2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

LiDAR-Camera Extrinsic Calibration with Hierachical and Iterative Feature Matching

Xuzhong Hu[1], Zaipeng Duan[1], Junfeng Ding[1], Zhe Zhang[1], Xiao Huang[2] and Jie Ma[1†]

***Abstract*— In autonomous driving, the LiDAR-Camera system plays a crucial role in a vehicle's perception of 3D environments. To effectively fuse information from both camera and LiDAR, extrinsic calibration is indispensable. Recently, some researchers have proposed deep learning-based methods that utilize convolutional networks to automatically extract features from LiDAR depth images and RGB images for calibration. However, these features do not sufficiently interact during feature matching, which limits the calibration accuracy. To this end, we introduce a novel extrinsic calibration network (HIFMNet) in this paper. It establishes a comprehensive connection between camera and LiDAR features by calculating a globally-aware map-to-map cost volume and hierachical point-to-map cost volumes. The former is used to regress large extrinsic offsets. The latter is employed to iteratively fine-tune extrinsic parameters, while the rigidity of LiDAR points is considered in each iteration to enhance regression robustness. Extensive experiments on the KITTI-odometry dataset demonstrate the superior performance of our HIFMNet compared to other state-of-the-art learning-based methods.**

I. INTRODUCTION

In the field of autonomous driving and robotics, multi-sensor fusion techniques have made remarkable advancements, greatly improving a system's adaptability to diverse environments. Among multi-sensor setups, LiDAR and camera exhibit a powerful synergy. Merging the rich texture features extracted from RGB images with the structural information derived from point clouds can significantly enhance the performance of numerous tasks, including object detection, semantic segmentation, and dense mapping. However, achieving this synergy necessitates precise LiDAR-Camera extrinsic calibration.

Early calibration methods rely on artificial objects such as chessboard to establish 2D-3D constraints. Other methods aim at automatic calibration by extracting edges or mutual information from arbitrary natural environments, but these approaches are time-consuming and often yield unstable results.

Recently, there has been a growing interest in learning-based methods due to their efficiency and effectiveness. These methods typically consist of two main procedures: feature extraction and feature matching. For example, as shown in Figure 1(a), RegNet [1] employs two separate network backbones with non-shared parameters to abstract features from RGB and depth images. During feature matching, the output feature maps are concatenated for regressing the rotation quaternion and the translation vector. However, just concatenating features along the channel dimension does not fully explore the similarities between different parts of RGB and depth images, resulting in non-ideal regression accuracy. Some researchers have noticed between optical flow estimation and extrinsic calibration tasks – both aim to establish correspondences in misaligned data. Therefore, they incorporate feature matching techniques from optical flow estimation models into extrinsic calibration networks. As shown in Figure 1(b), LCCNet [2] represents the relationship between LiDAR and camera information using cost volume. Nevertheless, focusing solely on low-resolution deep feature interaction may result in the loss of detailed feature similarities. As shown in Figure 1(c), CFNet [3] iteratively matches multi-level feature maps to estimate calibration flows fol-

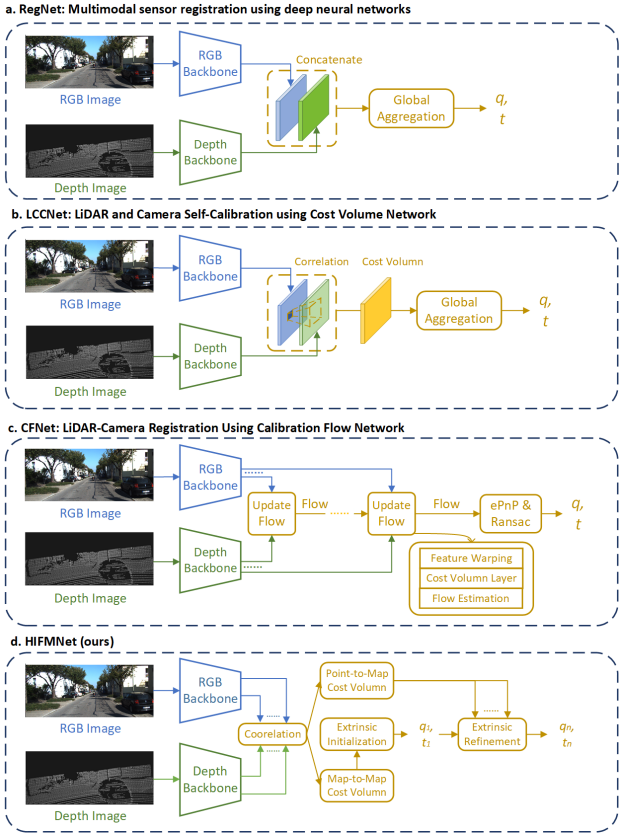


Fig. 1: Comparsion of learning-based LiDAR-Camera extrinsic calibration methods. The blue and green parts represent feature extraction branches of RGB and depth images. The yellow parts represent the feature matching and parameter regression process.

[1]School of Artificial Intelligence and Automation, Huazhong University of Science and Technology (HUST), Wuhan, Hubei 430074, China
[2]China ship Development and Design Center, Wuhan, Hubei 430064, China
† Corresponding author.

lowed by ePnP and RANSAC algorithms to optimize the extrinsic parameters. However, due to modality gaps, the clues for calibration flow prediction are much scarcer than those used for estimating optical flow between two consecutive RGB images. For example, in many scenarios, regions with high intensity gradients doesn't necessarily have large depth gradients. This misalignment results in significant flow errors in some parts of images. After feature warping, noise is introduced into the next feature matching, which worsens the subsequent flow prediction and ultimately impacts the calibration result

As shown in Figure 1(d), we propose a novel extrinsic calibration network, HIFMNet, which attempts to fully and robustly interacts features extracted from RGB and depth backbones. We integrate two key concepts into our method: hierachical feature interaction and progressive parameter optimization. During feature matching, they can explore multiscale similarities from the depth image and the RGB image and increase the localization accuracy of correspounding point-pixel pairs for iterative coarse-to-fine calibration. In addition, to mitigate noises caused by the aforementioned misalignment issue during iterations, we apply point reprojection in the update block to warp features rigidly with no need for calibration flows as intermediaries.

The contributions of this work is listed as follows:

- We propose a learning-based LiDAR-Camera extrinsic calibration method, HIFMNet, which achieves state-of-the-art accuracy on the KITTI odometry dataset.
- We introduce new feature matching paradigms for RGB and depth images: globally-aware map-to-map cost volume and hierachical point-to-map cost volumes. The former is responsible for extrinsic initialization, while the latter is used for iterative parameter optimization.
- We present an update block for robust parameter refinement. It contains point reprojection and feature rectification modules tailored for the calibration task.

II. RELATED WORK

In this section, we review LiDAR-Camera extrinsic calibration methods. They can be classified into three categories: target-based, targetless-based, and learning-based methods.

A. Target-based Methods

Target-based methods use various man-made objects, in which planar board [4] [5] or chessboard [6] [7] are widely applied. In the field of view of the camera and LIDAR, corner points, edges and planes are usually extracted as contraints for extrinsic optimization. To robustly establish 2D-3D constraints, some unique calibration objects are implemented. For instance, Garcia-Moreno et al. [8] devise hollow patterns to jointly calibrate camera intrinsic and LiDAR-camera extrinsic parameters. Pusztai et al. [9] extract vertexes of known-sized cubes for calibration, while Lee et al [10] determine spherical centers of balls for calibration. Both are unaffected by point cloud sparsity or partial occlusion as the corresponding points are obtained by surface fitting.

B. Targetless Methods

While driving, The drift of extrinsic parameters is inevitable. Hence, targetless methods are introduced to enable automatic online re-calibration in arbitrary environments.Pandey et al. [12] consider that both reflectance values of 3D LiDAR points and pixel intensities of RGB images are related to the material properties of objects. They employ mutual information (MI) to quantify this statistical relationship. The calibration task can be formulated as an optimization problem aimed at maximizing the mutual information function. To improve the robustness, some researchers incorporate surface normals [13] or semantic information [14] into MI. Except for calibration, MI can also be used for the localization technique [15]. When the LiDAR and camera are aligned, regions on the depth map with discontinuous depth are likely to concide with edges on the RGB image. Based on this observation, Levinson [16] and Castoria [17] accomplish extrinsic calibration through contour matching. Since straight lines are commonly found in driving scenarios, such as lanes and poles, Jiang [18] and Ma [19] extract corresponding 2D-3D line features for calibration.

C. Learning-based Methods

Using deep neural networks, stronger and more informative features can be automatically abstracted with no need for artificial objects or complex mathematical models. As a pioneering work, RegNet [1] leverages a convolutional neural network for regressing the 6-DoF extrinsic parameters between camera and LiDAR. CalibNet [21] introduces photometric and geometric losses to supervise training, which is beneficial in avoiding local optima. RGGnet [22] treats the calibration task as a geodesic distance optimization problem and utilizes Riemannian geometry and a generative model to construct a tolerance-aware loss function. Inspired by [25], Lv et al. [2] construct a cost volume that stores the matching costs between RGB and depth features. To improve the generalization, CFNet [3] predicts the calibration flows between LiDAR points and RGB pixels to calculate extrinsic parameters.

In most existing networks, feature encoders derive from well-established CNN backbones for other tasks. They are already capable of providing features with abundant information. In this work, we highlight the significance of effective interaction of these features for accurate calibration results.

III. METHODOLOGY

In this section, we will provide a detailed introduction to our method. The overall framework of HIFMNet is illustrated in Figure 2. It bears similarities to the optical flow estimation model RAFT [26]. Nevertheless, they are quite different in detail due to their distinct tasks. We will also carefully discuss these differences.

A. Projection of Point Cloud to Depth Map

Given the initial rotation quaternion q_{init} and translation vector t_{init} between LiDAR and camera, LiDAR points can

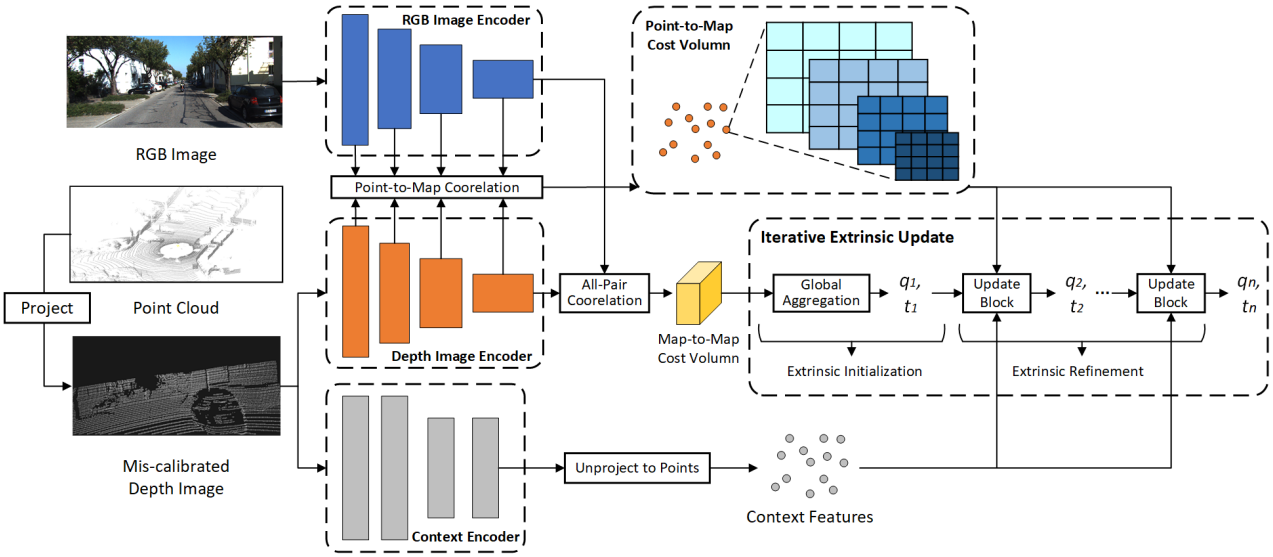


Fig. 2: Overview of HIFMNet. It takes a RGB image and a projected mis-calibrated depth image as input. Three encoders are implemented to extract their features. We obtain the map-to-map cost volume of the output feature maps from RGB and depth image encoders by all-pair correlation. The point-to-map cost volumes of multi-level features are also constructed. They along with context features are used for iterative extrinsic update including extrinsic initialization and refinement.

be transformed to the camera coordinate system using the following formula:

$$[0, P_n^C] = q_{init}[0, P_n^L](q_{init})^{-1} + [0, t_{init}] \tag{1}$$

Where $P_n^L = [x_n, y_n, z_n]$ represents the nth 3D point in the LiDAR coordinate system, and P_n^C in the camera coordinate system. The pinhole camera model is utilized to project P_n^C onto the camera plane:

$$p_n = K * P_n^C \tag{2}$$

Where, K and p_n denotes the camera intrinsic matrix, and the pixel coordinate. Sometimes, multiple points may be projected to the same pixel. We choose the nearest point. Points that fall outside image boundaries are filtered out. The remaining LiDAR points and their pixel positions are denoted as P^r and p^r. The z-coordinates of P^r are encoded into pixels to obtain the mis-calibrated depth image.

B. Feature Extraction

We employ two pre-trained ResNet-18 [27] backbones with non-shared parameters to extract features from RGB and depth images. Compared to feature encoders in RAFT [26], ours are deeper and have more parameters, which facilitates the alignment of heterogeneous data into a unified domain. We also retain the lightweight context encoding module from RAFT. It can provide additional spatial information for extrinsic regression.

Assuming the resolution of both the depth image and the RGB image is $H \times W$, they are resized to $\frac{H}{2} \times \frac{W}{2}$ and processed by a RGB image encoder and a depth image encoder. we collect feature maps from the last four convolutional modules, denoted as F_d^s and F_{rgb}^s, $s = 1, 2, 3, 4$. Their sizes are $\frac{H}{2^{s-1}} \times \frac{W}{2^{s-1}}$ with $h = \frac{W}{8}$ and $w = \frac{W}{8}$. The context encoder outputs a $h \times w$ map. We unproject its context features to points P^r by bilinear interpolation.

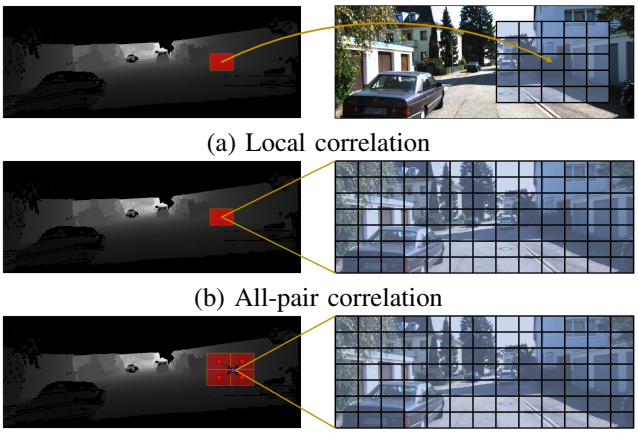
C. Cost Volume Construction

During feature matching, we calculate cost volumes to establish a relationship between F_d^s and F_{rgb}^s. It represents visual similarities between pixel features from depth and RGB images by computing correlation values [25]:

$$cv_s(p_1, p_2) = \frac{1}{N}(F_d^s(p_1))^T F_{rgb}^s(p_2) \tag{3}$$

Where p_1 and p_2 are pixel coordinates of $F_d^s(p_1)$ and $F_{rgb}^s(p_2)$, and N is the feature length. As shown in Figure 3 (a), in LCCNet [2], the cost volume for each depth image pixel is computed within a local window around its corresponding position in the RGB image. However, when there are significant extrinsic deviations, some corresponding pixels may fall outside the local window. As shown in Figure 3 (b), our method applies all-pair correlation to F_d^4 and F_{rgb}^4 to generate the map-to-map cost volume, which is a 4D tensor with dimension $\frac{w}{8} \times \frac{w}{8} \times \frac{h}{8} \times \frac{h}{8}$. It is then reshaped into a 2D feature map C_m for direct use in extrinsic estimation. This global feature interaction enables HIFMNet to handle larger offsets effectively.

As shown in Figure 3 (c), we also compute point-to-map cost volumes for iterative extrinsic refinement. The reason for building point-to-map cost volumes instead of map-to-map ones will be explained in Section III-D. We first map the depth features F_d^s back to the projected points p^r using bilinear interpolation, and then calculate their correlation values with all pixels in F_{rgb}^s. Applying this operation to each pair of feature maps, we can obtain hierachical cost volumes C_p^s, $s = 1, 2, 3, 4$ with size $N_p \times \frac{w}{2^{s-1}} \times \frac{h}{2^{s-1}}$, where N_p denotes the number of projected points p^r. In RAFT, the author only performs correlation on a pair of feature maps (equivalent to F_d^1 and F_{rgb}^1 in our network) to obtain the first cost volumn with the size $h \times w \times h \times w$. The



(a) Local correlation

(b) All-pair correlation

(c) Point-to-map correlation

Fig. 3: Comparsion of different correlation methods. The red and blue grids are pixels in depth and RGB feature maps. (a) LCCNet performs correlation only within a local windows. (b) We compute correlation values in an one-to-all manner, which can bulid long-range correspondences between depth and RGB features. (c) We first interpolate the feature of the projected point from surrounding pixels and then calculate the correlation values between it and all RGB features.

other three are computed by leveraging average pooling with kernel sizes $\{2, 4, 8\}$ on the last two dimensions. In contrast, our approach offers two advantages: (1) the deep feature interaction effectively bridges the modality gap between RGB and depth images. (2) In driving scenarios, objects vary significantly in size within the field of view based on their depths. Our parallel interaction among multi-level features allows the network to better exploit similarities at different scales.

D. Iterative Extrinsic Update

Based on cost volumes, we iteratively infer a sequence of extrinsic parameter residuals. It comprises two steps: extrinsic initialization and extrinsic refinement and we define the former the first update iteration.

Extrinsic Initialization: The reshaped map-to-map cost volume C_m is fed into a global aggregation layer. After passing through several convolutional blocks, the output feature map is reshaped into a 1D vector. Then it is send to several fully connected layers to produce a 1×3 translation vector t_1, and a 1×4 rotation quaternion q_1. **Extrinsic Refinement:** In RAFT, the author employs correlation lookup operation on cost volumes to retrieve costs within local windows for the prediction of optical flow residuals. These residuals are then used for the next lookup operation. This recurrent scheme is also beneficial for our calibration task since it's akin to conducting multiple fine-grained feature matching, increasing the search range and localization accuracy for corresponding points and pixels. However, as we argued in the introduction, directly adopting off-the-shelf optical flow estimation structures by defining so-called calibration flow may encounter the challenge of point-pixel misalignment.

Actually, optical flow estimation is a dense prediction task, whereas extrinsic calibration only requires regressing a few parameters. The fundamental distinction lies in the assumption of a rigid transformation between the entire LiDAR point cloud and the camera image plane. Based on this fact, we choose to directly regress a rigid offset in each iteration and use it to update the depth image for the next lookup operation. Thanks to the rigid constraint, even if there remains mismatched points, they won't be assigned abnormal offsets that carry noise to the next iteration. However, unlike RGB-D cameras, the LiDAR-derived depth image have sparse valid pixels. Not all pixels have the necessary depth values for a 3D rigid warping [28]. Therefore, we turn to warping LiDAR points, which is why we construct point-to-map cost volumes.

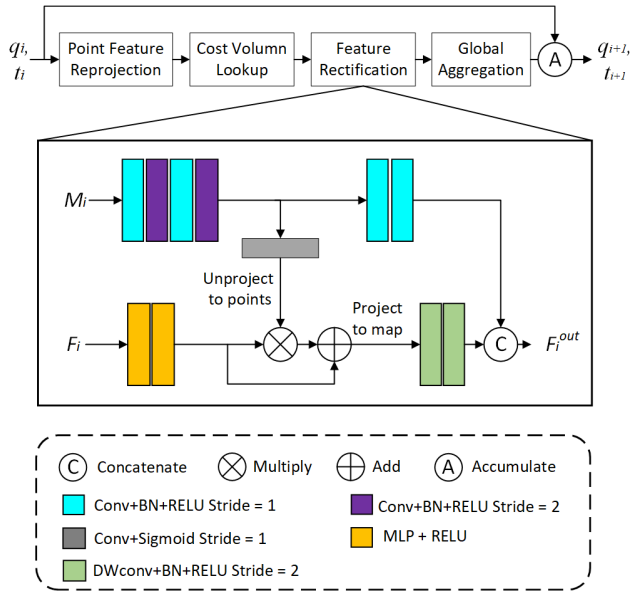


Fig. 4: The pipeline of the update block in extrinsic refinement, where the feature rectification module is shown in detail.

The pipeline of the update block is shown in Figure 4. These update blocks share weights between iterations. After the ith iteration, we first reproject LiDAR points using the following formulas:

$$[0|P^r_{n,i}] = q_i[0|P^r_{n,0}](q_i)^{-1} + [0|t_i] \tag{4}$$

$$p^r_{n,i} = KP^r_{n,i} \tag{5}$$

Where $P^r_{n,i}$ and $p^r_{n,i}$ denote new 3D and pixel positions of the nth LiDAR point. $P^r_{n,0}$ is the initial 3D coordinate. Based on all the new pixel positions p^r_i, a lookup operation is used to extract multi-scale correlation features from $C^1_p, C^2_p, C^3_p, C^4_p$. They are concatenated with context features F_c, resulting in combined feature vectors F_i.

However, F_i is obtained from the RGB image and initial miscalibrated depth image and may not be suitable for the current iteration. Therefore, we design a feature rectification module to address this issue. Like in [28], we encode each point's z-coordinate $P^r_{n,i}$ and pixel position displacement $p^r_{n,i} - p^r_{n,i}$, as well as current predicted extrinsic q_i, t_i, into

one vector. All these vectors containing motion information are projected onto a map M_i with the size $\frac{W}{2} \times \frac{H}{2}$ and multiple vectors in the same pixel are averaged. M_i is then downsampled to $h \times w$. After passing a sigmoid activation function, features are unprojected onto points p_i^r to reweight the channels of F_i. They are also projected onto a $\frac{W}{2} \times \frac{H}{2}$ map and downsampled to $h \times w$. It is fused densely with motion features through channel-wise concatenation. The output F_i^{out} is passed to a global aggregation layer to generate extrinsic residuals Δq_{i+1} Δt_{i+1} and we update extrinsic parameters according to following equations:

$$q_{i+1} = \Delta q_{i+1} q_i \tag{6}$$

$$[0|t_{i+1}] = \Delta q_{i+1}[0|t_i](\Delta q_{i+1})^{-1} \tag{7}$$

A noteworthy difference from RAFT is that both motion information vectors and combining features F_i are projected to updated positions p_i^r in each iteration instead of original positions p_0^r. It can make CNNs easier to estimate extrinsic residuals since the correct 2D-3D constraints are maintained.

E. Loss Function

For each iteration, we apply three losses to supervise the output accumulated extrinsic parameters: rotation loss, translation loss, and point distance loss:

$$L_i = L_i^p + L_i^r + L_i^t \tag{8}$$

Where the rotation loss L_i^r is defined as the quaternion distance between q_i and $(q_{init})^{-1}$, and the translation loss is represented as the L1 smooth loss between t_i and the inversed translation vector t_{inv}:

$$[0|t_{inv}] = -(q_{init})^{-1}[0|t_{init}]q_{init} \tag{9}$$

We use increasing loss weights for different iterations. The total loss is formulated as follows:

$$L_{total} = \sum_{i=1}^{N_u} \gamma^{n-i} L_i^p \tag{10}$$

In this paper, we set the number of update iterations $N_u = 4$ and the gain factor $\gamma = 0.9$.

IV. EXPERIMENTS

We evaluate our proposed method on the KITTI odometry dataset and also assess its transferability on the NuScenes dataset. This section introduces the dataset preparation and training details for HIFMNet, along with the analysis of quantitative and qualitative results from our experiments.

A. Dataset Preparation

The KITTI-odometry dataset comprises 21 sequences from different driving scenarios and provides groundtruth LiDAR-Camera extrinsic parameters for each frame. Similar to previous works [29]–[31], we use sequences 01 to 20 for the training set and sequence 00 for the testing set. For the NuScenes dataset, we only calibrate the extrinsic parameters between the front camera and the LiDAR. The network is evaluated on the validation set. During both training and testing, we add random perturbations in the range of [$\pm 20°$, $\pm 1.5m$] to groundtruths to obtain the initial rotation quaternions and translation vectors.

B. Training Details

The network is trained on a NVIDIA 4090 GPU with a batch size of 6 for 120 epochs, using the Adam optimizer. We first implement warmup with 10 epochs. We start with a warmup phase of 10 epochs and then adopt a cosine annealing learning rate strategy for our learning rate decay. The maximum and minimum learning rates are 3e-4 and 1e-6. Additionally, we also apply a model ensemble strategy for better performance when comparing with other methods. The first model is trained to handle perturbations in the range of [$\pm 20°$, $\pm 1.5m$]. Two additional models are trained to handle perturbations in the ranges [$\pm 5°$, $\pm 0.5m$] and [$\pm 1°$, $\pm 0.1m$], respectively. We obtain the smaller-range model by fine-tuning the former larger-range model for 50 epochs.

C. Results And Analysis

We compare HIFMNet with other learning-based methods. As shown in Table I, our method achieves a mean square translation error of 0.636 cm and a quaternion angle error of 0.0347°. The mean errors for x, y, z translation are 0.13 cm, 0.239 cm, and 0.54 cm, while the mean errors for roll, pitch, and yaw angles are 0.0081°, 0.0105°, and 0.0258°. HIFMNet achieves state-of-the-art calibration accuracy by only ensembling three models, while other methods except CalibNet [21] ensemble five ones. Compared to others, the main advantage of our method is the comprehensive interaction between depth and RGB image features, enabling the network to excavate more underlying geometric correspondences for extrinsic regression.

We also conduct an ablation study to further analyze the effect of our feature matching scheme. As shown in Table II, we set the iteration number to 1 (only maintain extrinsic initialization). The mean errors of all components except the roll angle are smaller than LCCNet's. This demonstrates the effectiveness of the globally-aware map-to-map cost volume. When increasing the iteration number, the mean errors continued to decrease, ultimately converging after 4 iterations. Similar to CFNet [3], we modify the network to regress calibration flows with 5 iterations and compute extrinsics using RANSAC and ePnP algorithms. However, it results in unsatisfactory performance. As illustrated in Figure 1, when warping LiDAR points to new positions according to predicted calibration flows, we observe distortions in some areas. This phenomenon still persists throughout the iterations, which have detrimental implications to extrinsic regression. On the other hand, Our approach maintains the rigidity of LiDAR points, enhancing the robustness of feature matching and extrinsic residual estimation at each step.

Table III reports a comparison of model parameters between HIFMNet and LCCNet [2]. LCCNet doesn't have the extrinsic refinement process. However, it applies dense connection in the global aggregation layer during extrinsic initialization, bring numerous parameters. In HIFMNet, the

Fig. 5: We show the qualitative results of HIFMNet and its calibration flow revision HIFMNet(flow). (a)(b) are calibration and flow estimation results of HIFMNet and HIFMNet(flow) after first extrinsic update iteration. We enlarge the areas within the red boxes. (c)(d)(e) and (f)(g)(h) are results after first, second and fourth iterations. Even with an increase in the iteration number of HIFMNet(flow), noticeable distortion patterns still exist.

TABLE I: Performance Comparsion On KITTI-Odometry Dataset Among the Different Learning-based Methods.

Method	Mis-calibrated range	Translation Error (cm)				Rotation Error (°)			
		E_t	X	Y	Z	E_r	Roll	Pitch	Yaw
RegNet [1]	$[-1.5m, 1.5m]/[-20°, 20°]$	–	7	7	4	–	0.24	0.25	0.36
CalibNet [32]	$[-0.2m, 0.2m]/[-10°, 10°]$	–	4.2	1.6	7.22	–	0.18	0.9	0.15
LCCNet [2]	$[-1.5m, 1.5m]/[-20°, 20°]$	1.588	0.243	0.38	0.459	0.163	0.03	0.019	0.04
RLCA [29]	$[-1.5m, 1.5m]/[-20°, 20°]$	1.374	0.589	0.426	1.24	0.091	0.026	0.022	0.027
CalibDepth [30]	$[-1.5m, 1.5m]/[-20°, 20°]$	1.17	1.31	1.02	1.17	0.123	0.064	0.226	0.08
DEdgeNet [31]	$[-1.5m, 1.5m]/[-20°, 20°]$	1.109	0.258	0.330	**0.153**	0.159	0.052	0.014	**0.018**
HIFMNet(ours)	$[-1.5m, 1.5m]/[-20°, 20°]$	**0.636**	**0.13**	**0.239**	0.54	**0.0347**	**0.0081**	**0.0105**	0.0258

TABLE II: Ablation Study without Model Ensemble.

Method	Iteration number	Translation Error(cm)			Rotation Error(°)		
		X	Y	Z	Roll	Pitch	Yaw
LCCNet	1	11.85	5.169	7.613	0.170	0.676	0.594
HIFMNet(Flow)	5	33.06	24.13	45.56	1.03	1.00	1.24
HIFMNet	1	8.203	4.833	7.417	0.263	0.554	0.423
	2	6.230	3.408	4.962	0.177	0.389	0.283
	3	5.603	2.984	4.378	0.149	0.338	0.199
	4	5.367	**2.829**	**4.298**	**0.139**	0.324	**0.182**
	5	**5.359**	2.843	4.302	0.140	**0.323**	**0.182**

TABLE III: Comparsion of Model Parameters(M).

Method	Backbones	Extrinsic Initialization	Extrinsic Refinement	Overall
LCCNet	20.88	45.87	-	66.75
HIFMNet(ours)	21.36	2.05	2.16	25.57

backbone parameters are only slightly more than those in LCCNet due to the inclusion of the context encoder. The CNN blocks used for processing matched features and extrinsic regression constitute only 16.5% of all parameters. This proves that the excellent performance of our method is attributed to a superior feature matching strategy rather than an increase in network size.

Nevertheless, our method still has some limitations. As shown in Table IV, the extrinsic refinement is time-consuming since the construction of cost volumes and the lookup operation are computationally heavy. Further optimization is needed in this regard. We also evaluate HIFMNet pretrained on the KITTI-odometry dataset on the Nuscenes dataset. As shown in Table V, Without retraining on the NuScenes dataset, it performs poorly since it assumes a constant camera instrinsic. To achieve the transferability while ensuring robust regression, a possible approach is to predict calibration flows at each iteration and use a parameter-free extrinsic estimation module, such as ePnP to undistort them.

TABLE IV: Runtime Analysis(ms).

Method	Backbones	Extrinsic Initialization	Extrinsic Refinement	Overall
LCCNet	7.8	1.1	-	8.9
HIFMNet	7.8	0.6	16.1	24.5

TABLE V: The Performance of HIFMNet on Nuscenes.

Retrained on Nuscenes	Translation Error(cm)			Rotation Error(°)		
	X	Y	Z	Roll	Pitch	yaw
No	56.92	51.73	64.11	3.813	4.327	5.907
Yes	8.530	4.981	7.691	0.768	0.684	0.323

V. CONCLUSIONS

In this paper, we propose a novel learning-based LiDAR-Camera extrinsic calibration method, HIFMNet. During feature matching, to fully exploit the similarities between depth images and RGB images extracted by well-established CNN networks, we establish a globally-aware map-to-map cost volume and hierachical point-to-map cost volumes for iterative extrinsic regression within a single model. Experimental results demonstrate the superiority of our approach. In the future, we aim to enhance the efficiency and generalization of our network to meet practical requirements.

REFERENCES

[1] N. Schneider, F. Piewak, C. Stiller, and U. Franke, "RegNet: Multimodal sensor registration using deep neural networks," *2017 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1803–1810, June 2017.

[2] X. Lv, B. Wang, Z. Dou, D. Ye, and S. Wang, "LCCNet: LiDAR and Camera Self-Calibration using Cost Volume Network," *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 2888–2895, June 2021.

[3] X. Lv, S. Wang, and D. Ye, "CFNet: LiDAR-Camera Registration Using Calibration Flow Network," *Sensors*, vol. 21, no. 23, p. 8112, Dec. 2021.

[4] S. Mishra, G. Pandey, and S. Saripalli, "Extrinsic calibration of a 3D-LIDAR and a camera," *2020 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1765–1770, 2020.

[5] S. Mishra, P. R. Osteen, G. Pandey, and S. Saripalli, "Experimental evaluation of 3D-LIDAR camera extrinsic calibration," *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 9020–9026, 2020.

[6] J. Zhang, P. Siritanawan, Y. Yue, C. Yang, M. Wen, and D. Wang, "A Two-step Method for Extrinsic Calibration between a Sparse 3D LiDAR and a Thermal Camera," *2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pp. 1039–1044, Nov. 2018.

[7] G. Pandey, J. McBride, S. Savarese, and R. Eustice, "Extrinsic Calibration of a 3D Laser Scanner and an Omnidirectional Camera," *IFAC Proceedings Volumes*, vol. 43, no. 16, pp. 336–341, 2010.

[8] A.-I. García-Moreno, J.-J. Gonzalez-Barbosa, F.-J. Ornelas-Rodriguez, J. B. Hurtado-Ramos, and M.-N. Primo-Fuentes, "LIDAR and Panoramic Camera Extrinsic Calibration Approach Using a Pattern Plane," in *Pattern Recognition*, D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, J. A. Carrasco-Ochoa, J. F. Martínez-Trinidad, J. S. Rodríguez, and G. S. Di Baja, Eds., vol. 7914. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 104–113.

[9] Z. Pusztai and L. Hajder, "Accurate Calibration of LiDAR-Camera Systems Using Ordinary Boxes," *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pp. 394–402, Oct. 2017.

[10] G.-M. Lee, J.-H. Lee, and S.-Y. Park, "Calibration of VLP-16 Lidar and multi-view cameras using a ball for 360 degree 3D color map acquisition," *2017 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pp. 64–69, Nov. 2017.

[11] B. Fu, Y. Wang, X. Ding, Y. Jiao, L. Tang, and R. Xiong, "LiDAR-Camera Calibration Under Arbitrary Configurations: Observability and Methods," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 6, pp. 3089–3102, June 2020.

[12] G. Pandey, J. R. McBride, S. Savarese, and R. M. Eustice, "Automatic Extrinsic Calibration of Vision and Lidar by Maximizing Mutual Information: Automatic Extrinsic Calibration of Vision and Lidar," *Journal of Field Robotics*, vol. 32, no. 5, pp. 696–722, Aug. 2015.

[13] Z. Taylor, J. Nieto, and D. Johnson, "Automatic calibration of multi-modal sensor systems using a gradient orientation measure," *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1293–1300, Nov. 2013.

[14] P. Jiang, P. R. Osteen, and S. Saripalli, "SemCal: Semantic LiDAR-Camera calibration using neural mutual information estimator," *2021 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pp. 1–7, 2021.

[15] S. Mishra, A. Parchami, E. Corona, P. Chakravarty, A. Vora, D. Parikh, and G. Pandey, "Localization of a smart infrastructure fisheye camera in a prior map for autonomous vehicles," *2022 International Conference on Robotics and Automation (ICRA)*, pp. 5998–6004, 2021.

[16] J. Levinson and S. Thrun, "Automatic Online Calibration of Cameras and Lasers," in *Robotics: Science and Systems IX*. Robotics: Science and Systems Foundation, June 2013.

[17] J. Castorena, U. S. Kamilov, and P. T. Boufounos, "Autocalibration of lidar and optical cameras via edge alignment," *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2862–2866, Mar. 2016.

[18] J. Jiang, P. Xue, S. Chen, Z. Liu, X. Zhang, and N. Zheng, "Line Feature Based Extrinsic Calibration of LiDAR and Camera," *2018 IEEE International Conference on Vehicular Electronics and Safety (ICVES)*, pp. 1–6, Sept. 2018.

[19] T. Ma, Z. Liu, G. Yan, and Y. Li, "CRLF: Automatic Calibration and Refinement based on Line Feature for LiDAR and Camera in Road Scenes," *ArXiv*, Mar. 2021.

[20] R. Ishikawa, T. Oishi, and K. Ikeuchi, "LiDAR and Camera Calibration Using Motions Estimated by Sensor Fusion Odometry," *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7342–7349, Oct. 2018.

[21] G. Iyer, R. K. Ram, J. K. Murthy, and K. M. Krishna, "CalibNet: Geometrically Supervised Extrinsic Calibration using 3D Spatial Transformer Networks," *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1110–1117, Oct. 2018.

[22] K. Yuan, Z. Guo, and Z. J. Wang, "RGGNet: Tolerance Aware LiDAR-Camera Online Calibration With Geometric Deep Learning and Generative Model," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6956–6963, Oct. 2020.

[23] J. Shi, Z. Zhu, J. Zhang, R. Liu, Z. Wang, S. Chen, and H. Liu, "Calibrcnn: Calibrating camera and lidar by recurrent convolutional neural network and geometric constraints," *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 10 197–10 202, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:231913589

[24] Y. Sun, J. Li, Y. Wang, X. Xu, X. Yang, and Z. Sun, "ATOP: An Attention-to-Optimization Approach for Automatic LiDAR-Camera Calibration via Cross-Modal Object Matching," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 1, pp. 696–708, Jan. 2023.

[25] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, "PWC-Net : CNNs for Optical Flow Using Pyramid , Warping , and Cost," 2018.

[26] Z. Teed and J. Deng, "RAFT: Recurrent All-Pairs Field Transforms for Optical Flow," in *Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds., vol. 12347. Cham: Springer International Publishing, 2020, pp. 402–419.

[27] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2015. [Online]. Available: https://api.semanticscholar.org/CorpusID:206594692

[28] Teed, Zachary and Deng, Jia, "Raft-3d: Scene flow using rigid-motion embeddings," *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8371–8380, 2020.

[29] A. Zhu, Y. Xiao, C. Liu, and Z. Cao, "Robust lidar-camera alignment with modality adapted local-to-global representation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 33, pp. 59–73, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:251453063

[30] J. Zhu, J. Xue, and P. Zhang, "Calibdepth: Unifying depth map representation for iterative lidar-camera online calibration," *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 726–733, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:259338217

[31] Y. Hu, H. Ma, L. Jie, and H. Zhang, "Dedgenet: Extrinsic calibration of camera and lidar with depth-discontinuous edges," *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 11 439–11 445, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:259338884

[32] G. Iyer, R. K. Ram, J. K. Murthy, and K. M. Krishna, "CalibNet: Geometrically Supervised Extrinsic Calibration using 3D Spatial Transformer Networks," *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1110–1117, Oct. 2018.