Calibration-Net:LiDAR and Camera Auto-Calibration using Cost Volume and Convolutional Neural Network

1st An Nguyen Duy
Department of Information Communication Convergence
Soongsil University
Seoul, Korea Republic of.
nguyenduyan710@gmail.com

2nd Myungsik Yoo
School of Electronic Engineering
Soongsil University
Seoul, Korea Republic of.
myoo@ssu.ac.kr

***Abstract*—A fusion of multi-sensor has been utilized widely for improving the environment perception in autonomous vehicles and robot navigation. Calibration is an essential procedure for preprocessing the data fusion between multiple sensors. Most target-based calibration techniques require manual works and specific calibration targets to achieve high accuracy. It gradually becomes outmoded for Light Detection and Ranging (LiDAR) and camera with the development of deep learning techniques. This paper proposed an online LiDAR-camera calibration that automatically predicts the extrinsic parameters by taking advantage of convolutional neural networks (CNNs). We take depth maps of stereo camera prediction and depth maps of the LiDAR projection as two separated branches as inputs for the proposed network. Unlike the current CNN-based calibration method, we construct a cost volume of the correlation between two corresponding pixels of depth maps in stereo camera and LiDAR, respectively. The proposed model gains a reasonable capability to adjust to different initial calibration error ranges. We evaluate the proposed architecture on the KITTI dataset and achieve 0.378 degree in rotation error and 2.353cm translation error.**

***Index Terms*—sensor fusion, LiDAR, stereo camera, supervised learning, deep-learning,**

I. Introduction

In the last decade, multi-sensor fusion has developed rapidly in many applications such as object detection, 3D reconstruction, classification, and depth prediction. LiDAR and camera are the most widely used sensors to provide accurate and stable perception for the surrounding environment. While LiDAR obtains the spatial depth information with high accuracy, it lacks color and texture information at low resolution. The camera brings the benefits of high-resolution RGB images but no distance information. Therefore, calibration is an essential preprocessing of data fusion to ensure the precision to transform the LiDAR coordinate to camera coordinate.

Most early LiDAR-camera calibration methods [1-3] used specific calibration targets and complex manual setups to extract the 2D-3D corresponding feature to find the external parameters. However, these methods operate offline and are not suitable for running in real-time for the autonomous vehicle. Deep learning techniques [4-7] are raising currently to give accurate calibration between LiDAR-camera through the driven-data source collected in real-time (KITTI) [9] which does not require any specific calibration target or manual setups. In this paper, we proposed a novel deep learning network to automatically estimate the 6 DoF transformation.

Our design consists of a stereo depth map estimated by a stereo camera and a LiDAR depth map projected from the point cloud as inputs. The network extracts multiscale features, the correlation layer is used to match the information from both multiscale features of LiDAR and stereo camera. Then, to predict the transformation, we stack two fully connected layers for global regression and a loss function to optimize the learning process.

II. METHODOLOGY

In this section, we introduce our proposed model for estimating extrinsic calibration with stereo and LiDAR depth maps as inputs. This work aims to find the rigid transformation by minimizing the loss function compared to the ground truth. The ground truth consists of projected depth maps from the 3D point clouds to the image plane, which provides the same depth at each arbitrary pixel location of the depth map derived from the stereo camera. The data representation of the network inputs, the network architecture, and the training are discussed in detail in the following sections.

A. Data Representation

The first input of our network is the stereo depth maps. Based on the known intrinsic and extrinsic parameters of stereo cameras, the depth information of a point from the left-right camera can be calculated as the following equation:

$$depth = \frac{B.f}{disparity} \tag{1}$$

where B and f are the baseline and focal length of the stereo camera, respectively, and the disparity is the difference of two corresponding pixels present for the same point in the world coordinate. By given initial LiDAR-camera transformation H_{init} and camera intrinsic K, we can project each 3D point cloud $P_i = [X_i, Y_i, Z_i] \in \mathbb{R}^3$ into a virtual image plane

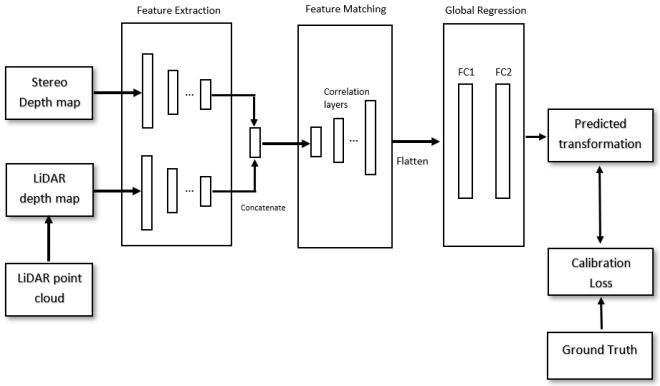

Fig. 1. The working flow of the proposed model.

with corresponding pixels $p_i = [u_i, v_i] \in \mathbb{R}^2$. The projection process is described as follows:

$$Z_i^{init}.p_i = K.H_{init}.P_i$$
$$H_{init} = \begin{bmatrix} R_{init} & t_{init} \\ 0 & 1 \end{bmatrix} \quad (2)$$

where R_{init} and t_{init} are the initial rotation matrix and translation vector of the transformation H_{init}. At each pixel p_i the depth value Z_i^{init} is preserved. If the pixel does not match any LiDAR point, this pixel will be set as zero.

B. Network Architecture

The proposed network architecture includes of three parts to solve the tasks of feature extraction, feature matching and global regression of the calibration. Since the three parts are merged as one CNN but with different parameter for each part, the network can be trained end-to-end. The working flow of the network is shown in the Fig.1 and the function of each part will be described as following section

1) Feature extraction: The depth maps of the RGB prediction and LiDAR projection are calculated individually as mention in the previous section. However, the formats of the depth maps are different because of two different sensor modalities. Since the depth map are pre-processing individually mentioned in previous section. The data is calculated in different senors with different modalities. There are two parallel feature extraction network to use to extract the rich features and reduce their dimensions. The output of the final feature maps will be down-sampled by six times and has 196 channels which extract the high-level features of the original inputs.

2) Feature Matching: The feature maps are concatenated along the channel dimension after extracting features from both input modalities. This network is motivated by PWC-Net [12] who introduces a correlation layer for feature matching. A cost volume is constructed to calculate the matching cost for connecting depth value of a pixel in the RGB depth prediction branch x_D^{rgb} with its corresponding pixel in depth feature maps projected from LiDAR x_D^{lidar}. The matching cost can be defined as:

$$cv(p_1, p_2) = \frac{1}{N}(c(x_D^{rgb}(p_1)))^T c(x_D^{lidar}(p_2)) \quad (3)$$

where $c(x)$ is the flattened vector of the feature map x and T is the transpose operator, N is the length of the column vector c(x). For different level of the pyramid layer setting, the cost volumes is needed to compute with a limited range of d pixels, i.e., $|p_1 - p_2|_\infty \leq$ d. The size of feature maps (conv6 in PWC-Net) are very small. Therefore, we set the value of the range d to be small. The dimension of the 3D cost volume $cv(p_1, p_2)$ is $d^2 \times$ H $\times$ W, where H and W are denoted as the height and width of the final pyramid feature maps x_D^{rgb} and x_D^{lidar}, respectively.

3) Global Aggregation: According to the regression network, it consits of two fully connected layers with 512 neurons and 256 neurons to regress the rotation and translation. The output of the network is 1x4 rotation r_{pred} and 1x3 translation vector t_{pred}. The results of the estimated rotation r_{pred} and the estimated translation t_{pred} can be evaluated by calculating the loss function compared to the ground truth with well-calibrated scenes.

C. Loss Function

Given an input pair of a depth image predicted by RGB image D_{rgb} and a depth image projected from LiDAR point cloud D_{lidar}, we used the following loss function described as Eq. (4):

$$\mathcal{L}(D_{rgb}, D_{lidar}) = \lambda_1 \mathcal{L}_r(D_{rgb}, D_{lidar}) + \lambda_2 \mathcal{L}_t(D_{rgb}, D_{lidar}) \quad (4)$$

where the $\mathcal{L}_r(D_{rgb}, D_{lidar})$ is the rotation loss and the $\mathcal{L}_t(D_{rgb}, D_{lidar})$ is the translation loss, λ_1 and λ_2 denote the respective loss weight to the rotation and translation loss. According to the rotation loss, the predicted rotation and the ground truth present in quaternions which are difficult to evaluate through Euclidean different distance between prediction and the ground truth. Therefore, we need to present the difference between quaternions into angular distance to evaluate the rotation loss:

$$\mathcal{L}_r = \mathcal{D}_a(r_g t, r_{pred}) \quad (5)$$

where r_{gt} and r_{pred} are the ground truth and prediction of quaternion, respectively. $\mathcal{D}_a$ is the angular distance of two quaternions [4] For the translation lossm we use a smooth $\mathcal{L}_1$ loss [8] which is much smoother regarding to the square function's usage near zero.

III. Experiment and Result

A. Experimental Setup

1) Dataset: Our proposed network is evaluating on the raw branch of the KITTI dataset [9]. The ground truth of the extrinsic parameters are provided by [11] for each sensor. The camera depth will be generated by using the left and right images of the camera images. The depth maps of the LiDAR will be obtained by projecting point cloud into a virtual image

Fig. 2. The initial calibration

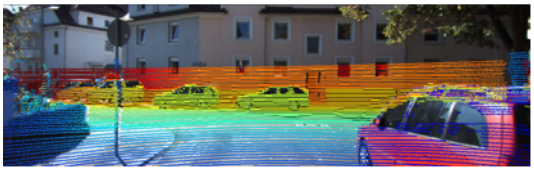

Fig. 3. The calibration result

plane with random initial transformation H_{init} and known intrinsic parameters of the camera K. There are 15967 frames for training and 4541 frames for testing. The testing set is spatially separated from the training set.

2) Evaluation Metrics: The calibration results are evaluated regarding to the rotation and translation errors of the predicted extrinsic parameters compared to the ground truth transformation. The rotation error can be described as follows:

$$E_r = \mathcal{D}_a(r_{gt} * inv(r_{pred})) \tag{6}$$

$$\mathcal{D}_a(m) = atan2(\sqrt{b_m^2 + c_m^2 + d_m^2}, |a_m|) \tag{7}$$

where $\{a_m, b_m, c_m, d_m\}$ is four components of the quaternion m, and * denotes as the quaternion multiplication and *inv* presents the inverse of a quaternion. The translation error is evaluated by the difference of the Euclidian distance between the predicted translation vector and the ground truth. It can be expressed as follows:

$$E_t = ||t_{gt} - t_{pred}||_2 \tag{8}$$

3) Training Details: We implemented our model with PyTorch (1.10.1) and trained on a RTX 3060 GPU. During the training, we choose Adam Optimizer [10] with learning rate $1e^{-4}$ with batch size 24 and total epoch 100.

B. Results and Discussion

In this section, the visual results of the calibration are shown in the Figs. 2 and 3.We sampled the decalibration in range of [-20°, 20°] / [-1.5m, 1.5m]. TABLE I expresses that our method is superior to other architecture due to the same training dataset. However, it still contains a large gap between our performance and the state-of-the-art CFNet in both rotation and translation errors. After investigating, the reason for the performance differences is the iterative calibration refinement. By predicting the calibration flow and valid 2D-3D corresponding set, CFNet applies the EPnP algorithm with the RANSAC scheme to refine the initial extrinsic parameters, which improves the extrinsic calibration accuracy after five times refinement. Despite not having the best performance, our proposed architecture points out some improvements compared to other models with a mean calibration error 0.378° in rotation and 2.353cm in translation.

TABLE I
COMPARISON WITH OTHER METHODS

Methods	Rotation (°)			Translation (cm)		
	Roll	Pitch	Yaw	X	Y	Z
CFNet [13]	0.059	0.110	0.092	1.025	0.092	1.042
Ours	0.105	0.21	0.19	2.82	2.35	1.89
CalibRCNN [14]	0.19	0.64	0.44	6.2	4.3	5.4
CalibNet [6]	0.15	0.9	0.18	4.2	1.6	7.22

IV. CONCLUSION

In this paper, we have proposed a novel method for 3D LiDAR-Camera extrinsic calibration using a deep neural network. By extracting the depth maps of the stereo camera and LiDAR point cloud, the model construct a cost volume between two depth maps. Our model achieves an improvement in performance comparing to other methods.

ACKNOWLEDGMENT

This research was supported by the National Research Foundation of Korea (NRF) grant funded by the Goverment of South Korea (MSIT)(NRF-2021R1A2B5B01002559)

REFERENCES

[1] S. Verma, J.S. Berrio, S. Worral, and E. Netbot, "Automatic extrinsic calibration between a camera and a 3D LiDAR using 3D point and plane correspondences," *arXiv preprint arXiv*:1904.12433, 2019

[2] P. An, T. Ma, K. Yu, B. Fang, J. Zhang, W. Fu, and J.Ma, "Geometric calibration for LiDAR-camera system fusing 3D-2D and 3D-3D point correspondence," *Optics Express*, vol. 28, no. 2, pp. 2122-2141, 2020

[3] A.-S. Vaida and S. Nedevschi, "Automatic extrinsic calibration of LiDAR and monocular camera images," in *IEEE 15th International Conference on Intelligent Computer Communication and Processing (ICCP)*, Cluj-Napaca, Romania, pp. 117-124, 2019

[4] Kendall, A. Grimes, M. and Cipolla,R. "Posenet: A convolutional network for real-time 6 DoF camera relocalization" in *Proceeding of the IEEE international conference on computer vision*, 2938-2946 (2020)

[5] Schneider, Piewak, F, Stiller,C and Franke,U. "Regnet: Multimodal sensor registration using deep neural networks," in *IEEE intelligent vehicles symposium (IV)*, 1803-1810 IEEE (2017)

[6] Iyer, G. Ram, R. Murthy, J.K and Krishna, "Calibnet: Geometrically supervised extrinsic calibration using 3D spatial transformer networks" arXiv preprint arXiv: 2018 *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1110–1117, IEEE (2018)

[7] Lv, X., Wang, B., Ye, D., and Wang, S., "Lidar and camera self-calibration using costvolume network," arXiv preprint arXiv:2012.13901 (2020).

[8] R. Girshick, "Fast R-CNN" in *Proceeding of the IEEE international conference on computer vision*, 2015, pp. 1440-1448

[9] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in CVPR, 2012.

[10] D. Kingma and J. Ba, "Adam: A method for stochastic optimization." arXiv preprint arXiv:1412.6980, 2014.

[11] Andreas Geiger, Frank Moosmann, and Bernhard Schuster ”Automatic camera and range sensor calibration using a single shot” in 2012 *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3936-3943, IEEE, 2012.2,5

[12] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In IEEE Conf. Comput. Vis. Pattern Recog., pages 8934–8943, 2018

[13] Lv, X.; Wang, S.; Ye, D. CFNet: LiDAR-Camera Registration Using Calibration Flow Network. Sensors 2021, 21, 8112. https://doi.org/10.3390/s21238112

[14] J. Shi et al., ”CalibRCNN: Calibrating Camera and LiDAR by Recurrent Convolutional Neural Network and Geometric Constraints,” 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2020, pp. 10197-10202, doi: 10.1109/IROS45743.2020.9341147.