https://doi.org/10.7840/kics.2025.50.5.703

Deep Learning-Based Lightweight LiDAR and Fisheye Camera Online Extrinsic Calibration

Sang-Chul Kim◆°, Yeong-min Jang*

ABSTRACT

Autonomous driving has been extensively researched in recent years. To improve the mobility and decision-making capabilities of autonomous vehicles, multiple sensors have been integrated to complement the limitations of individual sensors. For example, Light Detection and Ranging (LiDAR) is frequently combined with camera data to overcome the narrow field-of-view (FoV) of traditional pinhole cameras. The fisheye cameras of LiDAR expand the FoV from up to 80° in traditional cameras to 180°, which is advantageous for autonomous driving applications. This study introduces a lightweight, deep learning-based LiDAR-fisheye camera fusion model for real-world environments. The mean translation errors are 1.375, 0.753, and 1.208 cm along the X, Y, and Z axes, respectively, and the mean rotation errors are 0.171°, 0.150°, and 0.089° in the roll, pitch, and yaw directions, respectively. These results demonstrate the efficiency and proficiency of our sensor-fusion approach for autonomous driving.

Key Words : Online extrinsic calibration, LiDAR, Fisheye camera, Sensor fusion

Ⅰ. Introduction

Autonomous driving technology, an emerging field that enables vehicles or robots to navigate and operate without human intervention, can potentially revolutionize transportation and has therefore attracted much attention from both academicians and industrial researchers. Many state-of-the-art methods achieve autonomous functionality through advanced technologies, seeking more localized and efficient sensor solutions than satellite-based systems such as Global Positioning Systems and Global Navigation Satellite Systems, which are primarily limited by high latency. The combination of camera and Light Detection and Ranging (LiDAR) data is considered as a standout approach for autonomous driving systems[1].

Cameras capture high-resolution, color-rich imagery of the surrounding environment, enabling the detection and recognition of objects, textures, and other visual details with remarkable clarity. The standard pinhole camera is widely used because it is both simple and effective, but its field-of-view (FoV) is limited to 80°. An ultrawide FoV (180°) is provided by fisheye cameras, which can capture substantially more data and features in a single image than pinhole cameras[2].

However, as camera sensors cannot measure distances directly, they cannot adequately capture the depth information. The limited depth perception challenges the interpretation of spatial relationships between the vehicle and its surroundings objects.

Unlike traditional cameras, LiDAR sensors provide precise distance measurements. The collected data are represented as a three-dimensional array of points

※ This work was supported by Korea Research Institute for defense Technology planning and advancement(KRIT) -Grant funded by Defense Acquisition Program Administration(DAPA) (KRIT-CT-23-041)
◆° First and Corresponding Author : Kookmin University School of Computer Science, sckim7@kookmin.ac.kr, 종신회원
* Kookmin University School of Electronics Engineering, yjang@kookmin.ac.kr, 종신회원
논문번호 : 202502-031-A-RU, Received February 12, 2025; Revised March 11, 2025; Accepted April 3, 2025

with X, Y, and Z coordinates in a Cartesian system. These arrays, called point clouds, provide a detailed spatial representation of the environment. Accordingly, LiDAR is a critical component in autonomous driving systems.

Despite these advantages, the high cost and computational intensity of LiDAR devices greatly raises the cost of large-scale deployment. In addition, LiDAR data are inherently sparse and lack red - green - blue (RGB) color information, complicating their interpretation and processing[3].

To avoid these limitations, LiDAR and camera sensors data are commonly combined in recent autonomous driving systems. The integration of cameras and LiDAR can notably enhance the capabilities of both autonomous driving and robotics systems. However, the simultaneous deployment of multiple sensors increases the computational complexity of the system. The data generated by these sensors can be highly diverse and complex, requiring sophisticated processing to extract meaningful insights.

Traditionally, feature extraction and decision-making tasks based on fusion sensor data have been performed by algorithms such as Iterative Closest Point[4], RANSAC[5], and K-Means[6]. Although these methods perform effectively in structured or static environments, they are often less successful under dynamic driving conditions.

In recent years, deep learning has become a focal point of both research and practical applications. Researchers in the autonomous driving and robotics fields have explored ways of integrating deep learning techniques into their workflows. Given high-quality and normally distributed data, a well-designed deep learning methods leverage neural networks to adapt to variations in sensor viewpoints, lighting conditions, and environmental dynamics, making them effective for extrinsic LiDAR-camera calibration in complex and unstructured environments.

The present study proposes a deep learning-based online extrinsic calibration method that fuses the data of LiDAR and fisheye camera sensors. Leveraging the complementary strengths of the two sensors, the method aims to enhance the capabilities of autonomous driving and robotic systems. Whereas LiDAR provides precise distance measurements between the sensor (or vehicle) and surrounding objects, the fisheye camera captures a comprehensive 180° RGB representation of the environment.

Ⅱ. Related Research

Both the intrinsic and extrinsic parameters of each sensor in the data-fusion system must be calibrated to ensure accurate data integration. In this work, we assume that the intrinsic parameters of the fisheye camera and LiDAR provided in the dataset have been calibrated by the manufacturer, as we follow similar studies[7,8]. Therefore, only extrinsic calibration is considered here. However, for real-environment application, we acknowledge that even minor residual errors in intrinsic calibration can affect the extrinsic calibration accuracy.

The existing LiDAR-Camera calibration methods can be broadly categorized into target-based, targetless, and learning-based methods. These three categories are summarized below.

2.1 Target-based Methods

For accurate calibration, target-based methods frequently establish the correspondence between two-dimensional (2D) camera images and three-dimensional (3D) LiDAR points of specially designed objects.

Checkerboards have been extensively utilized in target-based extrinsic calibration studies. For instance, Geiger et al.[9] calibrated LiDAR and cameras by detecting the corners of checkerboard patterns and estimating the transformation matrix between the two sensors. Similarly, Wang et al.[10] proposed an online extrinsic calibration method employing a checkerboard for 3D LiDAR and panoramic cameras.

2.2 Targetless methods

In real-world autonomous driving and robotics scenarios, where ideal calibration targets such as checkerboards are often unavailable, researchers have developed various targetless methods.

For instance, Song et al.[11] proposed the Galibr method, which performs extrinsic LiDAR and camera calibration through ground plane features. Borer et

al.[12] calibrated LiDAR and its fisheye camera using a continuous online extrinsic calibration approach based on a mutual-information matching method.

2.3 Learning-based Methods

Deep learning has been widely adopted in extrinsic calibration scenarios. Lv et al.[7] introduced LCCNet, a neural network model for LiDAR and camera self-calibration, which leverages a cost volume layer to facilitate feature matching between RGB image features and depth map features. Wang et al.[8] proposed Multiresolution LiDAR-Camera Calibration Network (MRCNet), an end-to-end neural network that inputs the RGB image and the depth image generated from the LiDAR point cloud and predicts the transformation matrix for extrinsic calibration. Their network adopts a geometric feature-constraint module and performs multiresolution feature extraction and feature matching. Zhu et al.[13] proposed CalibDepth, which also relies on RGB and depth images. The depth image in CalibDepth provides a unified representation, bridging the gap between the RGB image data and the point cloud inputs. CalibDepth also incorporates long short-term memory for autoregressive generation of the calibration actions at each step.

The abovementioned studies highlight a notable lack of research on extrinsic calibration between LiDAR and fisheye cameras, particularly in learning-based approaches. In this study, we design an end-to-end, deep learning-based online extrinsic calibration method for LiDAR and fisheye cameras.

Ⅲ. Proposed Methods

The proposed model integrates a residual spherical convolution layer for feature extraction from fisheye images with an optical flow based on the depth map.

3.1 Problem Formulation

Given a multimodal input composed of RGB images and a LiDAR point cloud, online extrinsic calibration obtains a transformation matrix T consisting of a 3×3 rotation matrix and 1×3 translation vector. During the learning process, a random transformation T_{rand} is applied to matrix T, obtaining an initial matrix

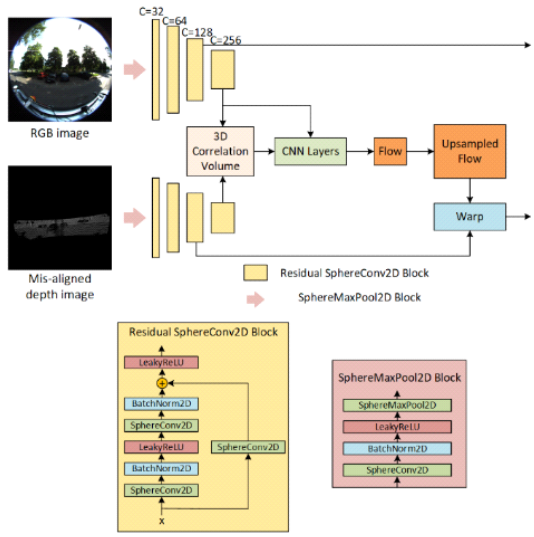

Fig. 1. Architecture of the proposed model.

T_{init}. T_{init} is then applied to the point cloud to obtain a mis-calibrated depth image. Projection matrix fisheye-depth images were generated as described in [14]. The objective of the proposed model is to estimate the extrinsic calibration matrix T_{rand}, denoted as T_{pred}, based on the RGB image and the point cloud using the mis-calibrated depth image.

3.2 Network Architecture

Our proposed model is divided into three main components: feature extraction, feature matching, and a regression layer. This subsection describes each component in detail.

3.2.1 Feature Extraction

Feature extraction by the proposed model occurs through two parallel branches, one for the fisheye RGB image, the other for the mis-calibrated depth map. To effectively handle the spherical images generated by the fisheye camera, we introduce a residual spherical feature-extraction technique to both branches. This approach builds upon a modified spherical convolution of SphereNet[15], adapting it to a residual framework to capture the semantic features in both the original and refined inputs, ultimately enhancing the quality of the extracted features.

The spherical convolution of SphereNet lifts the kernel of a 3×3 convolution layer into spherical

space. The kernel shape is defined in spherical space with $j,k \in \{-1, 0, 1\}$ and step sizes of Δ_θ and Δ_ϕ.

The kernel sampling s is defined as

$$s_{(0,0)} = (0,0) \tag{1}$$

$$s_{(\pm1,0)} = (\pm\Delta_\phi, 0) \tag{2}$$

$$s_{(0,\pm1)} = (0, \pm\Delta_\theta) \tag{3}$$

$$s_{(\pm1,\pm1)} = (\pm\Delta_\phi, \pm\Delta_\theta) \tag{4}$$

where θ measures the inclination from the positive z-axis downward (ranging from 0° at the north pole to 180° at the south pole), while ϕ measures the rotation around the z-axis in the xy-plane, increasing counterclockwise from a reference direction (typically the positive x-axis).

3.2.2 Feature Matching

The feature matching layer is inspired by MRCNet[8], which creates a 3D correlation volume representing the spatial correlation between the features extracted from RGB image and LiDAR depth map. The 3D correlation volume is defined as

$$C\left(x_{rgb}(p_i), x_{depth}(p_j)\right) = \bigcup_{i,j \in D}\left(\left(x_{rgb}(p_i)\right)^T, x_{depth}(p_j)\right), \tag{5}$$

where x_{rgb} and x_{lidar} denote a feature in the RGB image and depth map respectively, D represents the search region where feature correspondences are computed, while p_i and p_j denote pixel locations in the RGB image and the depth map. The optical flow between the two feature sets is then estimated from the 3D correlation volume within a learning-based layer, comprising a regular convolutional layer that outputs a channel of size two representing the horizontal and vertical flow vectors. In this study, the depth map was taken as the "target" of the optical flow.

3.2.3 Regression Layer

The regression layer inputs the features x with the finest resolution and processes them through multiple convolutional layers, followed by an average pooling layer that condenses the features. The pooled features are flattened and then passed into a fully connected layer along with an activation function LeakyReLU

$$LeakyReLU(x_i) = \begin{cases} x_i & if\ x_i \geq 0 \\ \alpha x_i & if\ x_i < 0 \end{cases} \tag{6}$$

where α is a small constant. Afterwards, the feature x is passed into a dropout layer. The process can be represented mathematically as

$$x \leftarrow avgpool(x) \tag{7}$$

$$x \leftarrow view(x, B, -1) \tag{8}$$

$$x \leftarrow FCN(x) \tag{9}$$

$$x \leftarrow LeakyReLU(x, \alpha) \tag{10}$$

$$x \leftarrow Dropout(x) \tag{11}$$

where B is batch size. This operation reshapes pooled feature x from B × C × 1 × 1 to a vector of size B × C.

The reshaped feature is passed into two prediction branches: one for rotation in quaternion q format and the other for predicting the X, Y, and Z components of the translation vector t. Subsequently, the predicted quaternion is normalized as below

$$q_{norm} = \frac{q}{\sqrt{\sum_{i=1}^{4} q_i^2 + \epsilon}} \tag{12}$$

where ϵ ($\epsilon = 1e-10$) is a small constant added for numerical stability. The output T_{pred} is a normalized quaternion vector and a translation vector, which can be represented as

$$(q_{norm}, t) \tag{13}$$

where $q_{norm} \in \mathbb{R}^4$ is the normalized quaternion vector and $t \in \mathbb{R}^3$ is the translation vector.

3.3 Loss Functions

In this study, the loss functions of both quaternion and translation predictions in T_{pred} are expressed in

terms of the Euclidean distance. The Euclidean-distance loss functions of translation and rotation are respectively defined as

$$\mathcal{L}_t^l = \| t_{gt}^l - t^l \|_2, \tag{14}$$

$$\mathcal{L}_r^l = \| q_{gt}^l - \frac{q^l}{\| q^l \|} \|_2. \tag{15}$$

where $\mathcal{L}_t^l$ is translation loss, t^l is predicted translation and t_{gt}^l is ground truth translation. Similarly, $\mathcal{L}_r^l$ is rotational loss, q^l is predicted quaternion and q_{gt}^l is ground truth quaternion.

3.4 bration Inference

Similarly to [7] and [8], the designed network does not directly predict the extrinsic calibration matrix. As the generated depth image is based on the mis-calibration matrix T_{rand}, the predicted matrix will express the deviations from the initial parameters. To obtain the extrinsic calibration matrix, the quaternion q in T_{pred} is first converted into a standard 3 × 3 rotation matrix, which can be represented as

$$R = \begin{bmatrix} 1-2(y^2+z^2) & 2(xy-wz) & 2(xz+wy) \\ 2(xy+wz) & 1-2(x^2+z^2) & 2(yz-wx) \\ 2(xz-wy) & 2(yz+wx) & 1-2(x^2+y^2) \end{bmatrix} \tag{16}$$

where R is 3 × 3 rotation matrix and w, x, y, z is the element in quaternion vector q = (w, x, y, z). Finally, the T matrix is computed as follows

$$T = T_{pred}^{-1} \cdot T_{init}. \tag{17}$$

Fig. 2. Examples of fisheye images from the KITTI-360 dataset

where T_{pred} is predicted transformation matrix misalignment and T_{init} is the matrix multiplication of transformation matrix misalignment T_{rand} and ground truth extrinsic parameters T.

Several learning-based methods for extrinsic calibration [7] and [8] employ an iterative inference approach, which progressively refines the calibration through multiple networks across different ranges. In contrast, this study only employs single model approach similar to that in [16].

Ⅳ. Experiment

4.1 Dataset

The experiment was conducted on the Karlsruhe Institute of Technology and Toyota Technological Institute (KITTI)-360 dataset, which comprises eight sequences taken by four cameras, two utilizing fisheye lenses. The fisheye cameras of the KITTI-360 dataset use the model of Mei and Rives[15] as the projection model.

The present experiment was conducted on fisheye images from camera 2 (76,251 images in total). We also used the Velodyne HDL-64E LiDAR point cloud

Table. 1. Model performance on two different initial misalignments.

Initial misalignment (tr/rot)	Metrics	Translation (cm)				Rotation (°)			
		E_t	X	Y	Z	E_r	*Roll*	*Pitch*	*Yaw*
0.5m/5.0°	Mean	2.264	1.375	0.753	1.208	0.282	0.171	0.150	0.089
	Median	1.991	1.085	0.579	0.941	0.241	0.126	0.119	0.070
	Std	1.095	1.228	0.716	1.095	0.197	0.181	0.137	0.081
0.2m/2.0°	Mean	1.608	0.936	0.486	0.929	0.104	0.130	0.108	0.073
	Median	0.754	0.756	0.394	0.754	0.094	0.104	0.088	0.060
	Std	0.927	0.814	0.436	0.784	0.057	0.115	0.091	0.059

data provided in the KITTI-360 dataset for depth map generation. The fisheye images and LiDAR have been already synchronized in the dataset. All sequences except sequence 00 were used for training; the remaining was reserved for testing and validation.

As mentioned in Section III, the point cloud data were mis-calibrated using T_{rand} and projected onto the 2D fisheye image. The initial translation and rotation misalignments were 0.5 meter and 5.0°, respectively. Following[7], the RGB images were augmented with color jittering alone.

4.2 Training Details

The proposed deep learning model was implemented using the PyTorch library and trained on an NVIDIA RTX 3090Ti GPU (24GB), paired with an Intel Xeon 4215R CPU and 32GB of random-access memory (RAM). The model includes approximately 8M parameters. Training was performed with a batch size of 16 over 75 epochs. The initial learning rate was set to 1e−4 and was reduced by a factor of 0.3 whenever the validation performance stagnated for 10 consecutive evaluations.

4.3 Results and Discussion

In this experiment, our approach operates within a maximum initial misalignment of 5.0° in rotation and 0.5 meter in translation. The translation error E_t, defining the Euclidean distance between the predicted translation t_{pred} and ground truth translation t_{gt}, is mathematically expressed as

$$E_t = \| t_{gt} - t_{pred} \|_2. \quad (19)$$

where t_{gt} and t_{pred} are ground truth and predicted translation respectively. Meanwhile, E_r is calculated using the quaternion distance formula. The quaternion distance is computed as

$$E_r(q_{gt}, q_{pred}) = 2\tan^{-1}\left(\frac{\sqrt{t_x^2 + t_y^2 + t_z^2}}{|t_w|}\right) \quad (19)$$

where q_{gt} is the ground truth quaternion, q_{pred} is predicted quaternion and t_w, t_x, t_y, t_z are the Hamilton products of q_{gt} and q_{pred}. In addition, we provide the absolute error on each translation axis (*X*, *Y*, *Z*) and in each rotational angle (*roll, pitch, yaw*). The roll, pitch, yaw are defined as

$$yaw = atan2\left(R_{pred_{21}}, R_{pred_{11}}\right) \quad (20)$$

$$pitch = atan2\left(-R_{pred_{31}}, \sqrt{R_{pred_{31}}^2 + R_{pred_{33}}^2}\right) \quad (21)$$

$$roll = atan2\left(R_{pred_{32}}, R_{pred_{33}}\right) \quad (22)$$

where R_{pred} is the 3 × 3 predicted rotation matrix converted from predicted quaternion q_{pred}.

Given the limited references of learning-based LiDAR and fisheye camera calibration, particularly on the KITTI-360 dataset, we also benchmarked our model against other models trained on regular pinhole camera images from the KITTI Odometry dataset using similar training methods. The KITTI Odometry consists of 22 sequences with 43,552 frames in total. All comparisons were performed at initial translation

Table. 2. Comparison of other methods with an initial misalignment of 0.5m/5.0° (translation/rotation).

Method (KITTI Odometry)	Translation (cm)				Rotation (°)			
	Mean	X	Y	Z	Mean	Roll	Pitch	Yaw
LCCNet [7]	1.673	1.669	1.319	2.032	0.157	0.055	0.308	0.109
MRCNet [8]	0.895	0.814	0.614	1.257	0.111	0.037	0.229	0.068
CalibDepth [13]	0.918	0.997	0.565	1.193	0.163	0.041	0.060	0.388
Method (KITTI-360)	**Translation (cm)**				**Rotation (°)**			
	Mean	X	Y	Z	Mean	Roll	Pitch	Yaw
Borer et al. [12]	-	-	-	-	0.147	0.175	**0.117**	0.151
Ours	**1.106**	**1.375**	**0.753**	**1.208**	**0.136**	**0.171**	0.150	**0.089**

and rotation misalignments of 0.5 meter and 5.0°, respectively, and all reported metrics are the means of all sequences. The comparison results are presented in Table 2.

The proposed method outperforms the approach of Borer et al.[12] in terms of rotation accuracy, but as Borer et al. did not report the translation performance, the translation misalignment in their method is not directly comparable with ours. Furthermore, Borer et al.'s method is not a deep learning-based method. Compared to KITTI Odometry dataset methods, our model achieves misalignment prediction performance comparable to those methods within a similar maximum misalignment range.

We also evaluate our method alongside other deep learning-based approaches by comparing model resource utilization during inference such as parameter count, inference time, GPU memory usage, and storage capacity. For iterative methods, since they have multiple trained models, we selected a single model with an initial misalignment similar to our approach. The results were replicated in a resource-constrained setup featuring an NVIDIA RTX 3060 (8GB), an Intel i7-12700 (12th Gen) processor, and 16GB of RAM. The operating system is Ubuntu 20.04. The number of workers used for the GPU is 0 and the batch size is set to 1. The findings are summarized in Table 3. This evaluation highlights the model's efficiency under limited computational resources, reinforcing its practicality for real-world deployment.

As seen in Table 3, our proposed model has a slightly slower inference time compared to MRCNet[8]. However, in other aspects, such as parameter count, GPU memory usage, and storage requirements, our model is significantly more efficient. This highlights the lightweight property of our model and its suitability for real-time inference.

Fig. 3. Estimated extrinsic calibration using the proposed deep learning model. From left to right, each row shows the initial misalignment, ground truth, and calibrated image results.

The extrinsic calibration results are visualized in Fig. 3. As shown in the image, the LiDAR point cloud projection closely resembles the ground truth, indicating that both the LiDAR and fisheye camera are calibrated with minimal misalignment.

V. Conclusion

We proposed a deep learning-based method for extrinsic calibration between LiDAR and fisheye cameras. The model introduces a feature-extraction approach using residual SphereConv2D, an enhancement of SphereConv2D from SphereNet. In addition, it incorporates optical flow with a depth image as the flow reference. This lightweight model with only 8M parameters is designed for real-time applications.

The proposed method calibrates the extrinsic pa-

Table. 3. Comparison of resource utilization across deep learning methods during inference

Method (KITTI Odometry)	Parameter count (params)	Average inference time (second)	Average GPU memory usage (MB)	Storage (MB)
LCCNet [7]	69,987,887	0.010	2,030	840.1
MRCNet [8]	76,750,343	0.020	2,496	827.5
Method (KITTI-360)	**Parameter count (params)**	**Average inference time (second)**	**GPU Memory usage (MB)**	**Storage (MB)**
Ours	**7,925,365**	0.021	**320**	**90.98**

rameters under initial misalignments of up to 0.5 meter in translation and 5.0° in rotation. It achieves mean translation errors of 1.375, 0.753, and 1.208 cm along the X, Y, and Z axes, respectively, and mean rotation errors of 0.171°, 0.150°, and 0.089° in the roll, pitch, and yaw directions, respectively.

References

[1] S. Atakishiyev, M. Salameh, H. Yao, and R. Goebel, "Explainable artificial intelligence for autonomous driving: A comprehensive overview and field guide for future research directions," *IEEE Access*, vol. 12, pp. 101603-101625, 2024. (https://doi.org/10.1109/ACCESS.2024.3431437)

[2] V. R. Kumar, C. Eising, C. Witt, and S. K. Yogamani, "Surround-view fisheye camera perception for automated driving: Overview, survey & challenges," *IEEE Trans. Intell. Transport. Syst.*, vol. 24, no. 4, pp. 3638-3659, Apr. 2023. (https://doi.org/10.1109/TITS.2023.3235057)

[3] S. Y. Alaba and J. E. Ball, "A survey on deep-learning-based LiDAR 3D object detection for autonomous driving," *Sensors*, vol. 22, no. 24, p. 9577, Jan. 2022. (https://doi.org/10.3390/s22249577)

[4] J. He, J. Fang, S. Xu, and D. Yang, "Indoor robot SLAM with multi-sensor fusion," *Int. J. Advanced Netw., Monitoring and Controls*, vol. 9, no. 1, pp. 10-21, Mar. 2024. (https://doi.org/10.2478/ijanmc-2024-0002)

[5] M. A. Rasyidy, Y. Y. Nazaruddin, and A. Widyotriatmo, "Regression with ensemble of RANSAC in camera-LiDAR fusion for road boundary detection and modeling," in *2022 IEEE Int. Conf. Multisensor Fusion and Integration for Intell. Syst. (MFI)*, pp. 1-6, Sep. 2022. (https://doi.org/10.1109/MFI55806.2022.9913856)

[6] M. A. V. Saucedo, et al., "Event camera and LiDAR based human tracking for adverse lighting conditions in Subterranean Environments," *IFAC-Papers OnLine*, vol. 56, no. 2, pp. 9257-9262, Apr. 2023. (https://doi.org/10.1016/j.ifacol.2023.10.008)

[7] X. Lv, B. Wang, D. Ye, and S. Wang, "LCCNet: LiDAR and camera self-calibration using cost volume network," *arXiv preprint arXiv:2012.13901*, Apr. 2021. (https://doi.org/10.48550/arXiv.2012.13901)

[8] H. Wang, Z. Wang, G. Yu, S. Yang, and Y. Yang, "MRCNet: Multiresolution LiDAR-camera calibration using optical center distance loss network," *IEEE Sensors J.*, vol. 24, no. 12, pp. 19661-19672, Jun. 2024. (https://doi.org/10.1109/JSEN.2023.3328267)

[9] A. Geiger, F. Moosmann, Ö. Car, and B. Schuster, "Automatic camera and range sensor calibration using a single Shot," in *2012 IEEE ICRA*, pp. 3936-3943, May 2012. (https://doi.org/10.1109/ICRA.2012.6224570)

[10] W. Wang, K. Sakurada, and N. Kawaguchi, "Reflectance intensity assisted automatic and accurate extrinsic calibration of 3D LiDAR and panoramic camera using a printed chessboard," *Remote Sensing*, vol. 9, no. 8, p. 851, Aug. 2017. (https://doi.org/10.3390/rs9080851)

[11] W. Song, M. Oh, J. Lee, and H. Myung, "Galibr: Targetless LiDAR-camera extrinsic calibration method via ground plane initialization," *arXiv preprint arXiv:2406.11599*, Jun. 2024. (https://doi.org/10.48550/arXiv.2406.11599)

[12] J. Borer, J. Tschirner, F. Ölsner, and S. Milz, "Continuous online extrinsic calibration of fisheye camera and LiDAR," *arXiv preprint arXiv:2306.13240*, Jun. 2023. (https://doi.org/10.48550/arXiv.2306.13240)

[13] J. Zhu, J. Xue, and P. Zhang, "CalibDepth: unifying depth map representation for iterative LiDAR-camera online calibration," in *2023 IEEE ICRA*, pp. 726-733, May 2023. (https://doi.org/10.1109/ICRA48891.2023.10161575)

[14] C. Mei and P. Rives, "Single view point

omnidirectional camera calibration from planar grids," in *Proc. 2007 IEEE ICRA*, pp. 3945-3950, Apr. 2007. (https://doi.org/10.1109/ROBOT.2007.364084)

[15] B. Coors, A. P. Condurache, and A. Geiger, "SphereNet: Learning spherical representations for detection and classification in omnidirectional images," in *Computer Vision – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds., Cham: Springer International Publishing, pp. 525-541, 2018. (https://doi.org/10.1007/978-3-030-01240-3_32)

[16] G. Zhao, J. Hu, S. You, and C.-C. J. Kuo, "CalibDNN: Multimodal sensor calibration for perception using deep neural networks," *arXiv preprint arXiv:2103.14793*, Mar. 2021. (https://doi.org/10.48550/arXiv.2103.14793)

Sang-Chul Kim

1994 : B.S. degree, Kyungpook National University

2005 : Ph.D. degree (MS-Ph.D Integrated), Oklahoma State University

2006~Present : Professor, School of Computer Science, Kookmin University

<Research Interests> Real-time operating systems, wireless communication, artificial intelligence

[ORCID:0000-0003-2622-0426]

Yeong-min Jang

1985 : B.S. degree, Kyungpook National University

1987 : M.S. degree, Kyungpook National University

1999 : Ph.D. degree, University of Massachusetts

2002~Present : Professor, School of Electrical Engineering, Kookmin University

<Research Interests> Artificial intelligence, optical wireless communication, visible light communication, internet of things

[ORCID:0000-0002-9963-303X]