https://doi.org/10.7840/kics.2025.50.5.693

Lightweight LiDAR-Camera Online Extrinsic Calibration with Multi-Dilation Encoder Blocks

Sang-Chul Kim◆°, Yeong-min Jang*

ABSTRACT

As the integration of multi-sensor systems, such as cameras and LiDAR, becomes increasingly common in various fields, the development of efficient and accurate extrinsic calibration techniques is emerging as a critical task. This article presents a novel lightweight deep-learning network for LiDAR—Camera targetless extrinsic calibration, which consists of only 4 million parameters. The proposed method utilized CNN-based multi-dilation encoder blocks which can extract multi-scale features, especially for sparse LiDAR depth image. The proposed block allows the network to be lightweight and excel in calibration performance. The proposed method achieved translation errors of 1.08 cm, 0.18 cm, and 0.56 cm along the X, Y, and Z axes, respectively. Additionally, it achieved rotation errors of 0.182°, 0.139°, and 0.141° for roll, pitch, and yaw, respectively. The proposed method also performs calibration in a one-shot approach, which is suitable for real-time applications. These results highlight the capabilities of the proposed method in enabling reliable fusion of LiDAR and camera data, enhancing the perception capabilities of autonomous vehicles.

Key Words : Camera, LiDAR, Extrinsic Calibration, Lightweight Deep Learning, KITTI odometry dataset, sensor fusion.

Ⅰ. Introduction

Research into drone and autonomous vehicle technology has advanced rapidly in the past few years. The environment of drone or autonomous driving is very complex and dynamic. It is shown that a single sensor cannot guarantee reliable recognition in all scenarios.

LiDAR—camera systems are widely used in robotic vision applications such as 3D object detection and navigation task solving. In these applications, the LiDAR sensor can generate a sparse point cloud of the surroundings based on the distance measured by the light. The main advantage of LiDAR is its active illumination, which can operate independently of the ambient light. However, the disadvantages of LiDAR are its high cost, limited resolution, and low refresh rate, such as the Velodyne-64 LiDAR, which only measures 64 channels. In addition, LiDAR sensors cannot measure RGB information. RGB cameras are relatively inexpensive and can produce high-resolution color images at high frame rates. However, they cannot directly measure the depth information. In other words, LiDAR sensors generate sparse 3D information, while cameras capture 2D dense information. Fortunately, by fusion of LiDAR and camera measurements, most of the shortcomings of LiDAR sensors can be compensated for by RGB cameras, and vice versa. After that, LiDAR-camera systems can detect and analyze target objects with more advanced and intelligent vision. However, sensor fu-

※ 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 : 202502-039-A-RU, Received February 21, 2025; Revised April 1, 2025; Accepted April 7, 2025

sion requires extrinsic parameters of LiDAR and camera. Therefore, it is essential to calibrate the external parameters of these sensors in [1].

Extrinsic calibration is necessary to align the relative spatial position between LiDAR and the camera. Since LiDAR and cameras collect data in different ways, it is essential to determine their relative relationship accurately. Extrinsic calibration determines extrinsic parameters which are represented as a transformation matrix. This transformation matrix calculates the relative position and rotation matrix between LiDAR and the camera to convert the data into a common coordinate system.

One approach of extrinsic calibration is to use a specified target, such as a chessboard pattern or other straight-line features to extract 2D-3D correspondences between the sensors and estimate the extrinsic parameters of the LiDAR and cameras. These methods are also called target-based calibration methods. In addition, it involves comparing feature points from LiDAR point clouds and camera images to estimate the relative transformation between the two sensors. Most target-based correction algorithms are time-consuming, tedious, and offline, making them unsuitable for drone or autonomous driving. During drone operation, the positions of sensors will vary slightly depending on the flight time. After a period of operation, the sensor must be recalibrated to remove accumulated errors due to drift. This necessitates the use of online and targetless calibration methods.

Several targetless methods have been developed over the years, which use local feature extraction, such as edges and planes. However, these methods still lack the capability in recognizing diverse environments. Deep learning is introduced to enhance the targetless scheme. However, deep learning methods may incur high computational costs. One of the parameters to measure computational cost is the number of trainable parameters used in the network, which determines the size of the deep learning model. Many current methods use architecture comprising of more than 10 million parameters, such as PSNet[2] and CalibDepth[3]. These methods typically use well-known architectures such as ResNet18, which consists of approximately 11 million parameters. Reducing model size allows for less memory requirements, enabling its application in onboard computers in autonomous vehicles.

Therefore, in this study, we propose a new online camera-LiDAR extrinsic calibration, which can independently calibrate the extrinsic parameters between a camera and a LiDAR sensor without using any specific pattern or calibration object. We focus on developing a lightweight model with a reduced number of parameters. More specifically, our contribution can be summarized as follows:

1) We proposed a one-shot extrinsic calibration method comprising multi-dilation feature extraction blocks capable of extracting sparse features and fine-grained features of both RGB camera image and LiDAR depth image.
2) The proposed method only comprises ~4 million trainable parameters, making it lightweight for autonomous vehicle applications. The proposed architecture consists of two branches of a feature extraction network, each comprising of only 330,000 parameters.
3) By implementing one-shot calibration and lightweight model architecture, exquisite real-time performance can be achieved. This method was trained and tested on the KITTI odometry dataset, which consists of diverse environments.

Ⅱ. Related Research

Mainly, research works in extrinsic calibration have primarily concentrated on three categories: target-based methods, target-less methods, and learning-based methods, each with its own unique approach and advantages.

The first category, target-based methods, relies on objects with specified structures or patterns as multi-sensor co-shooting targets to obtain the extrinsic parameters between point clouds and RGB images[4]. Giacomini et al.[5] develop a simple yet robust method for LiDAR and RGB camera extrinsic calibration that leverages small markers and requires minimal human intervention, widely available calibration targets such as A3/A4-sized checkerboard patterns to reduce dependency on complex or custom calibration targets, which often require specialized equipment and

manufacturing. Jeong et al.[6] proposed a method which uses perspective projection instead of orthogonal projection onto a checkerboard target to address noisy LiDAR points problems while obtaining extrinsic parameters.

The second category, the targetless methods, eliminates the need for predefined targets by estimating extrinsic parameters by extracting useful information from surrounding environments automatically[4]. Muñoz-Bañón et al.[7] introduced a method for calibrating the extrinsic parameters of camera and LiDAR by utilizing local edge features in arbitrary environments, allowing for the calculation of 3D-to-2D errors between the data from both sensors. To minimize these errors, they employed the perspective-three-point (P3P) algorithm in their solution. Song et al.[8] developed a method called Galibr, which uses ground planes and edge information to calibrate the extrinsic parameters of the camera and LiDAR.

Lastly, learning-based methods leverage advancements in machine learning and require no artificial definition of features, which can learn useful information using neural networks to automate and enhance the calibration process[9]. Schneider et al.[10] introduce RegNet as the first method for calibration that employs a deep convolutional neural network (CNN), extracting and matching features with a network before regressing the calibration parameters. CalibRCNN[11] and CalibDepth[3] incorporate both CNN and recurrent networks, such as LSTM, to perform calibration not only using spatial features, but also temporal features. Xiao et al.[12] incorporate transformer architecture inside of the deep learning model which will extract and leverage correlation features with higher contributions. Together, these methodologies represent a comprehensive framework for addressing the challenges of LiDAR-camera calibration.

In contrast to these methods, our approach emphasizes developing a deep learning model tailored for practical, on-the-field implementation. To achieve this, we focus on designing a lightweight deep learning model that meets the requirements for efficiency and real-world usability for calibrating the extrinsic parameters in real time.

Ⅲ. Proposed Methods

In this approach, a deep neural network architecture is proposed to estimate the extrinsic calibration parameters between LiDAR and camera sensors. Unlike conventional methods, the proposed framework introduces a significantly more lightweight prediction model with only ~4 million parameters, making it well-suited for real-world applications where computational efficiency is a critical consideration. In this section, the problem definition, the architecture and the functionality of each part, and loss function utilization, will be elaborated.

3.1 Problem Definition

Extrinsic calibration in camera-LiDAR system is the determination of the extrinsic parameters, which represent the relative pose or rigid transformation between a LiDAR and a camera placed on the same platform or vehicle. Extrinsic parameters are used to transform the coordinate frame of one sensor to the other, which is important in sensor fusion. To obtain a 3D LiDAR point coordinate relative to the camera, the following equation is used:

$$P_C = [R|t] \cdot P_L = T \cdot P_L \tag{1}$$

where $P_c : (X_c, Y_c, Z_c)$ is the camera coordinate system, $P_L : (X_L, Y_L, Z_L)$ is the LiDAR coordinate system, T are the extrinsic parameter matrix, R is the rotation matrix, and t is the translation vector of the LiDAR-camera system.

In this study, the proposed extrinsic calibration method aims to predict the amount of misalignment in the extrinsic parameters. This misalignment is defined as the difference or shift from the known initial extrinsic parameters T_{known} to the actual extrinsic parameters T_{actual}. The drift in extrinsic parameters occurs during vehicle operation, mostly caused by vibrations, vehicle motions, and temperature changes, which accumulate over time. The accumulated drift in extrinsic can be expressed as:

$$T_{actual} = \Delta T \cdot T_{known} \tag{2}$$

where ΔT is the extrinsic parameter misalignment caused by this accumulated drift, denoted as rigid transformation between T_{known} and T_{actual}. The proposed method will estimate $\Delta T_{predicted}$ to cancel the misalignment term ΔT and calibrate the extrinsic parameters using the following equation:

$$T_{actual} = \Delta T_{predicted}^{-1} \cdot \Delta T \cdot T_{known} \quad (3)$$

If the prediction is accurate, $\Delta T_{predicted}^{-1}$ and ΔT will cancel each other, making T_{known} and T_{actual} equal.

3.2 Network Architecture

The proposed calibration network is composed of three main components: two branches of a feature extraction network, a feature matching layer, and fully connected layers. As all the parameters within these components are differentiable, the overall network can be trained end-to-end and simultaneously. The overall scheme of the proposed method is depicted in Figure 1.

The first part is the *feature extraction network*, which consists of two branches designed to extract features from the RGB images and the depth images from the point cloud data projection. The feature extraction for each branch is shown in Figure 2. The

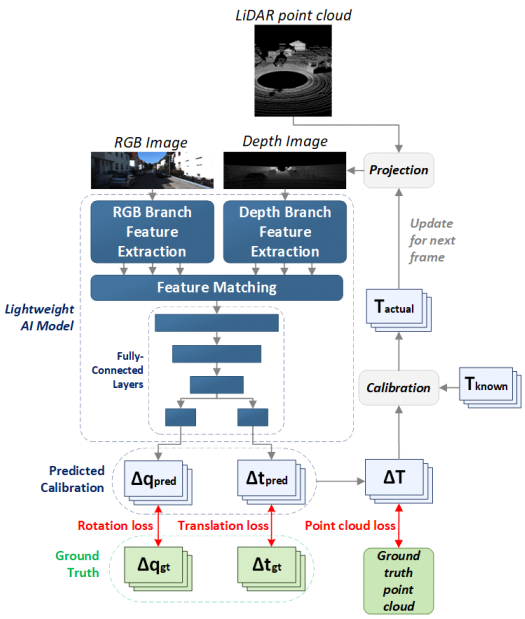

Fig. 1. The overall architecture of the proposed method.

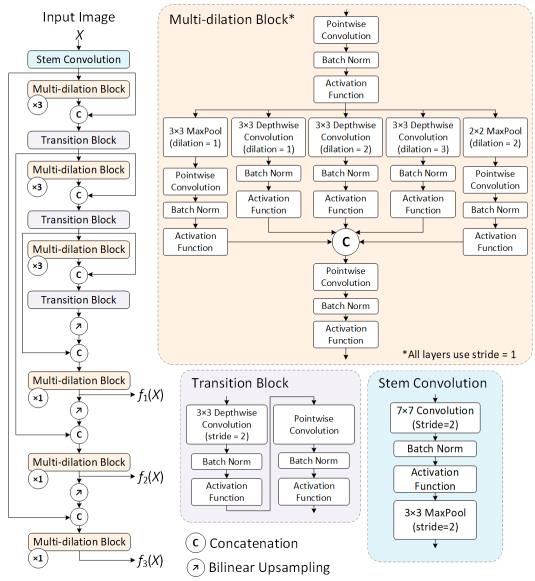

Fig. 2. The architecture of the feature extraction.

feature extraction comprises the multi-dilation block. The proposed multi-dilation block consists of several depthwise convolution layers that use various dilation rates to allow more efficient multiscale feature extraction.

Dilated convolution offers several advantages over regular convolution. By introducing gaps between kernel elements, it expands the receptive field without increasing the number of parameters, allowing the network to capture wide-range dependencies efficiently. Additionally, dilated convolutions help preserve spatial resolution by avoiding excessive downsampling. However, dilated convolution with larger dilation may cause gridding artifact problems, where some information is lost or overlapping in several pixels[13]. Thus, multiple dilation rates are applied to overcome the gridding artifacts problem. Additionally, multiple dilation rates can provide multiple resolution quality in the feature extraction, allowing richer representation in the feature maps. By using multiple dilation rates, sparse features of depth images can be extracted more effectively. The proposed feature extraction also obtains three feature map resolutions to be utilized as coarse-to-fine feature matching in the next part of the network. These multiresolution feature maps can be denoted as $f_1(X)$, $f_2(X)$, and $f_3(X)$, from smallest

resolution to largest resolution, respectively, where X is the input image (RGB image X_{RGB} and depth image X_{depth}). Other parts of the feature extraction, transition block and stem convolution, are used to reduce the resolution of the feature maps.

The second part, the *feature matching layer*, compares the obtained feature maps from both branches by calculating the 3D correlation cost of corresponding pixels in RGB and depth feature maps (X_{rgb} and x_{depth}). The feature matching part is shown in Figure 3. This part was inspired by the optical flow estimation scheme of PWC-Net[14]. A correlation layer is used to compute the cost volume $c(x_1, x_2)$ as the correlation between the flattened feature vectors of X_{rgb} and x_{depth}, which is expressed as follows:

$$C\left(x_{rgb}(p_i), x_{depth}(p_j)\right) = \bigcup i,j \in D\left(\left(x_{rgb}(p_i)\right)^T, x_{depth}(p_j)\right) \quad (4)$$

To reduce complexity, a local cost volume is computed within a small disparity range ($d = 9$). The resulting cost volume has dimensions $d^2 \times H \times W$ where H and W are the height and width of the feature maps. The cost volume calculation is performed in three stages, each for $f_1(X)$, $f_2(X)$, and $f_3(X)$, where the feature matching is performed from fine features to coarse features. For every stage, the pixel optical flow

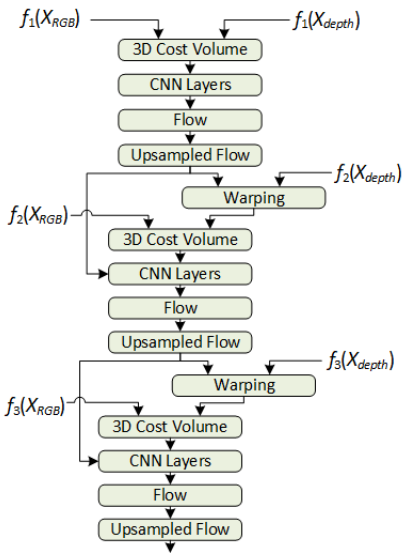

Fig. 3. Feature matching architecture.

is predicted using a CNN layer which determines the misalignment in both input feature maps. Feature warping is also performed to match the predicted flow from the previous stage to the next one.

The third part, *fully-connected layers*, predicts the transformation matrix between LiDAR and camera sensors from the feature matching result of the RGB branch and depth branch. The network includes three layers of fully connected layers, followed by two separate branches, each comprising stacked fully connected layers for estimating rotation and translation. The network outputs a 1×3 translation vector and a 1×4 rotation quaternion.

3.3 Loss Function

For training, the model utilizes an input pair consisting of an RGB image and a misaligned depth image. Three types of loss functions are employed: a translation loss (L_T), a rotation loss (L_R), and a point cloud distance loss (L_P).

$$L = \lambda_T L_T + \lambda_R L_R + \lambda_P L_P \quad (5)$$

where λ_T, λ_R, and λ_P denotes respective loss weight.

The smooth L1 loss is applied to the translation vector (t_{pred}). Because it provides a smoother gradient near zero due to the incorporation of a squared term. For the rotation loss (L_R), the quaternion angular distance form is used instead of Euclidean distance, as quaternions represent directional information, and Euclidean distance fails to capture their differences accurately. The angular distance is defined as follows:

$$L_R = D_a\left(q_{gt}, q_{pred}\right) \quad (6)$$

where q_{gt} is the ground truth of quaternion, q_{pred} is the prediction, and D_a is the angular distance of two quaternions.

Besides the regression loss, we also compute the point cloud distance in the loss function by computing the distance between the predicted point cloud and ground truth point cloud for each of the batch data using the L2 normalization equation and dividing it by the amount of the overall data. The point cloud loss is expressed as follows:

$$L_P(P_{pred}, P_{gt}) = \frac{1}{N}\sum_{i=1}^{N}\left\|P_{pred\ i} - P_{gt\ i}\right\|_2 \quad (7)$$

This type of loss function, which combines translation loss, rotation loss, and point cloud distance loss, is widely used in 3D computer vision, including extrinsic calibration. Related works such as RegNet[10], CalibDNN[15], CalibNet[16], and PSNet[2] employ similar combined loss functions. Inspired by these methods, we adopt this approach.

The concept of this loss function originates from the PoseNet[17] paper (for rotation and translation loss) and the work of Fan et al.[18] (for point cloud distance loss). These papers discuss the impact of these loss components in detail. According to the PoseNet paper, combining translation and rotation loss is essential, as regressing position and orientation separately leads to poor performance. Additionally, the CalibNet paper highlights that incorporating point cloud distance loss significantly reduces translation error by ensuring proper alignment in 3D space

Ⅳ. Experiment and Result

4.1 Dataset

The KITTI odometry dataset[19] was used to train the proposed model. The KITTI dataset comprises diverse driving environments in 22 driving sequences. The dataset is split into a training set, which consists of sequence 01-21 with 39,011 frames, and a validation and test set, which consists of 4541 frames of sequence 00. To simulate the misalignment condition in the LiDAR-camera system, the extrinsic parameters are misaligned within the range of 0.25 m and 10°.

4.2 Training Environment

The training process was conducted using an Intel Xeon Silver 4215R CPU and an NVIDIA Titan XP GPU. The model was implemented with Python v3.10 and the PyTorch v2.2 library, utilizing CUDA v11.8 and cuDNN v11.8. The training dataset was processed in batches of 16 over 200 epochs. At the start of the training, the learning rate is set to 10^{-4} and reduced by a factor of 0.1 if the validation loss stagnates for 10 consecutive epochs. Adam optimizer is also used to update the model weights during training.

4.3 Evaluation Metrics

The performance of the proposed method is evaluated by measuring the rotation and translation errors of the predicted extrinsic parameters. Absolute translation errors are evaluated by measuring the Euclidean distance between predicted translation vectors and ground truth translation vectors. The absolute translation error is expressed as follows:

$$E_t = \left|t_{pred} - t_{gt}\right| \quad (8)$$

Absolute rotation errors are evaluated by measuring the Euler angle difference (E_{yaw}, E_{pitch}, and E_{roll}) between predicted rotation and ground truth rotation.

4.4 Results and Discussions

The performance of the proposed method was assessed using the KITTI odometry dataset with a misalignment range of 0.25 m and 10°. This method employs a one-shot approach, which runs the model only once per frame. This approach ensures good real-time performance, efficient data processing, and efficient inference. The performance of the proposed method is then compared to several existing methods.

Table 1 presents the performance comparison in terms of translation errors and Table 2 presents the performance comparison in terms of rotation errors. The performance of all existing methods, except CalibDepth[3], were evaluated on the same misalignment range with the proposed method (0.25 m and 10°), whereas CalibDepth were evaluated on misalignment range of 1.5 m and 20°. For CalibRCNN[11], CALNet[20], and PSNet[2], we include the results presented in the CalibFormer paper[12].

Based on Table 1, the proposed method achieved the best accuracy in all translation errors metrics, whereas, based on Table 2, the proposed method achieved best accuracy in one rotation errors metrics, which is the pitch rotation error. The proposed method shows exquisite performance, particularly in translation accuracy, whereas in terms of rotation accuracy, the proposed method still requires improvements while it is still comparable to the existing methods.

Table 1. The translation performance comparison with existing methods.

Method	Translation (cm)			
	Mean	X	Y	Z
CalibRCNN [11]	5.3	6.2	4.3	5.4
CalibDNN [15]	5.07	3.8	1.8	9.6
CALNet [20]	3.03	3.65	1.63	3.80
PSNet [2]	3.07	3.8	2.8	2.6
CalibFormer [12]	1.19	1.10	0.90	1.56
CalibDepth [3]	1.17	1.31	1.02	1.17
Proposed	**0.61**	**1.08**	**0.18**	**0.56**

Table 2. The rotation performance comparison with existing methods.

Method	Rotation (degree)			
	Mean	Roll	Pitch	Yaw
CalibRCNN [11]	0.428	0.199	0.64	0.446
CalibDNN [15]	0.3	0.11	0.35	0.44
CALNet [20]	0.20	0.10	0.38	0.12
PSNet [2]	0.15	**0.06**	0.26	0.12
CalibFormer [12]	0.141	0.076	0.259	0.087
CalibDepth [3]	**0.123**	0.064	0.226	**0.080**
Proposed	0.154	0.182	**0.139**	0.141

The slight inferiority in rotation accuracy might be caused by the smaller model size compared to the existing method and the utilization of non-pre-trained feature extraction, unlike CalibFormer[12] which uses pre-trained ResNet18. However, the rotation accuracy of the proposed method is still applicable to the actual application. Additionally, we focus on small model size that allows extrinsic calibration to be performed on smaller onboard computers.

The performance of the proposed method is also evaluated qualitatively by observing the comparison between the projected depth image onto RGB camera image before and after the calibration. The visual result is presented in Figure 4, where the misaligned input, calibrated image, and the ground truth are presented. Figure 4 shows the proposed method is capable in aligning the point cloud to match the original position and orientation shown in the ground truth. The point cloud projections of the surrounding objects and landmarks (such as trees, cars, and poles) are well-aligned with their corresponding camera images.

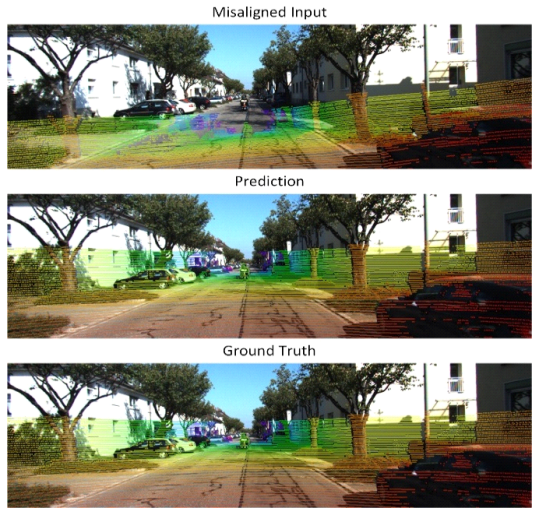

Fig. 4. The calibration results visualization of the proposed method.

We also compare the number of parameters with the existing methods, as presented in Table 3. For this comparison, only PSNet[2] and CalibDepth[3] mentioned the number of parameters in their articles explicitly, where PSNet only specified the size of its one branch of feature extraction part. Other methods, such as CalibRCNN[11], CalibDNN[15], CALNet[20], and CalibFormer[12], mentioned the utilization of ResNet-18, which has around ~11 M parameters. These methods are not included in Table 3 because the exact numbers are not mentioned explicitly in their articles. Comparing the proposed method with methods included in the Table 3 and aforementioned excluded methods, it shows that the proposed method utilized a much lower number of parameters while still outperforming these existing methods, especially in translation parameters.

In terms of real time performance, the proposed method completes calibration for each frame in ap-

Table 3. Number of parameters comparison with existing methods.

Methods	Number of parameters
PSNet (one branch of feature extraction part only) [2]	13.5 M
CalibDepth [3]	43 M
Proposed	4.03 M

Table 4. Computational cost analysis of the proposed method.

Parameter count	Average inference time (ms)	GPU Memory (MB)	Storage (MB)
4.03 M	30	900	48

proximately 30 milliseconds, making it well-suited for real-time applications. The computational analysis of the proposed model is included in Table 4.

V. Conclusion

This study proposes a deep learning-based targetless LiDAR-camera extrinsic calibration for autonomous vehicle application. The proposed method utilizes a novel feature extraction method comprising multi-dilation blocks that can extract sparse and fine-grained features of both RGB camera image and LiDAR depth image. The proposed method also consists of only 4 million parameters, making it lightweight. The model is also executed in a one-shot approach, allowing for efficient real-time applications.

The performance of the proposed method is also evaluated under initial misalignment of up to 0.25 m and 10.0°. It outperforms several existing methods, especially in translation errors. The proposed method achieved translation errors of 1.08 cm, 0.18 cm, and 0.56 cm for *X*-axes, *Y*-axes, and *Z*-axes, respectively. It also achieved rotation errors of 0.182°, 0.139°, and 0.141°, for roll, pitch, and yaw, respectively. In terms of rotation accuracy, our proposed method still requires some further improvements, such as training on more datasets, improving the training methods, or incorporating advanced deep-learning architecture, such as transformers, where a lightweight self-attention mechanism needs to be developed.

For future directions, we plan to expand the application of this deep learning based extrinsic calibration for adverse driving conditions, such as rain, snow, fog, or night driving conditions, by training the model on other available datasets supporting these scenarios. Additionally, this calibration method can be implemented on other modes of autonomous vehicles, such as unmanned aerial vehicle (UAV), where lightweight deep learning models become crucial. Furthermore, we aim to optimize the size of the deep learning network by not only considering the number of trainable parameters but also other computational cost metrics, such as float precision, FLOPs, inference time, and more.

References

[1] P. An, et al., "Geometric calibration for LiDAR-camera system fusing 3D-2D and 3D-3D point correspondences," *Opt Express*, vol. 28, no. 2, p. 2122, Jan. 2020. (https://doi.org/10.1364/OE.381176)

[2] Y. Wu, M. Zhu, and J. Liang, "PSNet: LiDAR and camera registration using parallel subnetworks," *IEEE Access*, vol. 10, pp. 70553-70561, 2022. (https://doi.org/10.1109/ACCESS.2022.3186974)

[3] 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)

[4] Z. Tan, X. Zhang, S. Teng, L. Wang, and F. Gao, "A review of deep learning-based LiDAR and camera extrinsic calibration," *Sensors*, vol. 24, no. 12, p. 3878, Jun. 2024. (https://doi.org/10.3390/s24123878)

[5] E. Giacomini, L. Brizi, L. D. Giammarino, O. Salem, P. Perugini, and G. Grisetti, "Ca2Lib: Simple and accurate LiDAR-RGB calibration using small common markers," *Sensors*, vol. 24, no. 3, p. 956, Feb. 2024. (https://doi.org/10.3390/s24030956)

[6] S. Jeong, S. Kim, J. Kim, and M. Kim, "O^3 LiDAR-camera calibration: One-shot, one-target and overcoming LiDAR limitations," *IEEE Sensor J.*, vol. 24, no. 11, pp. 18659-18671, 2024. (https://doi.org/10.1109/JSEN.2024.3390170)

[7] M. A. Munoz-Banon, F. A. Candelas, and F. Torres, "Targetless camera-LiDAR calibration in unstructured environments," *IEEE Access*,

vol. 8, pp. 143692-143705, 2020. (https://doi.org/10.1109/ACCESS.2020.3014121)

[8] W. Song, M. Oh, J. Lee, and H. Myung, “Galibr: Targetless LiDAR-camera extrinsic calibration method via ground plane initialization,” in *2024 IEEE Intell. Veh. Symp. (IV)*, pp. 217-223, 2024. (https://doi.org/10.1109/IV55156.2024.10588429)

[9] X. Li, Y. Xiao, B. Wang, H. Ren, Y. Zhang, and J. Ji, “Automatic targetless LiDAR-camera calibration: A survey,” *Artificial Intell. Rev.*, vol. 56, no. 9, pp. 9949-9987, Nov. 2022. (https://doi.org/10.1007/s10462-022-10317-y)

[10] N. Schneider, F. Piewak, C. Stiller, and U. Franke, “RegNet: Multimodal sensor registration using deep neural networks,” *arXiv preprint arXiv:1707.03167*, 2017. (https://doi.org/10.48550/ARXIV.1707.03167)

[11] J. Shi, et al., “CalibRCNN: Calibrating camera and LiDAR by recurrent convolutional neural network and geometric constraints,” in *2020 IEEE/RSJ Int. Conf. IROS*, pp. 10197-10202, Oct. 2020. (https://doi.org/10.1109/IROS45743.2020.9341147)

[12] Y. Xiao, Y. Li, C. Meng, X. Li, J. Ji, and Y. Zhang, “CalibFormer: A transformer-based automatic LiDAR-camera calibration network,” in *2024 IEEE ICRA*, pp. 16714-16720, May 2024. (https://doi.org/10.1109/ICRA57147.2024.10610018)

[13] F. Yu, V. Koltun, and T. Funkhouser, “Dilated residual networks,” in *2017 IEEE CVPR*, pp. 636-644, Jul. 2017. (https://doi.org/10.1109/CVPR.2017.75)

[14] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, “PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume,” *arXiv preprint arXiv:1709.02371*, Sep. 2017. (https://doi.org/10.1109/CVPR.2018.00780)

[15] 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*, 2021 (https://doi.org/10.48550/ARXIV.2103.14793)

[16] G. Iyer, R. K. Ram, J. K. Murthy, and K. M. Krishna, “CalibNet: Geometrically supervised extrinsic calibration using 3D spatial transformer networks,” in *2018 IEEE/RSJ Int. Conf. IROS*, pp. 1110-1117, 2018. (https://doi.org/10.1109/IROS.2018.8593693)

[17] A. Kendall, M. Grimes, and R. Cipolla, “PoseNet: A convolutional network for real-time 6-DOF camera relocalization,” in *2015 IEEE ICCV*, pp. 2938-2946, Dec. 2015. (https://doi.org/10.1109/ICCV.2015.336)

[18] H. Fan, H. Su, and L. Guibas, “A point set generation network for 3D object reconstruction from a single image,” *arXiv preprint arXiv:1612.00603*, Dec. 2016. (https://doi.org/10.1109/CVPR.2017.623)

[19] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving? The KITTI vision benchmark suite,” in *CVPR*, 2012. (https://doi.org/10.1109/CVPR.2012.6248074)

[20] H. Shang and B.-J. Hu, “CALNet: LiDAR-camera online calibration with channel attention and liquid time-constant network,” in *2022 26th ICPR*, pp. 5147-5154, Aug. 2022. (https://doi.org/10.1109/ICPR56361.2022.9956145)

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]