calibration. The equidistance and catadioptric models are modeled using a fisheye or mirror-based lens, and are used for wide camera calibration. In this paper, we used the perspective model for narrow camera–LiDAR calibration, the catadioptric model for wide camera–LiDAR calibration, LiDAR-to-camera fusion, and the equidistance model for AVM.

In performing an extrinsic calibration after intrinsic calibration, the calibration is performed based on the normalized undistorted plane whose origin is located at the center of the camera image.

$$C = \begin{bmatrix} f_x & skew & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}, \textit{Camera Matrix} \quad (1)$$

$$\mathrm{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}, \textit{Rotation Matrix} \quad (2)$$

$$T = \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix}, \textit{Translation Matrix} \quad (3)$$

$$\begin{bmatrix} u_{Undistored} \\ v_{Undistored} \\ 1 \end{bmatrix} = C \begin{bmatrix} X_{Real} \\ Y_{Real} \\ Z_{Real} \\ 1 \end{bmatrix} \quad (4)$$

B. EXTRINSIC CALIBRATION

The ultimate reason for using sensors in autonomous driving solutions boils down to one thing: to detect objects. We want to make the ego vehicle detect and avoid objects, by attaching a sensor that acts as a human eye or skin to an autonomous vehicle or robot. It is necessary not only to determine the existence of an object, but also to classify the object and accurately estimate the position of the object. Extrinsic calibration estimates the position and poses of the sensor attached to the ego vehicle. Then, we can estimate the object's position from the reference coordinates, and the camera system can project the 3D point on the coordinate system to the undistorted image plane. (Eq. (1)∼(4))

We mainly use the rotation matrix and translation matrix to express extrinsic calibration data, and also to use homography for testing through AVM. The rotation matrix is a 3×3 matrix, including pose information from the reference coordinates. The translation matrix is a 3×1 matrix, including position information from the reference coordinates. Additionally, homography is a 3×3 matrix used as a transformation equation between the normalized undistorted coordinates and the AVM coordinates, projected as Z=k (k is an arbitrary constant) (Eq. (5)∼(6)). This can be inferred by combining a camera matrix and a translation matrix, and a rotation matrix. In addition, there is a method to obtain it directly from point to point without going through this process (Eq. (7)∼(8)).

$$\mathrm{H} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}, \textit{Homography} \quad (5)$$

$$\begin{bmatrix} u_{Undistorted} \\ v_{Undistorted} \\ 1 \end{bmatrix} = H \begin{bmatrix} X_{AVM} \\ Y_{AVM} \\ 1 \end{bmatrix} \quad (6)$$

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = C_{toAVM} C \begin{bmatrix} r_{11} & r_{12} & \mathbf{r_{13}} & t_1 \\ r_{21} & r_{22} & \mathbf{r_{23}} & t_2 \\ r_{31} & r_{32} & \mathbf{r_{33}} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ \mathbf{Z} \\ 1 \end{bmatrix} \quad (7)$$

$$H \cong C_{AVM} C \begin{bmatrix} r_{11} & r_{12} & t_1 \\ r_{21} & r_{22} & t_2 \\ r_{31} & r_{32} & t_3 \end{bmatrix} \quad (8)$$

The process of finding the rotation matrix and translation matrix is performed through PnP (Perspective and Points) [28]. Prepare many pairs of 2D points on the camera image and 3D points on the 3D real world (or LiDAR), perform intrinsic calibration in advance, know the camera matrix, and perform PnP with undistorted camera points. We can estimate the rotation matrix and the translation matrix.

During this study, we referred to Levenberg-Marquardt optimization [29], P3P [30], EPnP [31], [32], DLS [33], UPnP [34], IPPE [35], AP3P [36], and SQPnP [37]. SQPnp did not diverge, even with a relatively small amount of corresponding pairs, however Levenberg-Marquardt optimization outputs best result, when there are enough data. We can

obtain fine extrinsic parameters with Levenberg-Marquardt optimization.

C. CONSIDERED METHODS

Research related to intrinsic calibration and extrinsic calibration has been continuously conducted, and related research and implementation software are being released. However, most of the software related to extrinsic calibration depends on the user's environment, so it does not work or the performance does not come out as the developer says when the environment or sensor is different. We have run some calibration papers and software and briefly described them.

Zhou et al. [38] is the most commonly used camera-LiDAR calibration method, and it is a method of estimating the checkerboard based on the detected feature points. Basically, low-channel LiDAR, which lacks feature points to extract feature points from a single-shot, has relatively poor performance compared to high-channel LiDAR.

Yuan et al. [39] is a calibration method using infrastructure rather than checkerboard, and performs calibration by matching the edge extracted from the camera and LiDAR. Due to extracting edge from LiDAR, dense point cloud data is required like high-channel LiDAR, so we used the proposed point of this paper, the accumulated cloud, to create a dense point cloud for executing the algorithm. Experimentally, a scene with a lot of planes and corners in the camera area outputs better results than an outdoor scene with a variety of bushy textures. Algorithms that perform calibration using infrastructure even require a denser point cloud and much depend on environments.

Papers related to corner detectors that detect feature points on the camera are as follows. Geiger et al. [40] was performed using KITTI [41] indoor calibration data, and a single-shot camera and 64-channel LiDAR were used. Geiger et al. propose an excellent corner detector, and in this paper, we use the corner detector to detect feature points. Cviši'c et al. [42] proposes a method of classifying checkerboards through horizontal/vertical and positive/negative edges and detecting corners through contact points between different types of edges. Based on the index evaluated in the paper, it can be said to be the best corner detector. In the case of the above two papers, calibration was carried out based on KITTI, and since KITTI provides stationary calibration data, our paper has the disadvantage that it is difficult to evaluate with KITTI, so comparing the results is difficult.

At Chungbuk National University, there is a test track called C-track [43], [44], and inside the C-track, there is a tolerance compensation room. Fig. 1 shows the tolerance correction room. The tolerance compensation room has a flat floor and consists of a regular grid in the form of a checkerboard. We used to use the tolerance compensation room for calibration and making AVM. We extract the camera features from corner detectors such as Geiger et al., extract the LiDAR map feature from handcraft, match the camera-LiDAR pair and find out the calibration results.

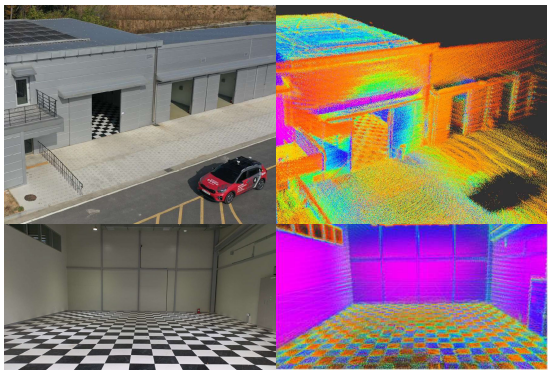

FIGURE 1. C-track tolerance compensation room.

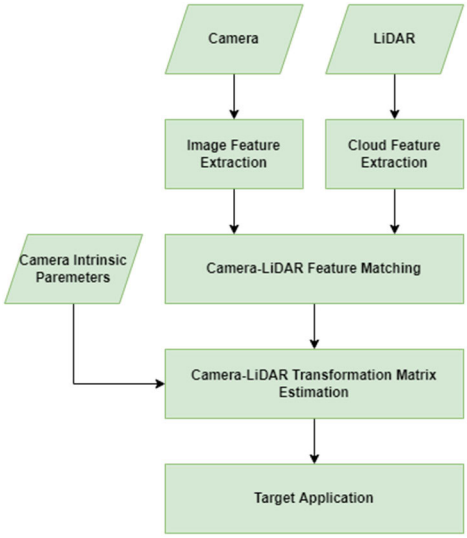

FIGURE 2. General camera-lidar calibration process.

III. PROPOSED METHOD

General camera–LiDAR calibration extracts extrinsic parameters based on a pair of matching features extracted from a single camera frame and a single frame of LiDAR. This paper uses LiDAR frame accumulation and LiDAR-based localization (an estimation of the conversion formula between the accumulated LiDAR data and a single frame) to solve harrowing feature extractions such as low-channel LiDAR, or the problem where the view ranges of two sensors do not overlap. We augment the LiDAR features and increase the number of camera–LiDAR feature matching pairs.

Fig. 2 shows the existing general camera–LiDAR extrinsic calibration process, and Fig. 3 shows the process proposed in this paper. Darker blocks are added in this paper, and the darker block signifies our proposed solution. In LiDAR, a sequential frame rather than a single frame is the input, and the accumulated map (global) is produced by multiple LiDAR frames. The transformation matrix between the accumulated map and a single frame (local) is estimated through the point cloud registration algorithm (NDT [45] or GICP [46]),

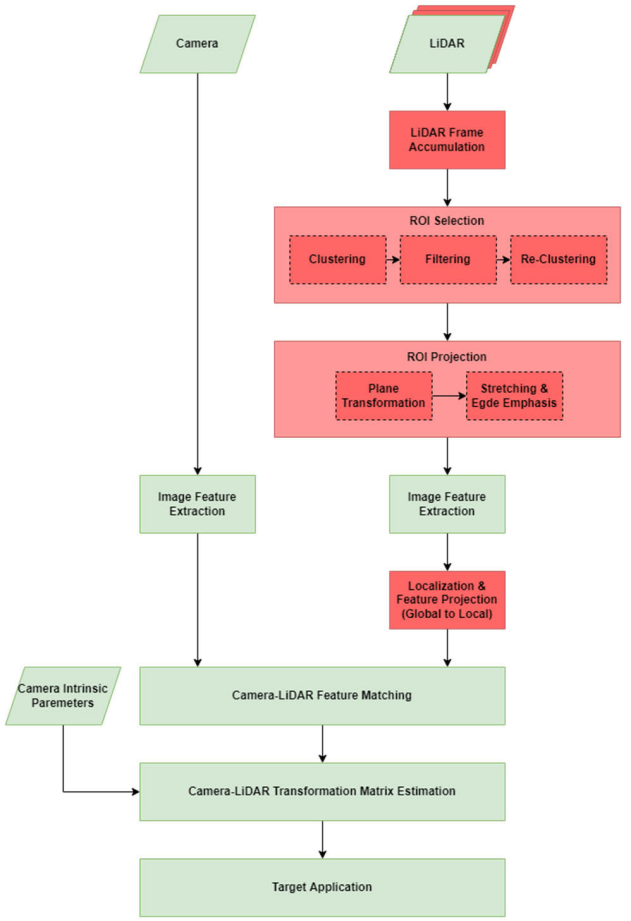

FIGURE 3. Proposed camera-LiDAR calibration process.

FIGURE 4. Less target points from LiDAR.

and features are extracted from the global. We use the localization and global features to obtain local frame features simultaneously.

A. LIDAR FRAME ACCUMULATION

In the case of a single LiDAR frame, it is often difficult to extract features for calibration, due to the lack of textures in the data. Although it is possible to infer features by using the characteristics of a planar-based board or a specific pattern, in this paper, a method of accumulating LiDAR frames is used to extract more accurate features.

We can find difficult cases to perform a calibration between single frame camera and LiDAR. As shown in Fig. 4, the number of features may be small because the channel of the LiDAR raw data is not significant. As shown in Fig. 5,

FIGURE 5. Non-overlapped case between camera and LiDAR.

LiDAR is attached at an angle to the front, and the camera is attached to the rear. In that case, it cannot perform calibration using single frame LiDAR. However, in the proposed camera–LiDAR extrinsic calibration method of the map accumulation, we can perform calibration at non-overlapped cases if the camera's target calibration feature is seen. LiDAR moves and collects the target, so camera–LiDAR calibration can be performed even if the view ranges between the sensors do not overlap.

Before accumulating LiDAR, we unified the coordinate systems of GNSS, INS, and LiDAR to reduce error between sensors used for position recognition during accumulation. Separately from the proposed method, the LiDAR trajectory result and the GNSS and INS-based trajectory were compared in the generated infrastructure SLAM map, and finally, the GNSS and INS coordinate systems were projected into the LiDAR coordinate system and utilized.

The data required to accumulate the map are as follows. Point cloud data are the primary data, and GNSS and INS data are used as initial values when performing point cloud registration (NDT or GICP). Let the first LiDAR frame be a reference frame, and a transformation matrix with other frames is found and accumulated to create a map (Eq. (9)).

Most of the raw GNSS data output longitude, latitude, and altitude, based on the WGS84 coordinate system. In order to match the LiDAR and the coordinate system, GNSS data is converted to the X-, Y-, and Z-based UTM coordinate system (unit: m) [47], and then used as a 3×1 type translation matrix. The INS orientation information as quaternion is converted to a 3×3 type rotation. Reference [48] concatenate the obtained rotation matrix and translation matrix; a transformation matrix in the form of 4×4 is formed; point cloud registration is performed using this as an initial value, and the point cloud is accumulated with the obtained correction value. Fig. 6 shows the cumulative result performed in different scenes, and the green arrow is a schematic representation

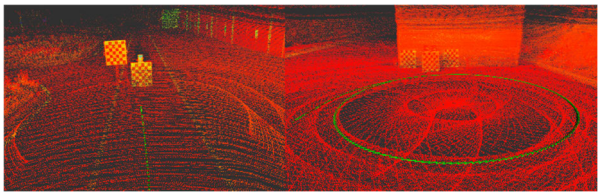

FIGURE 6. Example of accumulated frame.

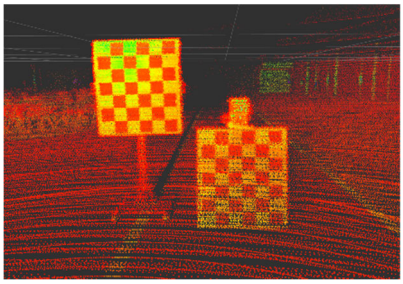

FIGURE 7. Accumulated checkerboard.

of the transformation matrix corrected for each frame, including the location and direction information for multiple frames.

$$\begin{bmatrix} X_{local} \\ Y_{local} \\ Z_{local} \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_{global} \\ Y_{global} \\ Z_{global} \\ 1 \end{bmatrix} \tag{9}$$

B. ROI SELECTION

Looking at Fig. 7, the accumulated checkerboard has a shape so that features can be easily identified with the naked eye. Unlike a single frame point cloud, the accumulated point cloud map contains various textures. However, since the accumulated point cloud map includes numerous objects except for the checkerboard used for extrinsic calibration, we performed several processes to select only the area of the checkerboard that we want from among these numerous data, and designate it as ROI. The process is as follows.

1. Clustering,
2. Filtering,
3. Re-clustering.

After the three processes are completed, do the section C procedure using the cluster selected by the user from among the determined cluster candidates.

1) CLUSTERING

We clustered checkerboards using Euclidean clustering. Looking at the point cloud data accumulated through Fig. 6, it can be found that the data's ground surface is sparser than the checkerboard. Based on these points, the checkerboard has a denser cluster. We can handle the hyper-parameters of Euclidean clustering by the dense property.

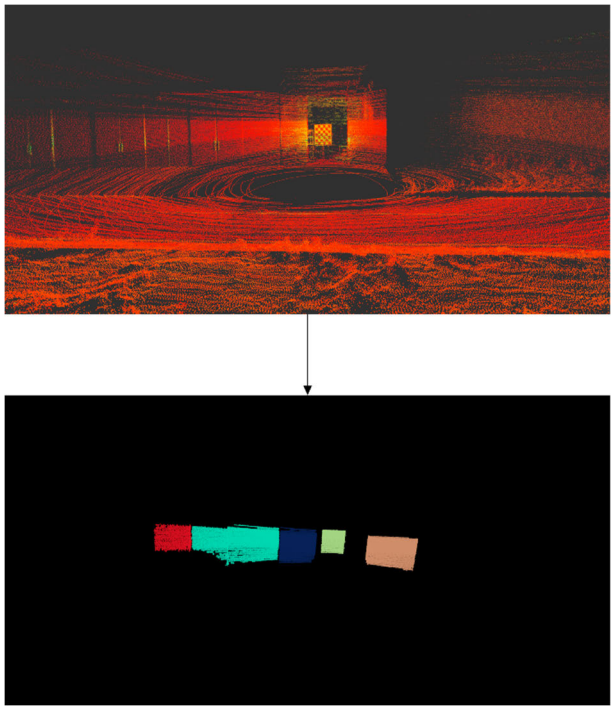

FIGURE 8. Cluster extraction from accumulated frame.

Euclidean clustering was performed using EuclideanClusterExtraction of the PCL library. As in Fig. 8, clusters with sufficient clusters were extracted. In the clustering situation, unnecessary clusters were removed in the bottom region with a shallow degree of clustering, and in the region farther from the center of the accumulated point cloud map.

2) FILTERING

Although a cluster with high clustering was extracted through the clustering operation, extreme outliers were mixed in the cluster, such as the noise of the sensor itself and the noise generated during the accumulation stage. In order to remove this noise, we performed two noise removal methods. We obtained a three-dimensional planar equation by utilizing the property of a plane recognized as a precondition.

First, a method through RANSAC [49], traditionally used in distinguishing inliers and outliers, was utilized. In this study, we extracted features through the checkerboard, and the checkerboard has a flat feature. At the same time, using RANSAC, we separated inliers and outliers, based on the planar equation, to remove relatively large noise.

Second, the cluster from which the significant noise has been removed was projected onto the planar model through the foot of the perpendicular. The relatively small noise features that had not been removed through the planar model and RANSAC were filtered. Following the procedure in Fig. 9 and Eq. (10)-(19), the foot of perpendicular was induced through the standard vector component of the planar

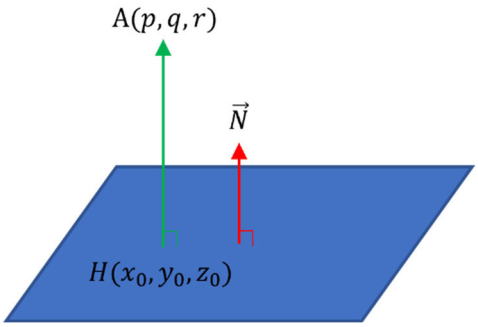

FIGURE 9. Foot of perpendicular, A→H.

model, and the points of slight noise were projected onto the plane and removed simultaneously.

$$ax + by + cz + d = 0 \tag{10}$$

$$\vec{N} = (a, b, c)\,, \textit{Plane Normal Vector} \tag{11}$$

$$\vec{OA} = (p, q, r)\,, \ (\textit{Input}) \tag{12}$$

$$\vec{OH} = (x_0, y_0, z_0)\,, \ (\textit{Output}) \tag{13}$$

$$\vec{AH} = k\vec{N} = \vec{OH} - \vec{OA}, \textit{where } \vec{AH} // \vec{N} \tag{14}$$

$$(x_0 - p, y_0 - q, z_0 - r) = k\,(a, b, c) \tag{15}$$

$$ax + by + cz + d = 0 \begin{cases} x_0 = p + ka \\ y_0 = q + kb \\ z_0 = r + kc \end{cases} \tag{16}$$

$$a\,(p + ka) + b\,(q + kb) + c\,(r + kc) + d = 0 \tag{17}$$

$$k = \frac{-ap - bq - cr - d}{\sqrt{a^2 + b^2 + c^2}} \tag{18}$$

$$\vec{OH} = k\vec{N} + \vec{OA} \tag{19}$$

3) RE-CLUSTERING

Through clustering and filtering, a lot of unnecessary data were purified, and the point counts of each cluster were reduced. However, there were still unnecessary data mixed with planar data in each cluster, and to refine the data, clustering and planar model-based filtering were performed again.

The reduced points while performing Section 3.B.1 and Section 3.B.2 reduced the boundary between the plane and the non-plane part. When clustering and planar model-based filtering are performed, the result of the boundary is output more clearly. In Fig. 10, we compared before and after re-clustering. In the case of the former, non-plane objects are clustered together at the top of the checkerboard to the right, whereas in the latter case, the upper object is filtered, and only the plane remains to form a cluster.

C. ROI PROJECTION

The point cloud data that went through the ROI selection process included checkerboard data for each cluster. We needed to extract the edge features of the checkboard from each cluster. If the degree of clustering of the point cloud is sparse, planar modeling and other techniques should be used in the detection stage. However, the point cloud cluster we obtained were data with dense clustering and prominent features.

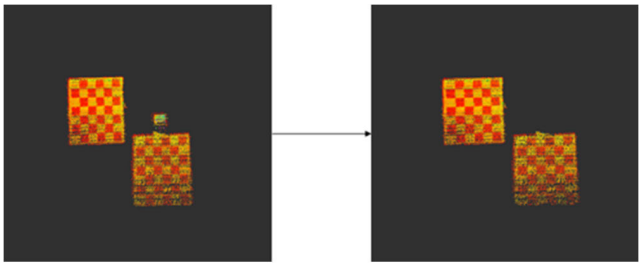

FIGURE 10. Re-clustering result. left: before n right: after.

We aimed to reduce prediction and extract good features from the data we had.

Since we know that the data that have undergone ROI selection are data located on a specific plane, we converted the data of this plane into an image plane with the left-top equal to 0. Then, we used the method from Geiger et al., one of the traditional calibration (intrinsic or extrinsic) feature detectors. Although data with a dense clustering are good data at the LiDAR level, aliasing occurs frequently at the image level, and some preliminary work is usually required to use an image processing-based detector. The process is as follows.

- Plane transformation,
- Stretching and interpolation, and edge emphasis.

1) PLANE TRANSFORMATION

We already know the 3D planar equation in the form of "ax + by + cz + d = 0" for each cluster, and we know that the 3D planar equation of the image plane, which is the x–y plane, is "z = 0". A rotation and translation matrix is obtained through Eq. (20)-(25), and the 3D plane data are converted into a 2D image plane.

$$\left\|\vec{X}\right\|_2 = \sqrt{\left(\vec{X}\,(1)\right)^2 + \left(\vec{X}\,(2)\right)^2 + \left(\vec{X}\,(3)\right)^2}, \tag{20}$$

where X represents unknown variable

$$\vec{C} = \vec{N_1} \times \vec{N_2}, \tag{21}$$

where $\vec{N_1} = (a, b, c)$ *and* $\vec{N_2} = (0, 0, 1)$

$$D = \vec{N_1}^{\circ}\vec{N_2}, \tag{22}$$

where $\vec{N_1} = (a, b, c)$ *and* $\vec{N_2} = (0, 0, 1)$

$$\vec{S} = \begin{bmatrix} 0 & -\vec{C}\,(3) & \vec{C}\,(2) \\ \vec{C}\,(3) & 0 & -\vec{C}\,(1) \\ \vec{C}\,(2) & \vec{C}\,(1) & 0 \end{bmatrix} \tag{23}$$

$$\vec{R} = \begin{bmatrix} 1\,0\,0 \\ 0\,1\,0 \\ 0\,0\,1 \end{bmatrix} + \vec{S} + \left(\vec{S}\right)^2 \frac{1 - D}{\left(\left\|\vec{C}\right\|_2\right)^2} \tag{24}$$

$$\vec{T} = \begin{bmatrix} 0 \\ 0 \\ d \end{bmatrix} \tag{25}$$

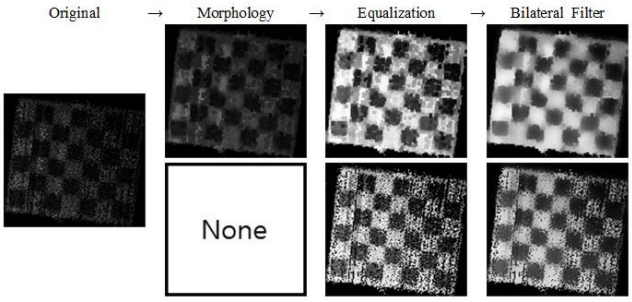

FIGURE 11. Image process of LiDAR feature points.

2) STRETCHING AND EDGE EMPHASIS

Most checkerboard-based image feature detectors extract features based on 8-bit single-channel images. In order to effectively utilize these characteristics, we processed data similar to a camera image by targeting an 8-bit single channel of the LiDAR feature data converted to the image plane.

Data processing proceeds in parallel in two ways. One is to utilize morphology transformation [50], and the other is to utilize the original. The method that utilizes morphology involves performing a close morphology operation, expanding and eroding the image data, and creating a base image by supplementing excessive aliasing, a characteristic of a point cloud-based image, without significantly touching the contour of the feature. The method that does not utilize morphology uses the original image converted from the point cloud as the base image. Based on the two base images, the image's dynamic range is stretched through the histogram-equalization [51] process, the image is smoothed between the edges using a bilateral-filter [52], and the edge of the feature is emphasized simultaneously (Fig. 11).

In Fig. 12, the checkerboard feature extraction result is based on the two previously generated images. The detected results are different, depending on whether morphology is applied.

D. LOCALIZATION AND FEATURE PROJECTION

The accumulated point cloud map is referred to as global, and a single LiDAR frame is referred to as local, and we detected checkerboard features in the global coordinates. In order to finally obtain the camera–LiDAR extrinsic calibration parameters we want, many feature coordinate pairs over the camera–LiDAR are required. Therefore, we need to transform the global feature into a local feature.

The point cloud registration algorithm (NDT or GICP) was used to find the position transformation between global and local. The transformation matrix obtained through this plays a role in converting global coordinates into local coordinates. Fig. 13 is a visualization of the features extracted from global and converted to local.

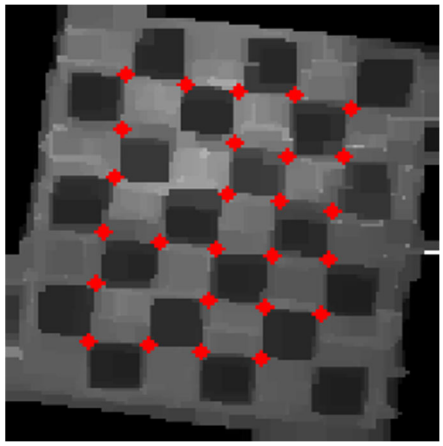

FIGURE 12. Extracted LiDAR feature as image.

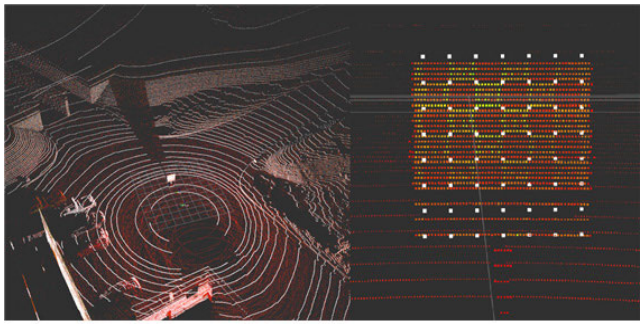

FIGURE 13. Transformed features(global to local) with localization.

E. TOTAL ALGORITHM

As in Algorithm 1, the overall algorithm extracts the rotation matrix and translation matrix by performing PnP with corner feature pairs extracted from the camera and LiDAR, camera matrix, and distortion coefficients as inputs.

Algorithm 2 and Algorithm 3 describe how to extract corner features from the camera and LiDAR, respectively.

Algorithm 1 Extrinsic Parameters Estimation

$corners_c$ [] : $CornerfromCamera(Array)$
$corners_l$ [] : $CornerfromLiDAR(Array)$
C : $CameraMatrix$
D : $CameraDistortionCoefficients$
R : $RotationMatrix$
T : $TranslationMatrix$
$[R, T] = solvePnP(corners_c[], corners_l[], C, D)$
$return[R, T]$

IV. EXPERIMENT

Many autonomous vehicles are equipped with multiple cameras. It is common to attach multiple cameras with different Fields of View (FoV), such as a narrow-angle camera for looking at medium and long distances, and a wide-angle camera for looking at close range. For this study, we experimented with two 60 degree narrow cameras (front

Algorithm 2 Camera Feature Extraction

F_{c_k} : *kth Camera Frame*
while $0 < i <$ *Accumulation Frame Size* ***do***
  $corner_c = cornerDetector(F_{c_i})$
  $corners_c[] \leftarrow corner_c$
end while
$returnconers_c[]$

Algorithm 3 LiDAR Feature Extraction

F_{l_k} : *kth LiDAR Frame*
$T_k : F_{l\_k} \leftrightarrow F_{l\_0} Transformation$
while $0 < i <$ *Accumulation FrameSize* ***do***
  $L_{accum} += T_i(F_{l\_i})$
end while
while $0 < i <$ *clustring* (L_{accum}) *.size* ***do***
  $filtered_1 = filtering_{model}(clustering\ (L_{accum})\ [i])$
  $filtered_2 = filtering_{foot}(filtered_1)$
  $reclutered = clustring(filtered_2)$
  $clusters\ [] \leftarrow reclutered$
end while
$cluster_{3d} = selectByHands(clusters\ [])$
$cluster_{2d} = 3dTo2d(cluster_{3d})$
$corner_{2d} = cornerDetector\ (cluster_{2d})$
$corner_{globalL} = reprojectionTo3d(corner_{2d})$
while $0 < i <$ *Accumulation FrameSize* ***do***
  $corner_l = projectLocalizatoin\left(corner_{globalL}, L_{accum}, F_{l\_i}\right)$
  $corners_l[] \leftarrow corner_l$
end while
return $corners_l[]$

TABLE 1. Implementation Detail.

Program Language	C++
Utilization Library	OpenCV / PCL / ROS
Time Synchronization Tool	NTP
Experimental Vehicle	KIA Niro EV
Camera Model	Sekonix SF3325-100 (60 degree) × 2 Sekonix SF3326-100 (190 degree) × 4
LiDAR Model	Hesai Pandar-40M × 1 Velodyne VLP-16 × 1
GNSS/INS Model	Novatel PwrPak7-E2 × 1
Checkerboard Spec	Width: 1000mm Height: 1100mm Square Size: 150mm × 150mm

and rear), 190 degree wide cameras (front, rear, left, and right), one LiDAR (center), and a vehicle with GNSS and IMU attached. The entire system was evaluated with these sensors. Table 1 shows what environment detail we use for implementation.

We performed extrinsic calibration between six cameras and LiDAR, using an experimental vehicle. Fig. 14 shows the

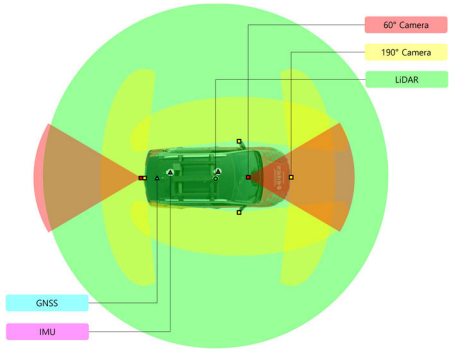

FIGURE 14. Sensor FoV.

FIGURE 15. C-track top view and HDMap.

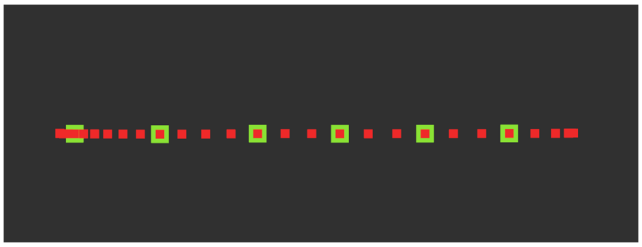

FIGURE 16. LiDAR frame sampling, red: collected n green: sampled.

position and FoV of the sensor installed in the test vehicle. The experiment used the calibration infrastructure and road of C-track, located at Chungbuk National University Ochang Campus. All test was performed at C-track(Fig. 15) and Chungbuk National University Gaesin Campus.

A. QUANTITATIVE EVALUATION

LiDAR generally operates at 10Hz, and the vehicle moves 2m per second when driving speed of 7.2kph. If the vehicle is driven slower, the number of frames acquired per distance increases, which causes unnecessary noise, so LiDAR data was sampled in units of 0.2 m. Fig. 16 shows the estimated position from the original LiDAR frame and the estimated position from sampled LiDAR frame.

We performed an evaluation for each total accumulation distance. from 1m to 10m, A total of 10 candidates were evaluated and the accumulation distance (n)m means accumulating (5n+1) LiDAR frames. Table 2 shows the evaluation results for each accumulation distance. In this

TABLE 2. Front narrow calibration result by accumulation distance.

Accumulation Distance	1m	2m	3m	4m	5m	6m	7m	8m	9m	10m
RMS (Levenberg-Marquardt optimization with Pandar-40M)	2.0447	1.4423	2.1048	0.8744	9.8569	**0.7486**	0.8524	0.8858	1.1181	2.3742
RMS (SQPnP with Pandar-40M)	2.0486	1.4673	2.1902	0.8778	9.8700	**0.7619**	0.8670	0.8855	1.1255	2.4239
RMS (Levenberg-Marquardt optimization with VLP-16)	9.8186	29.1425	1.1830	1.6346	1.3144	1.4606	**1.2234**	1.6054	1.5321	1.7185
RMS (SQPnP with VLP-16)	9.8416	29.1280	1.8517	1.6444	1.3259	1.5179	**1.2349**	1.6563	1.7504	1.8837

TABLE 3. Front narrow calibration result by PnP method.

PnP Method	Levenberg-Marquardt optimization	EPnP	IPPE	SQPnP
RMSE (Accumulation Distance: 6m with Pandar-40M)	**0.7486**	0.7631	0.9667	0.7619
RMSE (Accumulation Distance: 7m with VLP-16)	**1.2234**	1.2564	1.3507	1.2349

TABLE 4. Compared front narrow calibration result.

	Ours	Zhou *et al.*	Yuan *et al.*
Pandar-40m[1]	**0.7486**	4.8632	4.4608
VLP-16[2]	**1.2234**	12.2372	11.9356
Magnification[2÷1]	× **1.6343**	× 2.5163	× 2.6757

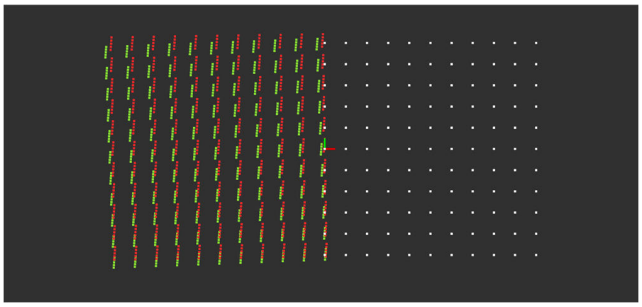

FIGURE 17. Front to rear points, red: reference n green: compared.

process, the PnP algorithm was evaluated heuristically using Levenberg-Marquardt optimization and SQPnP that does not diverge even in a small sample. Performance improves as the cumulative distance increases, but performance deteriorates when the critical point is crossed. Pandar shows the best performance at 6m and VLP at 7m.

We use PnP to obtain the Rotation/Translation Matrix by utilizing coupled camera-LiDAR features. There are several PnP methods, we performed four PnP methods with the highest performance accumulation distance for each lidar. The experiment in this paper results in Levenberg-Marquardt optimization was the best because feature points have increased enough due to accumulation, and followed by SQPnP. Table 3 shows the result.

For the narrow front camera, our proposed system was compared with Zhou et al. and Yuan et al. using an RMS (Root Mean Square)-based reprojection error (unit: pixel) [53]. We compared and analyzed the case using 40-channel and 16-channel LiDAR, by attaching Hesai Pandar-40m and Velodyne VLP-16. When performing calibration using Pandar-40m data, better performance was achieved, compared to Zhou et al. and Yuan et al.. In particular, when using VLP-16 with Zhou et al. and Yuan et al., the performance significantly decreased (vs. Pandar-40m) (Table 4). From Table 4, we can demonstrate our method is better than the estimate checkerboard method or calibration with infrastructure.

In the case of a wide camera, a different lens distortion model must be applied to the narrow camera. Since there are not many calibration-related solutions or academic data between the wide camera and LiDAR, the proposed system cannot be compared with other solutions or programs. So, we performed a comparison using calibration infra-difference.

The results were compared via calibration through the checkerboard and the tolerance compensation room.

Table 5 shows the reprojection error (RMS) for four wide and two narrow cameras. The results of the 190 degree wide cameras show that calibration using the checkerboard yields better results than the method through the tolerance compensation room. When manufacturing an AVM without LiDAR attached, calibration through a tolerance

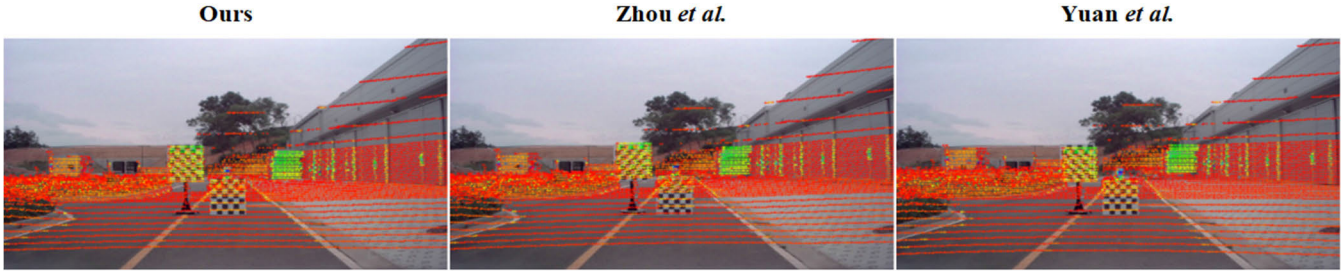

FIGURE 18. Front narrow camera calibration result with pandar-40M.

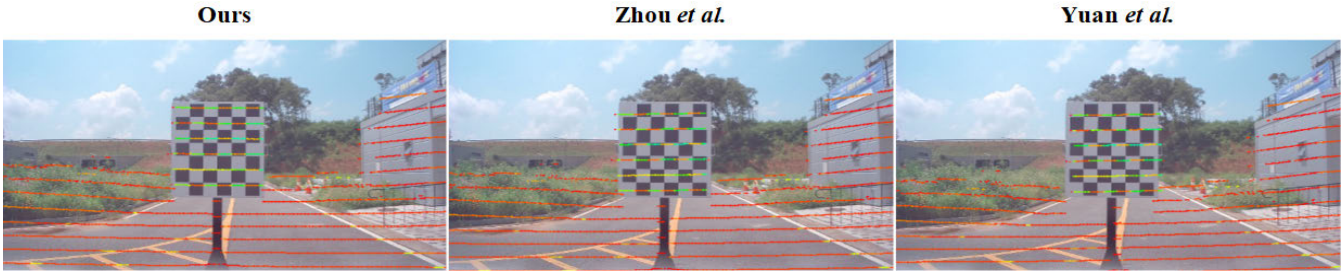

FIGURE 19. Front narrow camera calibration result with VLP-16.

TABLE 5. Calibration results according to target type.

FoV	Mount Location (on Vehicle)	Target	
		Checker board	Tolerance Compensation Room
190degrees	Front	**0.7796**	5.6311
190degrees	Rear	**0.9008**	3.3291
190degrees	Left	**2.6872**	11.2758
190degrees	Right	**2.8595**	11.7869
60degrees	Front	**1.3208**	-
60degrees	Rear	**1.4865**	-

compensation room is suitable for clarifying the reference coordinate system. However, when LiDAR is attached, the reference coordinate system becomes evident, and we can obtain a relatively accurate target coordinate system, and relative rotation/translation information can be obtained. This experiment shows that extrinsic parameters can be estimated based on the checkerboard in the camera–LiDAR system. AVM can be fabricated through this experiment, as well.

In the non-overlapped case, it is not possible to proceed with the evaluation by projecting the LiDAR data to the camera, and also, since there are no feature points in the overlapping area, the reprojection error cannot be utilized. So, we propose a new evaluation method.

In the case of non-overlapped cases, the indirect evaluation is performed by conversion between the LiDAR overlapped camera and the non-overlapped camera. In the case of the Pandar-40M we used, it overlaps with both the front and rear cameras, so the Pandar-to-front and Pandar-to-rear data can be trusted. In the case of VLP-16, VLP to the front is overlapped and VLP to the rear is in a non-overlap state. Through this, the front-to-rear result($(\boldsymbol{R}|\boldsymbol{T})_{compared}$) from VLP is inferred and compared with the Pandar-based front-to-rear result($(\boldsymbol{R}|\boldsymbol{T})_{reference}$). Finally, A total of 5 measures were used.

The first measure is Point Projection RMS, which creates virtual coordinates in the front, then converts the coordinates using $(\boldsymbol{R}|\boldsymbol{T})_{reference}$ and $(\boldsymbol{R}|\boldsymbol{T})_{compared}$, and then calculates the RMS for all paired points. Fig. 17 shows the generated virtual coordinates and two cases transformed from them.

The second measure is Quaternion Error. After converting $\boldsymbol{R}_{reference}$ and $\boldsymbol{R}_{compared}$ to quaternion in generated $(\boldsymbol{R}|\boldsymbol{T})_{reference}$ and $(\boldsymbol{R}|\boldsymbol{T})_{compared}$, the difference in direction between the two vectors is designated as an error. Eq. (26)-(27) shows the error calculation process, and the closer the error is to 0, the more the direction is similar.

$$q = (w, x, y, z)\,, quaternion \tag{26}$$

$$QuaternionError = 1 - q_1 \cdot q_2 \tag{27}$$

The remaining three methods utilize $\boldsymbol{T}_{reference}$ and $\boldsymbol{T}_{compared}$ in generated $(\boldsymbol{R}|\boldsymbol{T})_{reference}$ and $(\boldsymbol{R}|\boldsymbol{T})_{compared}$. The components for the X, Y, and Z coordinates are compared respectively and used as an error. Eq. (28)-(31) shows the operation process.

Table 6 shows all evaluated measures.

$$T = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{28}$$

$$X\ Error = abs(T_1[0] - T_2[0]) \tag{29}$$

$$Y\ Error = abs(T_1[1] - T_2[1]) \tag{30}$$

$$Z\ Error = abs(T_1[2] - T_2[2]) \tag{31}$$

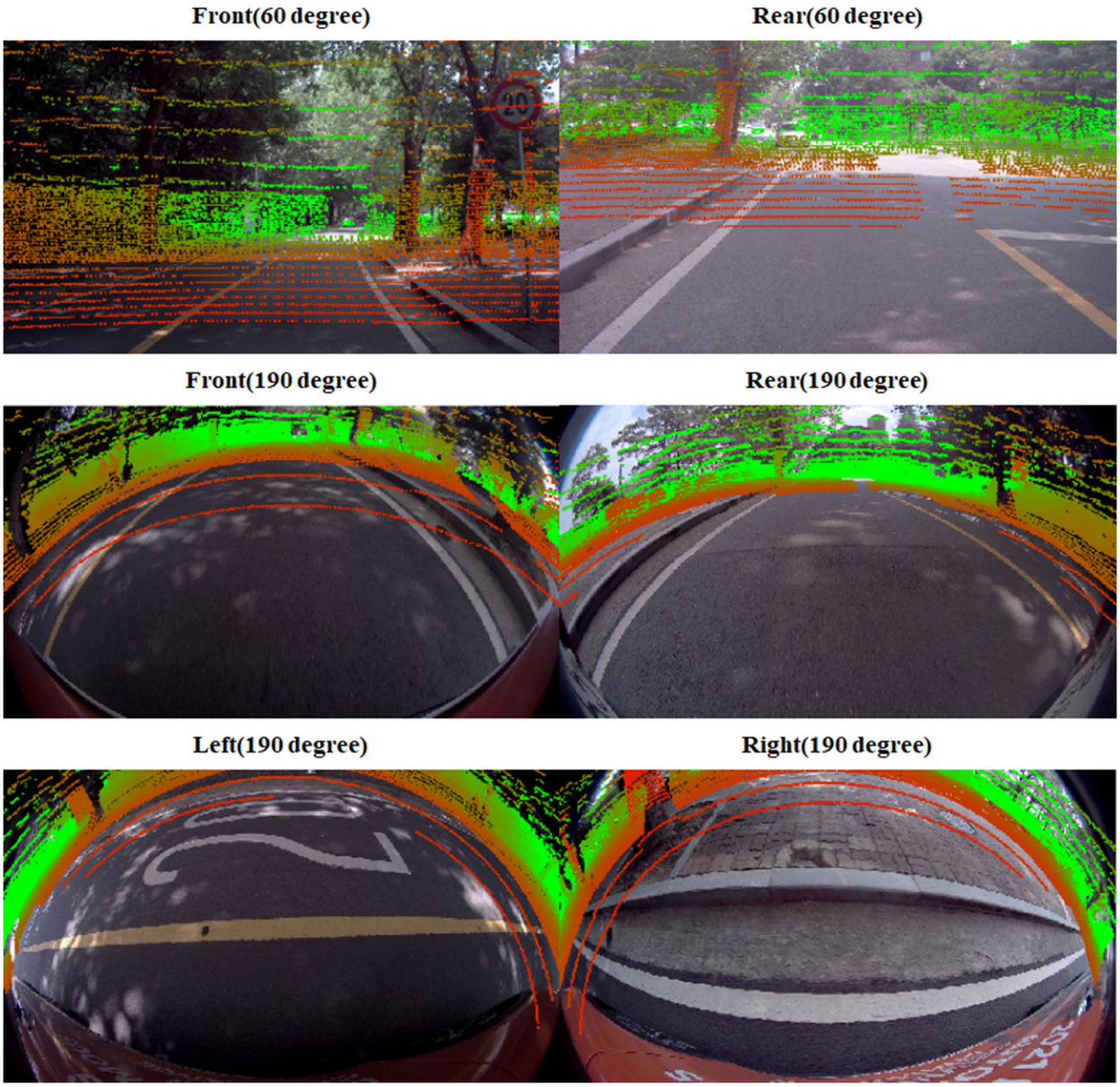

FIGURE 20. Six camera calibration result with pandar-40M(normal road scene).

TABLE 6. Comparative evaluation of non-overlapped sensors (non-overlapped lidar result/overlapped lidar result).

Error Type	Point Projection RMS	Quaternion Error	X Error	Y Error	Z Error
Value	0.4435	0.0001	0.1447	0.1885	0.2573

B. QUALITATIVE EVALUATION

In addition to the quantitative evaluation of extrinsic calibration, this paper used the qualitative evaluation method to analyze the results. We can see how exact the match was between the camera image and the overlayed LiDAR.

Fig. 18 shows the results of our system vs. Zhou et al. and Yuan et al. for the narrow front camera and Pandar-40m, and Fig. 19 shows the calibration results of our system vs. Zhou et al. and Yuan et al. for the narrow front camera and VLP-16.

Finally, Fig. 20 shows the calibration results of six cameras for road scenes using our solution. The scenes were collected at Chungbuk National University Gaesin Campus.

V. CONCLUSION

Numerous papers and software exist in the field of sensor calibration. When producing autonomous driving solutions, ADAS (Advanced Driving Assistance Systems), and generating Deep Learning GT (Ground Truth), sensor calibration

data are widely used, and are a requirement for many users. The types of calibration are diverse, as listed in Section 2 of our paper, and numerous studies and developers have improved their skills.

However, the widely used software are subject to many environmental constraints at calibration, such as using only a specific target in a specific environment. On the other hand, the proposed method has reusability, in that the user can select the target type and the method can be performed even when there is no intersection between the FoV of sensors performing the calibration, or when using low-cost equipment. It is a decisive advantage for sensor fusion and autonomous vehicles.

Additionally, in the case of general self-driving camera–LiDAR calibration, the vehicle is stopped, and a person moves the calibration target. Since our system is configured so that the vehicle or robot moves and calibrates, human error can be reduced in the research or development environment.

In this paper, features are extracted based on a commonly used planar-based checkerboard, but any object with a shape that can be mathematically modeled, such as a traffic cone, can be used as a target. If these characteristics are utilized, calibration can be performed via fusion with deep learning-based learners, such as 3D object detection and semantic segmentation, which are currently being actively studied, and the advantage of extracting features in a general road environment can be realized. It has the potential to advance research up to online calibration [54], which can be said to be an essential element in the future autonomous driving vision field.

In terms of performance, this paper has improved results compared to the method that is widely used in the existing calibration field. In addition, other algorithms and software do not consider low-cost LiDAR equipment or calibration between sensors that do not overlap FoV. Some algorithms apply the internal physical properties of a specific LiDAR to the detector. However, our system has independence with the LiDAR type and mount position, and so this system has a better flexibility for autonomous vehicle applications.

REFERENCES

[1] (2021). *HMG Autonomous Challenge*. Accessed: Aug. 8, 2022. [Online]. Available: https://young.hyundai.com/program/vehicle-tech-contest/info/detail.do?seq=7244

[2] (2022). *Autonomous Vehicles Competition for University Students (Ministry of Trade, Industry and Energy, MOTIE)*. Accessed: Aug. 8, 2022. [Online]. Available: http://autonomouscar.or.kr/sub101.html

[3] (2022). *International Student EV Car Competition*. Accessed: Aug. 8, 2022. [Online]. Available: http://carsa.kr/cev/01introduction01.php

[4] *Autonomous Vehicle Testbed & C-ITS*. Accessed: Aug. 8, 2022. [Online]. Available: https://topis.seoul.go.kr/openEngCits.do

[5] J. Domhof, J. F. P. Kooij, and D. M. Gavrila, "An extrinsic calibration tool for radar, camera and lidar," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, Montreal, QC, Canada, May 2019, pp. 8107–8113.

[6] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognit.*, vol. 47, pp. 2280–2292, Jun. 2014.

[7] E. Olson, "AprilTag: A robust and flexible visual fiducial system," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2011, pp. 3400–3407.

[8] J. Wang and E. Olson, "AprilTag 2: Efficient and robust fiducial detection," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Daejeon, South Korea, Oct. 2016, pp. 4193–4198.

[9] R. B. Rusu, "Semantic 3D object maps for everyday manipulation in human living environments," Ph.D. dissertation, Dept. Comput. Sci., Tech. Univ. Munich, Munich, Germany, 2009.

[10] G. Taubin, "Estimation of planar curves, surfaces, and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 11, pp. 1115–1138, Nov. 1991.

[11] R. Bergelt, O. Khan, and W. Hardt, "Improving the intrinsic calibration of a velodyne LiDAR sensor," in *Proc. IEEE SENSORS*, Glasgow, U.K., Oct. 2017, pp. 1–3.

[12] S. Ishida and J. E. Gayko, "Development, evaluation and introduction of a lane keeping assistance system," in *Proc. IEEE Intell. Vehicles Symp.*, Parma, Italy, Jun. 2004, pp. 643–944.

[13] K. Choi, H. Jung, and J. Suhr, "Automatic calibration of an around view monitor system exploiting lane markings," *Sensors*, vol. 18, no. 9, p. 2956, Sep. 2018.

[14] D. Lee, W. P. Tay, and S.-C. Kee, "Birds eye view look-up table estimation with semantic segmentation," *Appl. Sci.*, vol. 11, no. 17, p. 8047, Aug. 2021.

[15] J. K. Suhr and H. G. Jung, "Fully-automatic recognition of various parking slot markings in around view monitor (AVM) image sequences," in *Proc. 15th Int. IEEE Conf. Intell. Transp. Syst.*, Anchorage, AK, USA, Sep. 2012, pp. 1294–1299.

[16] D. Lee and S.-C. Kee, "Real-time implementation of the parking line departure warning system using partitioned vehicle region images," *Trans. Korean Soc. Automot. Engineers*, vol. 27, no. 7, pp. 553–560, Jul. 2019.

[17] S. Lee, D. Lee, and S.-C. Kee, "Deep-learning-based parking area and collision risk area detection using AVM in autonomous parking situation," *Sensors*, vol. 22, no. 5, p. 1986, Mar. 2022.

[18] M. Wilczkowiak, E. Boyer, and P. Sturm, "Camera calibration and 3D reconstruction from single images using parallelepipeds," in *Proc. 8th IEEE Int. Conf. Comput. Vis. (ICCV)*, Vancouver, BC, Canada, Jul. 2001, pp. 142–148.

[19] S. Izadi, A. Davison, A. Fitzgibbon, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, and D. Freeman, "KinectFusion: Real-time 3D reconstruction and interaction using a moving depth camera," in *Proc. 24th Annu. ACM Symp. User Interface Softw. Technol. (UIST)*, Barbara, CA, USA, 2011, pp. 559–568.

[20] A. E. Conrady, "Lens-systems, decentered," *Monthly Notices Roy. Astron. Soc.*, vol. 79, pp. 384–390, Mar. 1919.

[21] D. C. Brown, "Decentering distortion of lenses," *Photogramm. Eng. Remote Sens.*, vol. 32, no. 3, p. 444, 1966.

[22] J. Kannala and S. S. Brandt, "A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 8, pp. 1335–1340, Aug. 2006.

[23] S. K. Nayar, "Catadioptric omnidirectional camera," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Juan, PR, USA, Jun. 1997, pp. 482–488.

[24] C. Mei and P. Rives, "Single view point omnidirectional camera calibration from planar grids," in *Proc. IEEE Int. Conf. Robot. Autom.*, Rome, Italy, Apr. 2007, pp. 2945–3950.

[25] *Camera Calibration and 3D Reconstruction*. Accessed: Aug. 5, 2022. [Online]. Available: https://docs.opencv.org/3.4/d9/d0c/group__calib3d.html

[26] *Fisheye Camera Model*. Accessed: Aug. 5, 2022. [Online]. Available: https://docs.opencv.org/3.4/db/d58/group__calib3d__fisheye.html

[27] *Cv: Omnidir Namespace Reference*. Accessed: Aug. 5, 2022. [Online]. Available: https://docs.opencv.org/3.4/db/dd2/namespacecv_1_1omnidir.html

[28] Y. Wu and Z. Hu, "PnP problem revisited," *J. Math. Imag. Vis.*, vol. 24, no. 1, pp. 131–141, 2006.

[29] A. Ranganath, "The Levenberg–Marquardt algorithm," *Tutoral LM Algorithm*, vol. 11, no. 1, pp. 101–110, Jun. 2004.

[30] X.-S. Gao, X.-R. Hou, J. Tang, and H.-F. Cheng, "Complete solution classification for the perspective-three-point problem," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 8, pp. 930–943, Aug. 2003.

[31] F. Moreno-Noguer, V. Lepetit, and P. Fua, "Accurate non-iterative O(n) solution to the PnP problem," in *Proc. IEEE 11th Int. Conf. Comput. Vis.*, Rio de Janeiro, Brazil, Oct. 2007, pp. 1–8.

[32] V. Lepetit, F. Moreno-Noguer, and P. Fua, "EPnP: An accurate O(n) solution to the PnP problem," *Int. J. Comput. Vis.*, vol. 81, no. 2, pp. 155–166, 2009.
[33] J. A. Hesch and S. I. Roumeliotis, "A direct least-squares (DLS) method for PnP," in *Proc. Int. Conf. Comput. Vis.*, Barcelona, Spain, 2011, pp. 383–390.
[34] A. Penate-Sanchez, J. Andrade-Cetto, and F. Moreno-Noguer, "Exhaustive linearization for robust camera pose and focal length estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 10, pp. 2387–2400, Oct. 2013.
[35] T. Collins and A. Bartoli, "Infinitesimal plane-based pose estimation," *Int. J. Comput. Vis.*, vol. 109, no. 3, pp. 252–286, Sep. 2014.
[36] T. Ke and S. I. Roumeliotis, "An efficient algebraic solution to the perspective-three-point problem," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 7225–7233.
[37] G. Terzakis and M. Lourakis, "A consistently fast and globally optimal solution to the perspective-n-point problem," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 478–494.
[38] L. Zhou, Z. Li, and M. Kaess, "Automatic extrinsic calibration of a camera and a 3D LiDAR using line and plane correspondences," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Madrid, Spain, Oct. 2018, pp. 5539–5562.
[39] C. Yuan, X. Liu, X. Hong, and F. Zhang, "Pixel-level extrinsic self calibration of high resolution LiDAR and camera in targetless environments," *IEEE Robot. Autom. Lett.*, vol. 6, no. 4, pp. 7517–7524, Jul. 2021.
[40] A. Geiger, F. Moosmann, O. Car, and B. Schuster, "Automatic camera and range sensor calibration using a single shot," in *Proc. IEEE Int. Conf. Robot. Autom.*, Saint Paul, MN, USA, May 2012, pp. 3936–3943.
[41] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1231–1237, 2013.
[42] I. Cvisic, I. Markovic, and I. Petrovic, "Recalibrating the KITTI dataset camera setup for improved odometry accuracy," in *Proc. Eur. Conf. Mobile Robots (ECMR)*, Aug. 2021, pp. 1–6.
[43] S. C. Kee, "Research trend of Chungbuk national university smart car research center," in *Proc. SPG Conf. Korean Soc. Master Eng.*, Andong, GB, South Korea, 2022, p. 11.
[44] W. Lee, D. Lee, and S. C. Kee, "C-track HD map building on GNSS data for an autonomous driving simulator," in *Proc. 17th Int. Symp. Emerging Mater. Biomed. Eng. (ISET)*, Jeju, South Korea, 2022.
[45] P. Biber and W. Strasser, "The normal distributions transform: A new approach to laser scan matching," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Las Vegas, NV, USA, Oct. 2003, pp. 2743–2748.
[46] A. V. Segal, D. Haehnel, and S. Thrun, "Generalized-ICP," *Robot. Sci., Syst.*, vol. 2, no. 4, p. 435, Jun. 2009.
[47] *UTM—Universal Transverse Mercator*. Accessed: Aug. 8, 2022. [Online]. Available: http://geokov.com/education/utm.aspx
[48] Y. B. Jia, "Quaternions and rotations," *Com S*, vol. 477, no. 577, pp. 1–12, 2008.
[49] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with apphcatlons to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.
[50] S. Kollem and B. Panlal, "Enhancement of images using morphological transformations," *Int. J. Comput. Sci. Inf. Technol.*, vol. 4, no. 1, pp. 33–50, Mar. 2012.
[51] O. Patel, Y. P. S. Maravi, and S. Sharma, "A comparative study of histogram equalization based image enhancement techniques for brightness preservation and contrast enhancement," *Signal Image Process., Int. J.*, vol. 4, no. 5, pp. 11–25, 2013.
[52] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proc. 6th Int. Conf. Comput. Vis.*, Bombay, India, 1998, pp. 839–846.
[53] P. Gargallo, E. Prados, and P. Sturm, "Minimizing the reprojection error in surface reconstruction from images," in *Proc. IEEE 11th Int. Conf. Comput. Vis.*, Rio de Janeiro, Brazil, Oct. 2007, pp. 1–8.
[54] Y. Zhu, C. Li, and Y. Zhang, "Online camera-LiDAR calibration with sensor semantic information," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Paris, France, May 2020, pp. 4970–4976.

DONGKYU LEE was born in Seoul, South Korea, in 1993. He received the B.S. degree in electronic engineering and the M.S. degree in smartcar engineering from Chungbuk National University, South Korea, in 2017 and 2019, respectively, where he is currently pursuing the Ph.D. degree in smartcar engineering.

He was a Dispatch Researcher (Intern Student) at NTU Singapore for six months with the support of the Korean Government, in 2020. His research interests include object detection, lane detection, 3-D reconstruction for autonomous vehicles/advanced driving assistance systems (ADAS), and sensor calibration between each camera, each LiDAR, and camera-LiDAR.

SEOK-CHEOL KEE (Member, IEEE) received the B.S. and M.S. degrees in control and instrumentation engineering and the Ph.D. degree in electrical engineering from Seoul National University, Seoul, South Korea, in 1987, 1989, and 2001, respectively.

From 1989 to 2007, he was with the Samsung Advanced Institute of Technology, Suwon, South Korea, where he was the Group Leader responsible for image processing research. From 2010 to 2015, he was with Mando, Pangyo, South Korea, where he was the Research Director responsible for the Electronics Research and Development Laboratory. In 2015, he joined as a Faculty Member with Chungbuk National University, Cheongju, South Korea. His research interests include automotive computer vision, ADAS sensor systems, autonomous vehicle systems, and power electronic systems.