Targetless Lidar-camera Calibration via Cross-modality Structure Consistency

Ni Ou [1]

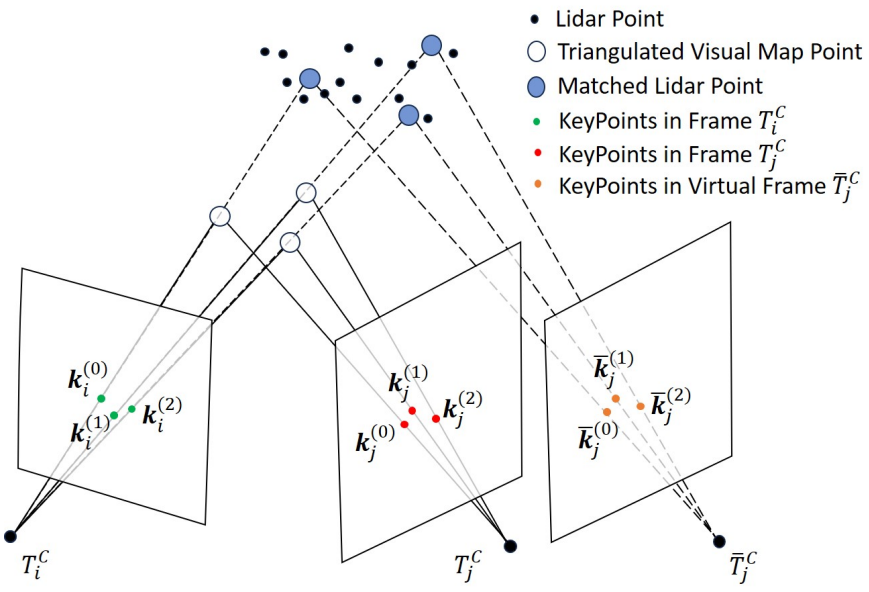
[1]Affiliation not available

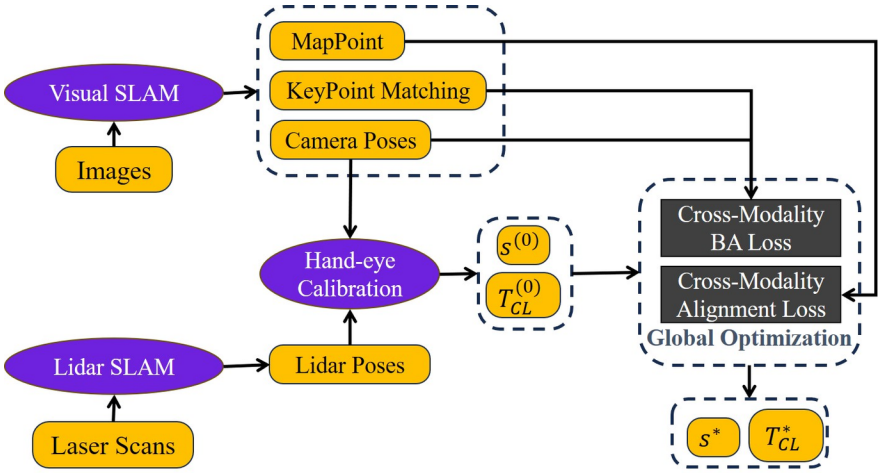
November 28, 2023

Abstract

Lidar and cameras serve as essential sensors for automated vehicles and intelligent robots, and they are frequently fused in complicated tasks. Precise extrinsic calibration is the prerequisite of Lidar-camera fusion. Hand-eye calibration is almost the most commonly used targetless calibration approach. This paper presents a particular degeneration problem of hand-eye calibration when sensor motions lack rotation. This context is common for ground vehicles, especially those traveling on urban roads, leading to a significant deterioration in translational calibration performance. To address this problem, we propose a novel motion-based Lidar-camera calibration framework based on cross-modality structure consistency. It is globally convergent within the specified search range and can achieve satisfactory translation calibration accuracy in degenerate scenarios. To verify the effectiveness of our framework, we compare its performance to one motion-based method and two appearance-based methods using six Lidar-camera data sequences from the KITTI dataset. Additionally, an ablation study is conducted to demonstrate the effectiveness of each module within our framework.

Our **codes** are now available on githubfor reproduction.





Targetless Lidar-camera Calibration via Cross-modality Structure Consistency

Ni Ou, Hanyu Cai, Junzheng Wang*

***Abstract*—Lidar and cameras serve as essential sensors for automated vehicles and intelligent robots, and they are frequently fused in complicated tasks. Precise extrinsic calibration is the prerequisite of Lidar-camera fusion. Hand-eye calibration is almost the most commonly used targetless calibration approach. This paper presents a particular degeneration problem of hand-eye calibration when sensor motions lack rotation. This context is common for ground vehicles, especially those traveling on urban roads, leading to a significant deterioration in translational calibration performance. To address this problem, we propose a novel targetless Lidar-camera calibration method based on cross-modality structure consistency. Our proposed method utilizes cross-modality structure consistency and ensures global convergence within a large search range. Moreover, it achieves highly accurate translation calibration even in challenging scenarios. Through extensive experimentation, we demonstrate that our approach outperforms three other state-of-the-art targetless calibration methods across various metrics. Furthermore, we conduct an ablation study to validate the effectiveness of each module within our framework.**

***Index Terms*—Calibration, Lidar, camera, automated vehicles**

I. Introduction

In the past decade, since Lidar and camera have complementary characteristics, Lidar-camera fusion has sparked increasing interest in the field of autonomous driving. Lidars are able to directly measure the distances of sparse points in the surrounding environment, while cameras can capture dense pictures with rich texture information. By fusing data from both sensors, intelligent vehicles are capable of effectively handling perception [1]–[3] and navigation tasks [4]–[6] in sophisticated environments.

The extrinsic calibration of the Lidar and camera sensors provides the relative transformation between the two, allowing for the processing of Lidar and camera data in a shared coordinate system. The establishment of cross-modality correspondences is a crucial step in Lidar-camera calibration. It often involves designing loss functions based on these correspondences for calibration purposes [7], [8]. Lidar-camera calibration algorithms can be broadly classified into two types based on whether a specific target is used for correspondence extraction: target-based and targetless.

Target-based calibration algorithms typically rely on man-made targets with known dimensions. The chessboard is the most common and widely studied target for Lidar-camera calibration [8]–[10]. With known grid size, its position and orientation can be easily estimated by a monocular vision sensor through corner detection algorithms [11], [12]. Meanwhile, when this target is scanned by an adequate number of Lidar beams, it can also be localized in the Lidar coordinate system through plane fitting [13]. Aside from chessboards, planar objects with regular features, such as tags [14], holes [15], or specifically shaped objects [16], are also feasible targets for calibration. Overall, target-based methods are useful when a calibration target with high machining accuracy is available, but they can be labor-intensive since the target needs to be moved to different places manually. Additionally, in certain cases, target-based methods may be less effective than targetless ones [17], [18].

On the opposite, targetless methods establish correspondences in natural scenes rather than relying on specific objects. These methods can be broadly categorized into two branches: appearance-based and motion-based. The former utilizes cross-modality correspondence by either designing artificial rules or learning them through self-supervision. For example, Lidar-camera edge association is a widely-studied appearance-based correspondence [19], [20], which is based on the comprehensive Euclidean distance between Lidar depth-discontinuous edges and camera intensity-discontinuous edges. Recent research has focused on extracting Lidar depth-continuous edges [17], [21] through voxel-level plane fitting, as this kind of Lidar edges are immune to the foreground inflation and blending points problems. Other branches of appearance-based algorithms rely on mutual information [22], [23] or self-supervised training [24]–[26]. In contrast, motion-based approaches rely on constraints across frames for calibration optimization. One practical technique for initial calibration estimation is hand-eye calibration (**HECalib**), which is based on pose-based constraints and has been widely studied [27], [28]. HECalib is globally convergent and only requires the relative poses (motions) of the Lidar and camera as inputs. There have been several improvements in HECalib. On the input side, its performance can be enhanced by refining Lidar and visual odometry [18], [29] and optimizing timing offsets [30], [31].

In addition to the aforementioned two classes of calibration algorithms, a growing number of hybrid methods have been developed to combine their respective strengths. For instance, certain proposals employ HECalib for an initial calibration and then utilize specific appearance-based metrics for further

This work is supported the National Natural Science Foundation of China under Grant 62173038

The authors are all with the State Key Laboratory of Intelligent Control and Decision of Complex Systems, Beijing Institute of Technology, Beijing, China. (Corresponding author: Junzheng wang, email: wangjz@bit.edu.cn)

improvement [31]–[33]. However, these methods tend to be less effective when HECalib is inadequate, particularly in degenerate situations involving limited rotational sensor motions [34]. Unfortunately, few methods have thoroughly resolved the degeneration problem in calibration, which becomes particularly severe when the vehicle travels in straight lines.

This paper presents a targetless hybrid method to effectively tackle the aforementioned problem. Our approach comprises a modified HECalib mechanism with regularization and a global optimization module. The former helps mitigate its degeneration, while the latter refines the initial calibration parameters across an extensive range of values. The global optimization is developed based on the principle of cross-modality structure consistency, which assumes that the 3D points detected by Lidar are also captured by the camera. Unlike other hybrid methods, our approach does not include any predefined appearance-based metrics or densification techniques. Instead, our method leverages the natural cross-frame geometry constraints and generalizes well to a wide array of scenarios. We validate the effectiveness of our approach across various degenerate scenarios, including an extreme case where the vehicle undergoes only unidirectional translational motions (Sequence 04, as indicated in Table I). Our main contributions are summarized below.

- We analyze the degeneration problem in HECalib and propose a regularization approach to suppress the translational calibration error.
- We propose a novel targetless Lidar-camera calibration method using cross-modality structure consistency. This method is globally convergent and immune to degeneration, and it can be applied to gray-scale cameras as well.
- We evaluate our method and compare it to a motion-based method [30] and two appearance-based methods [19], [24] using six sequences from the KITTI datasets (a total of 14,316 frames). We also conduct an ablation study to confirm the effectiveness of each module we designed. Additionally, we make our codes openly available on GitHub to benefit the research community.

II. Related Works

A. Sensor Motion Estimation

The input of HECalib are individual sensor motions of Lidar and camera, and we review relevant literature from two perspectives: Lidar odometry and visual odometry. Regarding the former, Iterative Closest Point (ICP) [35] has been widely applied to compute relative transformations between adjacent frames in many motion-based calibration methods [29], [31], [32]. It is closed-formed and efficient when the initial alignment is good enough. Moreover, our previous work [18] provides a unique solution to robust pose estimation for low-resolution Lidar with the assistance of camera data. Furthermore, Lidar SLAM (Simultaneous Localization and Mapping) has proven to be reliable for consecutive Lidar pose estimation [36]–[38]. This is due to its consideration of the motion model and the utilization of scan-map registration to reduce accumulated error. In the case of large-scale scenes, back-end optimization techniques are commonly employed to further enhance the performance of Lidar SLAM [39], [40].

Unlike Lidar, the estimation of monocular motions presents a challenge due to the scale ambiguity problem in translation. Although an initial scale factor can be determined by HECalib, maintaining a consistent scale throughout the entire trajectory remains difficult [41], [42]. Visual SLAM techniques [43]–[45] are able to address this problem by employing global and local bundle adjustment (BA) [46], especially when integrated with the use of loop closure detection [47]. In comparison to BA, Structure from Motion (SfM) is a more powerful technique for computing camera poses and recovering 3D mappoints [48], but it becomes computationally expensive as the number of frames increases, which limits its versatility. Aside from these systematic methods, an optical-based pipeline is proposed in [29] aiming to address the scale ambiguity, which jointly optimizes the extrinsic parameters and camera motions by tracking Lidar points.

B. Structure Consistency

The consistency of the Lidar-camera structure is closely tied to our method, and it has become increasingly popular in recent calibration studies. Some approaches leverage this prior knowledge by establishing cross-frame geometry constraints between the Lidar and camera. For instance, a self-supervision appearance-based method [49] incorporates a synthetic view constraint to validate the accuracy of the reprojected Lidar depth map. Additionally, Lidar poses can be directly utilized to create reprojection residuals for calibration within a visual SLAM framework [30]. A method similar to ours is presented in [50], where joint calibration is utilized to minimize the distance between scaled visual mappoints and Lidar points.

Another category of methods considers Lidar-camera calibration as an equivalent task of joint 3D reconstruction. In this case, the calibration error is regarded as the 3D distance between visual and Lidar maps. This theory can be simply applied to the calibration of stereo (or RGB-D) cameras and Lidar [51], and it can be extended to the calibration of monocular cameras and Lidar with the initial scale factor provided by HECalib [18]. Techniques such as vision densification [52] and semantic information [53] can also enhance the calibration performance in these scenarios. Moreover, similar to Lidar odometry [36], point-line and point-plane distances have also been employed to establish the connection between visual SfM points and Lidar points for calibration [54]. However, implementing visual 3D reconstruction [55], [56] in degenerate scenes is still challenging since this technique requires the camera to capture a fixed object from multiple angles.

III. Method

A. Overview

The whole framework of our method is present in Fig. 1. In the first stage, Visual and Lidar SLAM predict respect camera and Lidar pose from the sequence of raw data. Meanwhile, some intermediate data of the Visual SLAM are recorded, including the positions of triangulated keypoints and cross-frame keypoint-keypoint correspondences. Then, HECalib estimates

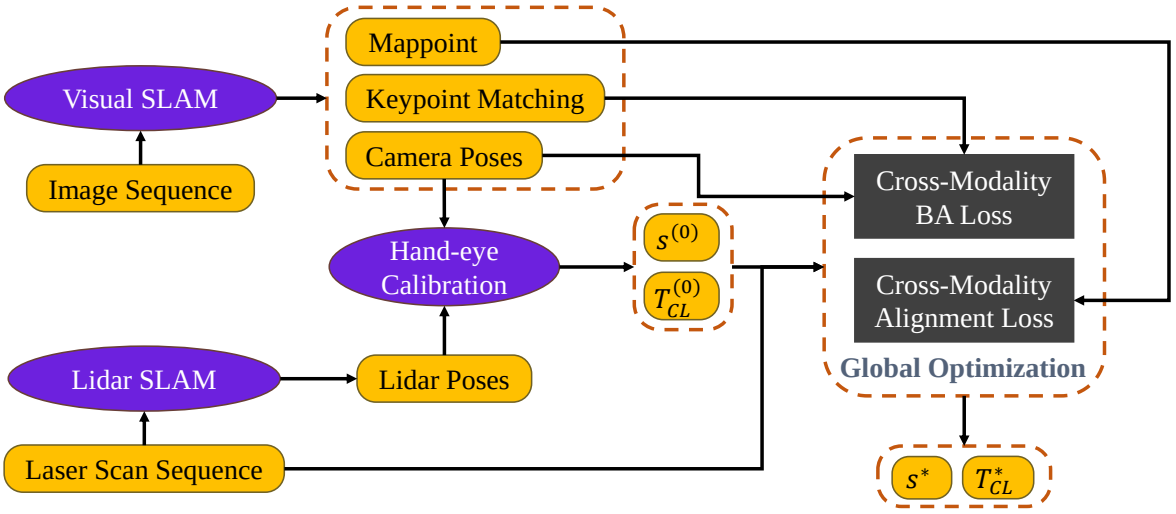


Fig. 1. Framework of our method. *Mappoint* indicates the triangulated keypoints in the visual map, and *Keypoint Matching* refers to the keypoint-keypoint correspondences for each pair of covisible keyframes.

the initial extrinsic matrix $\boldsymbol{T}_{CL}^{(0)}$ and initial monocular scale $s^{(0)}$ using the predicted sensor poses. Finally, a global optimization process is performed to find the best extrinsic matrix $\boldsymbol{T}_{CL}^{*}$ and monocular scale s^{*}. The two losses depicted in Fig 1, namely the Cross Modality BA Loss and the Cross-modality Alignment Loss, are designed using the principle of cross-modality structure consistency. These losses will be introduced in detail in Sections III-C and III-D.

The remaining parts of Section III are structured as follows. Firstly, Section III-B reviews the principles of HECalib and investigates the occurrence of degeneration caused by the absence of rotational sensor motions. In the same section, we apply a regularization term to modify the ordinary HECalib to suppress negative impacts made by degeneration. Then, the subsequent two sub-sections jointly introduce the main body of our method. Section III-C briefly reviews the structure of Visual SLAM and introduces the development of Cross-modality BA Loss, while Section III-D describes Cross-modality Alignment Loss and three constraints to form the entire global optimization module. Ultimately, for the purpose of reproduction, we provide implementation details in Section III-E.

B. HECalib with Regularization

Let F_i denote Frame i, and let $\boldsymbol{T}_{ij}^{C}, \boldsymbol{T}_{ij}^{L} \in \mathbb{R}^{4\text{x}4}$ represent the relative poses between F_i and F_j for camera and Lidar, respectively. As described in [27], the constraint for HECalib is formulated in (1), which can be expanded into (2) and (3).

$$\boldsymbol{T}_{ij}^{C}\boldsymbol{T}_{CL} = \boldsymbol{T}_{CL}\boldsymbol{T}_{ij}^{L} \tag{1}$$

$$\boldsymbol{R}_{ij}^{C}\boldsymbol{R}_{CL} = \boldsymbol{R}_{CL}\boldsymbol{R}_{ij}^{L} \tag{2}$$

$$(\boldsymbol{R}_{ij}^{C} - \boldsymbol{I})\boldsymbol{t}_{CL} + s \cdot \boldsymbol{t}_{ij}^{C} = \boldsymbol{R}_{CL}\boldsymbol{t}_{ij}^{L} \tag{3}$$

where $\boldsymbol{I} \in \mathbb{R}^{3\times3}$ is the identity matrix, $\boldsymbol{R}_{ij}^{C}, \boldsymbol{R}_{ij}^{L} \in \mathbb{R}^{3\text{x}3}$ are the corresponding rotation matrices of $\boldsymbol{T}_{ij}^{C}$ and $\boldsymbol{T}_{ij}^{L}$; $\boldsymbol{t}_{ij}^{C}, \boldsymbol{t}_{ij}^{L} \in \mathbb{R}^{3}$ are the respect translations of $\boldsymbol{T}_{ij}^{C}$ and $\boldsymbol{T}_{ij}^{L}$.

According to [34], $\boldsymbol{R}_{CL}$ can be solved individually from (2) using singular value decomposition (SVD). After substituting the obtained value of $\boldsymbol{R}_{CL}$ into (3), $\boldsymbol{t}_{CL}$ and s can be solved using the linear least square method.

Unfortunately, HECalib is prone to performance deterioration when rotational movements are insufficient to activate the calibration. Mathematically, when there is no rotation in the sensor motions, i.e., $\boldsymbol{R}_{ij}^{C} = \boldsymbol{R}_{ij}^{L} = \boldsymbol{I}$, Constraint (2) is no longer applicable. Moreover, the coefficient of $\boldsymbol{t}_{CL}$ in Constraint (3) becomes zero (matrix), indicating that the value of $\boldsymbol{t}_{CL}$ is no longer restricted by (19). We refer to this occurrence as **degeneration**. It is worth noting that degeneration always occurs in both sensors. Given that the Lidar and camera sensors are mounted on the same rigid body, if either $\boldsymbol{R}_{ij}^{C}$ or $\boldsymbol{R}_{ij}^{L}$ equals the identity, it implies that both of them must be identity. Although $\boldsymbol{R}_{ij}^{C}$ and $\boldsymbol{R}_{ij}^{L}$ can not be strictly equal to identity in practice, empirical observations indicate that a substantial number of sensor movements with minor rotation can lead to highly inaccurate solutions for $\boldsymbol{t}_{CL}$.

Based on the analysis above, the error in solving $\boldsymbol{t}_{CL}$ can primarily be attributed to a lack of appropriate constraints. Therefore, we develop a regularization term to impose loose constraints on $\boldsymbol{t}_{CL}$ while ensuring its adherence to the HECalib constraints. Extrinsic parameters are solved by the non-linear optimization on error items formulated in (4). The initial value of $\boldsymbol{R}_{CL}$ & s are solved through ordinary HECalib while the initial value of $\boldsymbol{t}_{CL}$ is replaced with a rough estimation $\boldsymbol{t}_{\mu}$. This regularization method is practical in real-world applications because $\boldsymbol{t}_{\mu}$ does not require a high level of precision—even errors within a few tens of centimeters are acceptable.

$$\xi_{ij} = \|(\boldsymbol{R}_{ij}^{C} - \boldsymbol{I})\boldsymbol{t}_{CL} + s \cdot \boldsymbol{t}_{ij}^{C} - \boldsymbol{R}_{CL}\boldsymbol{t}_{ij}^{L}\|_2^2 + w_r\|\boldsymbol{t}_{CL} - \boldsymbol{t}_{\mu}\|_2^2 \tag{4}$$

where w_r is a constant regularization weight for each ξ_{ij}.

This non-linear optimization problem can be easily implemented using specialized libraries such as g2o [57] or ceres-

solver [58]. We set $\boldsymbol{t}_\mu = \mathbf{0}$ for our experiments on the KITTI dataset. The effectiveness of translational regularization is demonstrated in the first two rows of each sequence group in Table I. With the proposed regularization, the calibration error in rotation remains almost unchanged, while the error in translation significantly decreases to a more reasonable magnitude.

C. Cross-modality Reprojection

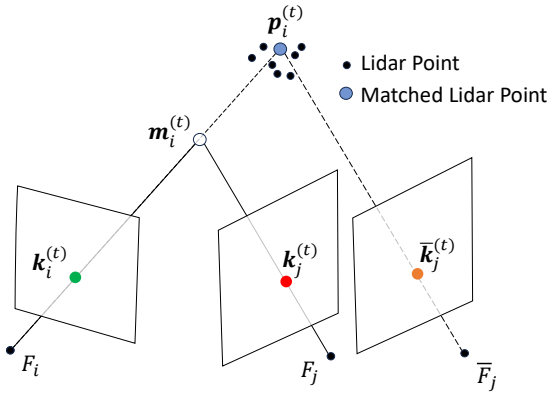


Fig. 2. Cross-modality Reprojection between Lidar and camera. F_i: Frame i, F_j: Frame j, $\overline{F}_j$: Virtual Frame j. Lidar points (black) shown in the figure have been transformed to the camera coordinate system of F_i (O_i^C), and that matched with the keypoint $\boldsymbol{k}_i^{(t)}$ is highlighted in blue.

In this section, we develop Cross-modality Reprojection error based on cross-modality structure consistency, aiming to further improve the accuracy of $\boldsymbol{t}_{CL}$. Different from other works reviewed in Section II-B, Cross-modality Reprojection can be applied to scenarios with degenerate sensor motions, as it does not rely on the camera being placed at different capturing angles. To facilitate comprehension of this theory, we will begin with a concise overview of Visual SLAM.

Taking Monocular SLAM as an example, the first stage involves extracting keypoints in each frame and employing keypoint matching between successive frames. Subsequently, two frames with sufficient parallax will be selected for initialization using matched keypoints. The initialization process includes the recovery of the relative pose through fundamental matrix computation and the construction of an initial visual map through triangulation. Following initialization, the visual map is tracked in each subsequent frame. Additionally, Bundle Adjustment [59] (BA) is implemented to simultaneously optimize camera poses and mappoint positions. If new triangulation operations are successful, the relevant mappoints will also be added. As a consequence, a mappoint can correspond to several keypoints in different frames (at most one keypoint in one frame), and keypoint-keypoint correspondences across frames are also established through the mappoint-keypoint connection. These steps can also be applied to stereo and RGB-D Visual SLAM. Detailed information on this subject can be found in [44], [45].

With the above preliminaries, we present the schematic diagram of Cross-modality Reprojection in Fig. 2, with corresponding notifications shown in its caption. This figure showcases a pair of matched keypoints $\boldsymbol{k}_i^{(t)}$ & $\boldsymbol{k}_j^{(t)}$ across F_i & F_j as well as their corresponding mappoint $\boldsymbol{m}_i^{(t)}$. For the convenience of the following description, we present the following **assumptions** for reference.

(1) The extrinsic matrix $\boldsymbol{T}_{CL}$ is completely accurate;
(2) The 3D point in real world corresponding to $\boldsymbol{m}_i^{(t)}$ is also scanned by Lidar, which is denoted as $\boldsymbol{q}_i^{(t)}$.

In fact, $\boldsymbol{p}_i^{(t)}$ in Fig. 2 is transformed from $\boldsymbol{q}_i^{(t)}$ using (5).

$$\boldsymbol{p}_i^{(t)} = \boldsymbol{T}_{CL}\boldsymbol{q}_i^{(t)} \tag{5}$$

First, we review the form of BA reprojection error [46]. The reprojection errors of $\boldsymbol{m}_i^{(t)}$ on F_i & F_j are formulated in (6) & (7), correspondingly. Note that ${}^{\mathbf{BA}}\boldsymbol{e}_i^{(t)}$ & ${}^{\mathbf{BA}}\boldsymbol{e}_j^{(t)}$ are vectors, and the same applies to the following error items.

$${}^{\mathbf{BA}}\boldsymbol{e}_i^{(t)} = \pi(\boldsymbol{m}_i^{(t)}) - \boldsymbol{k}_i^{(t)} \tag{6}$$

$${}^{\mathbf{BA}}\boldsymbol{e}_j^{(t)} = \pi(\boldsymbol{R}_{ij}^C\boldsymbol{m}_i^{(t)} + \boldsymbol{t}_{ij}^C) - \boldsymbol{k}_j^{(t)} \tag{7}$$

where $\pi(\cdot)$ stands for the projection function of the camera.

In contrast to ordinary BA, our reprojection is predicated on Lidar points instead of triangulated visual mappoints. Considering the scale equivalence of visual projection, our reprojection error in F_i can be expressed as (8). To extend this reprojection to F_j, we introduce the concept of virtual frames. The only difference between Frame F_j and Virtual Frame $\overline{F}_j$ lies in their poses. As the scale of $\boldsymbol{T}_{ij}^C$ is determined by visual initialization and $\boldsymbol{p}_i^{(t)}$ is of real size, we modulate the relative translation between F_i & $\overline{F}_j$ to $s\cdot\boldsymbol{t}_{ij}^C$, thereby maintaining the reprojection similarity relationship depicted in Fig. 2. With the concept of virtual frames, our reprojection error in $\overline{F}_j$ can be formulated as (9).

$${}^{\mathbf{CBA}}\boldsymbol{e}_i^{(t)} = \pi(\boldsymbol{p}_i^{(t)}) - \boldsymbol{k}_i^{(t)} \tag{8}$$

$${}^{\mathbf{CBA}}\boldsymbol{e}_j^{(t)} = \pi(\boldsymbol{R}_{ij}^C\boldsymbol{p}_i^{(t)} + s\cdot\boldsymbol{t}_{ij}^C) - \overline{\boldsymbol{k}}_j^{(t)} \tag{9}$$

This form of reprojection resembles the structure of Bundle Adjustment (BA) reprojection; hence, we term it as Cross-modality BA (CBA) reprojection. Intriguingly, the CBA reprojection would devolve to BA reprojection if Assumption (1) and (2) are both sustained. This property can be simply validated by substituting the ideal condition $\boldsymbol{p}_i^{(t)} = s\cdot\boldsymbol{m}_i^{(t)}$ into (8) & (9).

In fact, CBA errors (8) & (9) are both implicitly restricted by $\boldsymbol{T}_{CL}$ and can be applied to extrinsic parameters optimization. By substituting (5) into (8) and (9), we can explicitly display the error formulas with $\boldsymbol{T}_{CL}$ and s as independent variables, as shown in (10) and (11). For convenience, the superscript notation *CBA* of error items will be omitted henceforth.

$$\boldsymbol{e}_i^{(t)} = \pi(\boldsymbol{R}_{CL}\boldsymbol{q}_i^{(t)} + \boldsymbol{t}_{CL}) - \boldsymbol{k}_i^{(t)} \tag{10}$$

$$\boldsymbol{e}_j^{(t)} = \pi(\boldsymbol{R}_{ij}^C\boldsymbol{R}_{CL}\boldsymbol{q}_i^{(t)} + \boldsymbol{R}_{ij}^C\boldsymbol{t}_{CL} + s\cdot\boldsymbol{t}_{ij}^C) - \overline{\boldsymbol{k}}_j^{(t)} \tag{11}$$

Unfortunately, it is infeasible to directly utilize (10) and (11) for the optimization of $\boldsymbol{T}_{CL}$ and s because the acquisition of $\boldsymbol{q}_i^{(t)}$ (or $\boldsymbol{p}_i^{(t)}$) relies on the satisfaction of Assumptions (1) and (2). If Assumption (1) is not met, it will be impossible to

Algorithm 1: Cross-modality BA Loss

Input: $\boldsymbol{R}_{CL}$, $\boldsymbol{t}_{CL}$, s, δ_π, δ_1
Output: Loss L_1
$L_1 = 0$
$n = 0$
for $i = 1, 2, ..., N_f$ **do**
  $\boldsymbol{P}_i = \boldsymbol{R}_{CL}^{(0)} \boldsymbol{Q}_i + \boldsymbol{t}_{CL}^{(0)}$
  $\boldsymbol{U}_i = \pi(\boldsymbol{P}_i)$
  Build 2-D KD-Tree $\boldsymbol{\Gamma}_i$ on $\boldsymbol{U}_i$
  for $\boldsymbol{k}_i^{(t)}$ in keypoints of F_i **do**
    $a^* = \arg\min_{u_a \in \boldsymbol{U}_i} \|\boldsymbol{k}_i^{(t)} - u_a\|_2^2$ via $\boldsymbol{\Gamma}_i$
    if $\|\boldsymbol{k}_i^{(t)} - u_{a^*}\|_2^2 > \delta_\pi^2$ **then**
      continue
    end
    Select $\boldsymbol{p}_i^{(t)}$ from $\boldsymbol{P}_i$ using index a^*
    for F_j in covisible frames of F_i **do**
      if $\exists \boldsymbol{k}_j^{(t)}$ matched with $\boldsymbol{k}_i^{(t)}$ **then**
        Compute $\boldsymbol{e}_j^{(t)}$ using (11)
        if $\|\boldsymbol{e}_j^{(t)}\|_2^2 \leq \delta_1^2$ **then**
          $L_1 = L_1 + \|\boldsymbol{e}_j^{(t)}\|_2$
          $n = n + 1$
        end
      end
    end
  end
end
return L_1/n

select the correct $\boldsymbol{p}_i^{(t)}$ that corresponds to $\boldsymbol{k}_i^{(t)}$ since the entire Lidar point cloud is transformed using the erroneous $\boldsymbol{T}_{CL}$. Likewise, if Assumption (2) is not satisfied, it implies that a corresponding $\boldsymbol{p}^{(t)}$ does not exist in the Lidar point cloud. In other words, the Lidar sensor fails to capture the pertinent 3D structural point in the real-world context.

Furthermore, it is apparent that Assumption (1) will never be satisfied because $\boldsymbol{T}_{CL}$ & s (hereafter collectively denoted as $\boldsymbol{x}$) are variables to be optimized. To address this problem, we alternate between acquiring $\boldsymbol{p}_i^{(t)}$ and optimizing $\boldsymbol{x}$, rather than performing these actions simultaneously. This alternating optimization starts with $\boldsymbol{T}_{CL}^{(0)}$ and $s^{(0)}$ calculated by HECalib. Firstly, transform the Lidar point clouds into O_i^C using (5). Subsequently, $\boldsymbol{p}_i^{(t)}$ is identified by minimizing $\boldsymbol{e}_i^{(t)}$, whose index in the Lidar point cloud is determined via a KD-Tree-based nearest neighbor search within the projected Lidar points around $\boldsymbol{k}_i^{(t)}$. In the third step, the $\boldsymbol{x}$ is optimized by minimizing (11) with the selected $\boldsymbol{p}_i^{(t)}$. Finally, these steps are carried out iteratively, commencing each cycle from the first step.

In contrast, Assumption (2) might not hold for certain image keypoints, especially those that extend beyond the vertical scanning angle of the Lidar. Consequently, we devise an error-based criterion to determine if there is a corresponding $\boldsymbol{p}_i^{(t)}$ that matches keypoint $\boldsymbol{k}_i^{(t)}$. Specifically, we employ a KD-Tree search to locate the projected Lidar point that is nearest to $\boldsymbol{k}_i^{(t)}$. If the minimum distance exceeds a predefined threshold δ_π, we infer that there is no corresponding Lidar point for $\boldsymbol{k}_i^{(t)}$. Finally, we have also incorporated an error threshold δ_1 to simply filter out outliers. Denote $\boldsymbol{Q}_i$ as the i^{th} frame of Lidar point cloud; denote $\boldsymbol{P}_i = \boldsymbol{T}_{CL}\boldsymbol{Q}_i$; denote N_f as the number of frames. The complete algorithm to compute CBA loss is summarized in Algorithm 1. Please note that the values of $\boldsymbol{R}_{CL}, \boldsymbol{t}_{CL}, s$ are obtained from the current optimization iteration, and the same applies to Algorithm 2.

(a) before optimization (mean error: 5.91, #error: 108)

(b) after optimization (mean error: 2.80, #error: 126)

Fig. 3. Visual difference of CBA loss before (a) and after (b) global optimization. To improve visibility, we only sampled 50 keypoints using Farthest Point Sampling [60] for drawing. The number of error items is denoted by "#error". The solid and hollow small circles respectively denote the reprojection of the matched Lidar points onto $\overline{F}_j$ and their corresponding keypoints, which are the minuend and subtrahend on the right side of (9). A pair of solid and hollow circles of the same color indicates a match.

Finally, as depicted in Fig. 3, we showcase the visual disparity in CBA error (11) before and after optimization for one specific frame $\overline{F}_j$. In this figure, F_j is one of the covisible frames of F_i and matched Lidar points are selected by minimizing (8) in F_i and then reprojected to $\overline{F}_j$ using (9). Overall, these reprojected Lidar points (solid circles) are closer to the keypoints (hollow circles) in $\overline{F}_j$ after optimization. Two noticeable differences between Fig. 3(a) & 3(b) are annotated by red squares. Before optimization, the two hollow circles were positioned on the foreground objects, whereas their corresponding solid circles were located in the background. However, after optimization, both circles are correctly projected onto the foreground objects.

D. Global Optimization

Following the introduction of CBA loss, we describe the remaining components of the proposed global optimization in this section, including Cross-modality Alignment (CA) Loss and inequality constraints. We introduce CA loss as an auxiliary loss function to complement CBA loss and enhance the convergence performance of our optimization approach. The inspiration for the development of CA loss comes from the Iterative Closest Point (ICP) algorithm [35]. Our optimization process involves optimizing either the point-point or point-plane distance when the scaled visual mappoints $s \cdot \boldsymbol{m}_i^{(t)}$ are sufficiently close to the Lidar points $\boldsymbol{p}_i^{(t)}$, allowing for the refinement of the alignment.

Algorithm 2: Cross-modality Alignment Loss

Input: $\boldsymbol{R}_{CL}$, $\boldsymbol{t}_{CL}$, s, $\delta_2, \epsilon_p, r_{\min}$
Output: Loss L_2
$L_2 = 0$
$n = 0$
for $i = 1, 2, ..., N_f$ **do**
  $\boldsymbol{P}_i = \boldsymbol{R}_{CL}^{(0)} \boldsymbol{Q}_i + \boldsymbol{t}_{CL}^{(0)}$
  Build 3-D KD-Tree $\boldsymbol{\Gamma}_i$ on $\boldsymbol{P}_i$
  for $\boldsymbol{k}_i^{(t)}$ in keypoints of F_i **do**
    Retrieve mappoint $\boldsymbol{m}_i^{(t)}$ matched with $\boldsymbol{k}_i^{(t)}$
    Compute d_{pt} and select $\boldsymbol{p}_i^{(t)}$ using (12) with $\boldsymbol{\Gamma}_i$
    if $d_{pt} > \delta_2$ **then**
      continue;
    end
    Find neighboring points around $\boldsymbol{p}_i^{(t)}$ via $\boldsymbol{\Gamma}_i$
    Compute the normalized normal of $\boldsymbol{p}_i^{(t)}$: $\boldsymbol{n}_i^{(t)}$
    if (14) & (15) are satisfied under $\epsilon_p, r_{\min}$ **then**
      Compute d_{pl} using (13)
      $L_2 = L_2 + d_{pl}$
    else
      $L_2 = L_2 + d_{pt}$
    end
    $n = n + 1$
  end
end
return L_2/n

The detailed steps for computing CA loss are introduced as follows. Firstly, we retrieve the visual mappoint $\boldsymbol{m}_i^{(t)}$ corresponding to $\boldsymbol{k}_i^{(t)}$ from the established mappoint-keypoint correspondences built in Visual SLAM. Next, a KD-Tree is utilized to select the Lidar point $\boldsymbol{p}_i^{(t)}$ that is closest to $s \cdot \boldsymbol{m}_i^{(t)}$. Subsequently, the same KD-Tree is used to identify neighboring Lidar points of $\boldsymbol{p}_i^{(t)}$ and compute the normalized point normal [61] of $\boldsymbol{p}_i^{(t)}$. Then, the $\boldsymbol{p}_i^{(t)}$ and its minimal distance to $s \cdot \boldsymbol{m}_i^{(t)}$ are obtained using (12). If the Lidar points surrounding $\boldsymbol{p}_i^{(t)}$ can fit a plane, we replace point-point distance with point-plane distance as formulated in (13). Ultimately, similar to CBA, an error-based threshold δ_2 is also employed for outlier rejection.

$$d_{pt} = \min_{\boldsymbol{p}_i^{(t)} \in \boldsymbol{P}_i} \| s \cdot \boldsymbol{m}_i^{(t)} - \boldsymbol{p}_i^{(t)} \|_2 \tag{12}$$

$$d_{pl} = \left| \langle s \cdot \boldsymbol{m}_i^{(t)} - \boldsymbol{p}_i^{(t)}, \boldsymbol{n}_i^{(t)} \rangle \right| \tag{13}$$

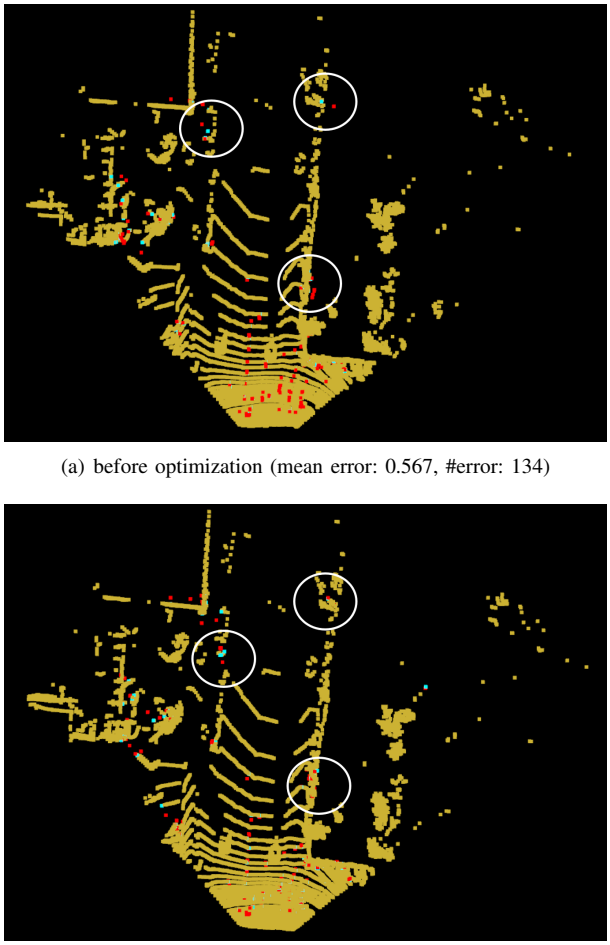
where $\langle \cdot, \cdot \rangle$ denotes the inner product operation between two vectors and $\| \cdot \|_2$ denotes the second norm of the vector.

Moreover, we devise two criteria (14) and (15) to assess the validity of the plane fitted by the Lidar points surrounding $\boldsymbol{p}_i^{(t)}$. (14) constraints the regression error of the plane while (15) ensure the fitted plane has sufficient size. Both criteria must be satisfied for the fitted plane to be considered valid. All the above procedures for computing CA loss are outlined in Algorithm 2. Furthermore, it is unnecessary to apply δ_2 to verify d_{pl} because d_{pl} is always less than d_{pt}.

$$\sum_j \left| \langle \boldsymbol{p}_i^{(j)} - \boldsymbol{p}_i^{(t)}, \boldsymbol{n}_i^{(t)} \rangle \right| < N_i^{(t)} \cdot \epsilon_p \tag{14}$$

$$\max_j \| \boldsymbol{p}_i^{(j)} - \boldsymbol{p}_i^{(t)} \|_2 > r_{\min} \tag{15}$$

where $N_i^{(t)}$ represents the number of $\boldsymbol{p}_i^{(j)}$ in the neighborhood of $\boldsymbol{p}_i^{(t)}$; $\boldsymbol{n}_i^{(t)}$ denotes the normalized point normal of $\boldsymbol{p}_i^{(t)}$; $\epsilon_p, r_{\min}$ are preset thresholds.



(a) before optimization (mean error: 0.567, #error: 134)

(b) after optimization (mean error: 0.353, #error: 162)

Fig. 4. Visual difference of CA loss before and after global optimization (bird-eye view). Yellow points denote the Lidar points $\boldsymbol{P}_i$ in Algorithm 2; red points denote the scaled visual mappoints $s \cdot \boldsymbol{m}_i^{(t)}$; light blue points mark the nearest yellow point matched by the red point.

Similar to Fig. 3, the visual difference of CA error before and after optimization for a specific example frame F_1 is also presented in Fig. 4. The visual mappoints observed by F_1 have been scaled by s, and the Lidar points of F_1 have been transformed into the camera coordinate system of F_1 using $\boldsymbol{T}_{CL}$. In comparison to Fig. 4(a), the transformed Lidar points (yellow) in Fig. 4(b) exhibit better alignment with the scaled visual mappoints (red). Three pairs of white circles highlight noticeable visual differences between the two figures.

Finally, as indicated in (16), we compute the aggregate loss by combining the L_1 and L_2 losses using their respective

weights w_1 and w_2. This aggregate loss is the actual objective function to be optimized.

$$L = w_1 \cdot L_1 + w_2 \cdot L_2 \tag{16}$$

Regarding the constraints of global optimization, we first utilize the Lie algebra formulation to remove the implicit constraints in $\boldsymbol{T}_{CL}$. Specifically, we formulate the 7-dimensional optimization variable $\boldsymbol{x}$ (used to denote $\boldsymbol{T}_{CL}$ & s before) in (17), where $\omega, \rho \in \mathbb{R}^3$ are the rotation and translation vectors of Lie algebra while the scalar s denotes the scale factor. With this representation, the initial value of $\boldsymbol{x}$ can be derived from $\boldsymbol{T}_{CL}^{(0)}$ and $s^{(0)}$. We define the bilateral bound constraint in (18) to determine the search range of $\boldsymbol{x}$, where $\boldsymbol{x}_0$ denotes the initial value of $\boldsymbol{x}$ and Δ_x is a 7-dimensional vector that represents the search radius in each degree.

$$\boldsymbol{x} = [\omega, \rho, s] \tag{17}$$

$$-\Delta_x + \boldsymbol{x}_0 \leq \boldsymbol{x} \leq \Delta_x + \boldsymbol{x}_0 \tag{18}$$

Furthermore, an inequality constraint (19) is imposed to ensure that the HECalib constraint (1) is satisfied within a specified error range. Despite the potential degeneration in hand-eye calibration, it can serve as a necessary but insufficient condition for verifying the correctness of the obtained solutions.

$$\sum_{i,j} \|\mathbf{Log}(\boldsymbol{T}_{CL}\boldsymbol{T}_{ij}^{L}) - \mathbf{Log}(\boldsymbol{T}_{ij}^{C}\boldsymbol{T}_{CL})\|_2^2 \leq N_f \cdot \delta_h \tag{19}$$

where $\mathbf{Log}(\cdot)$ transfers Lie groups to Lie algebras; N_f represents the number of Constraint (19); δ_h is a preset threshold.

The final constraint is imposed to limit the minimum inlier ratio of the CBA loss. When the Lidar and camera are accurately calibrated, the inlier ratio of the CBA reprojection errors should be significantly high since these errors are linked to correct correspondences. By applying Constraint (20), we can effectively exclude extreme cases where both the inlier ratio and the value of L_1 are low.

$$n_1^v / n_1 \geq \delta_v \tag{20}$$

where n_1^v denotes the number of valid $\boldsymbol{e}_j^{(t)}$ that satisfy $\boldsymbol{e}_j^{(t)} \leq \delta_1$; n_1 denotes the total number of $\boldsymbol{e}_j^{(t)}$; δ_v is a preset threshold.

Thus far, we have presented all the components of our proposed global optimization. Different from [21], [24], [26], [53], our method does not involve any appearance-based metrics in calibration or self-supervision training. Instead, we utilize cross-modality structure consistency to design adaptable and unsupervised loss functions. This strategic approach enhances generalization ability and eliminates human-labeled data and training requirements.

E. Implementation Details

Due to the existence of KD-Trees in Algorithm 1 and 2, neither L_1 nor L_2 is differentiable. As a consequence, we opt to employ a derivative-free optimization algorithm called Mesh Adaptive Direct Search (MADS) [62], [63] for the optimization process. MADS effectively explores the variable space and accommodates both bound and nonlinear constraints. In terms of the primary parameter settings of the MADS algorithm, the minimum mesh size is set to 10^{-6} for each degree, and the maximum number of black box evaluations (BBE) is set to 5000. Furthermore, the Variable Neighborhood Search (VNS) option, as proposed by reference [64], is enabled to suppress the occurrence of local optima. Ultimately, the weighted sum w_1 and w_2 given in (16) are both set to 1 in our experiments.

For our Visual SLAM and Lidar SLAM tools, we select ORB-SLAM [44] and F-LOAM [37], respectively. ORB-SLAM is a state-of-the-art monocular SLAM method that incorporates loop closure detection [65] and global bundle adjustment. Considering that F-LOAM lacks similar loop closure capabilities, we integrate Scan Context [40] into F-LOAM to enable loop closure detection and back-end optimization.

IV. Experiments and Results

A. Dataset Introduction

The KITTI dataset [66] is a widely used benchmark for evaluating computer vision and autonomous driving algorithms. It comprises 22 sequences of data captured by multiple sensors, including Lidar and camera. This dataset also includes the ground-truth Lidar-camera calibration extrinsic matrix. In our experiments, six KITTI sequences (00, 02, 03, 04, 05, 07) are selected for calibration evaluation. The majority of sensor motions in these sequences predominantly consist of straight-line movements, with only a small portion involving slight rotations due to curved road segments. Sequence 04 is the most extreme case, where the vehicle exclusively travels along a straight highway without any perceptible turns.

The sensors to be calibrated consist of a Velodyne HDL-64E Lidar and a gray-scale PointGray camera (camera-0). To provide clarity, we present a schematic diagram illustrating the relative transformation between the Lidar (O_L) and the camera (O_C) in Fig. 5. In this diagram, Axis X_L & Z_C are parallel to the forward direction of the vehicle while Axis Z_L & Y_C are perpendicular to the ground. For sequence 04, the sensor motions pertain solely to translation along Axis X_L (Z_C). For other sequences, considering the presence of vehicle turns, there are additional translational movements along Axis Y_L (X_C) and rotational motions around Axis Z_L (Y_C).

The following content of this section is organized below. First, we evaluate the calibration performance of our method and three baseline methods [19], [24], [30] on six KITTI sequences in Section IV-B. We also conduct an ablation study to verify the effectiveness of each component in our framework. Next, in Section IV-C, we present detailed information about the optimization process and analyze the properties of the proposed loss functions. Finally, we evaluate the degeneration-resistance performance of our method through an unidimensional analysis in Section IV-D. The experiments in Section IV-C and Section IV-D focus on sequence 04, which is the most representative degenerate KITTI sequence.

B. Calibration Accuracy

First, we compare the calibration accuracy of our method to that of other baselines. Eight metrics are defined for evaluation

TABLE I
CALIBRATION ERROR ON DIFFERENT KITTI SEQUENCES

Sequence	Method	Roll (°)	Pitch (°)	Yaw (°)	X (cm)	Y (cm)	Z (cm)	RRMSE (°)	TRMSE (cm)
00 (4541 | with)[1]	HECalib[2]	3.34	-0.58	-0.03	-281.2	-1386	-138.3	3.39	1421
	HECalib + Reg[3]	1.88	-2.83	1.72	31.47	-4.07	-7.15	3.81	32.53
	Levinson [19]	-0.03	-0.32	0.10	31.49	4.28	-7.26	0.33	32.60
	CalibNet [24]	2.71	-0.36	1.66	27.14	1.09	5.71	3.20	27.75
	Park [30]	1.78	-1.77	0.91	40.37	-62.99	-256.24	2.67	266.94
	CBA[4]	-0.02	-0.21	-0.09	3.41	-0.62	0.09	**0.23**	3.46
	CBA + CA (PT)[5]	-0.07	-0.21	-0.12	1.54	2.35	-0.46	0.25	**2.85**
	CBA + CA (PT+PL)[6]	-0.12	-0.22	-0.07	2.93	1.59	0.18	0.26	3.33
02 (4661 | with)	HECalib	8.25	-2.39	-1.56	-8919	5526	-1707	8.73	10630
	HECalib + Reg	8.25	-2.39	-1.56	29.18	-1.14	-5.62	8.73	29.74
	Levinson	-0.23	0.22	-0.17	23.69	0.30	8.11	0.36	25.04
	CalibNet	1.15	-0.92	-1.27	19.27	0.29	5.65	1.94	20.08
	Park	0.22	0.22	0.14	-29.80	91.91	2.85	0.34	96.66
	CBA	0.00	-0.38	-0.21	-2.67	-1.45	-2.15	0.43	3.72
	CBA + CA (PT)	-0.09	-0.39	-0.19	3.62	0.87	-1.93	0.44	4.19
	CBA + CA (PT+PL)	-0.01	-0.25	-0.16	-2.80	1.63	1.18	**0.29**	**3.45**
03 (801 | w/o)	HECalib	2.11	-0.48	-0.23	-903.9	-253.0	-543.2	2.18	1085
	HECalib + Reg	-2.36	-1.57	-2.67	26.47	-0.95	-8.54	3.89	27.83
	Levinson	-0.74	-0.67	0.05	26.46	0.91	8.57	1.00	27.82
	CalibNet	-4.56	0.17	0.20	13.85	13.06	3.99	4.57	19.45
	Park	0.07	0.08	-0.01	-7.62	13.10	66.05	**0.11**	67.76
	CBA	-0.05	-0.11	-0.11	-5.52	1.87	0.48	0.16	5.85
	CBA + CA (PT)	-0.08	-0.07	-0.13	1.75	6.85	0.44	0.17	7.08
	CBA + CA (PT+PL)	-0.08	-0.10	-0.09	-3.25	3.23	1.22	0.15	**4.74**
04 (271 | w/o)	HECalib	1.81	0.55	1.50	1658	336.5	179.6	2.41	1701
	HECalib + Reg	1.81	0.55	1.49	33.35	0.48	-7.56	2.42	34.20
	Levinson	-0.25	-1.05	-0.54	33.36	-0.48	-7.54	1.21	34.21
	CalibNet	1.82	-2.55	0.85	23.40	7.33	18.25	3.25	30.56
	Park	-0.06	0.07	-0.10	-106.8	44.61	37.16	0.14	121.5
	CBA	-0.05	-0.10	0.22	2.29	2.68	-1.10	0.25	3.68
	CBA + CA (PT)	-0.03	-0.08	-0.03	0.61	0.12	2.66	**0.09**	2.73
	CBA + CA (PT+PL)	0.01	-0.09	0.05	1.03	-0.19	1.10	0.11	**1.52**
05 (2761 | with)	HECalib	5.70	-0.78	0.06	-1369	-368.5	-360.8	5.76	1463
	HECalib + Reg	1.73	-2.99	-2.08	26.39	-4.88	-11.41	4.03	29.16
	Levinson	-0.06	-0.48	-0.52	30.06	-4.23	-13.75	0.71	33.32
	CalibNet	2.45	-1.84	0.85	21.54	8.95	13.08	3.18	26.74
	Park	1.22	-1.62	-2.00	91.63	24.61	-56.9	2.85	576.6
	CBA	-0.10	-0.29	-0.05	2.55	3.60	-1.23	0.31	4.58
	CBA + CA (PT)	0.03	-0.16	0.06	2.66	0.56	-0.73	**0.18**	2.81
	CBA + CA (PT+PL)	0.01	-0.19	-0.15	1.95	1.56	-0.54	0.24	**2.56**
07 (1101 | with)	HECalib	-4.91	0.20	-0.21	-773.4	-1010	1379	4.92	1876
	HECalib + Reg	1.60	-3.03	-2.08	30.55	-4.21	-13.34	4.01	33.60
	Levinson	-0.06	-0.48	-0.52	30.06	-4.23	-13.75	0.71	33.32
	CalibNet	2.45	-1.84	0.86	21.54	8.95	13.08	3.18	26.74
	Park	2.37	-7.60	-0.19	42.13	-76.06	16.75	2.37	88.55
	CBA	0.07	-0.66	-0.35	1.09	0.11	-10.89	0.75	10.95
	CBA + CA (PT)	0.08	-1.69	-2.41	3.76	-1.69	-2.41	0.30	4.78
	CBA + CA (PT+PL)	-0.05	-0.04	-0.20	1.07	2.07	0.14	**0.21**	**2.34**

[1] Number of Frames | with/without (w/o) loop closure
[2] Ordinary HECalib without translational regularization
[3] HECalib with translational regularization
[4] Calibration optimization only with Cross-modality BA Loss
[5] Calibration optimization with Cross-modality BA Loss and point-point Cross-modality Alignment Loss
[6] Calibration optimization with Cross-modality BA Loss and normal Cross-modality Alignment Loss (point-point + point-plane)

based on the definition of $\boldsymbol{T}_e$ formulated in (21). These metrics includes three Euler angles ($\Delta_{roll}, \Delta_{pitch}, \Delta_{yaw}$) of the rotation of $\boldsymbol{T}_e$, three translation components ($\Delta_X, \Delta_Y, \Delta_Z$) of $\boldsymbol{T}_e$. Additionally, we consider two other metrics, namely RRMSE and TRMSE, which are defined in (22) and (23), respectively. RRMSE and TRMSE reflect the comprehensive translational and rotational calibration error of methods.

$$\boldsymbol{T}_e = \left(\boldsymbol{T}_{CL}^{gt}\right)^{-1} \boldsymbol{T}_{CL}^{p} \tag{21}$$

where $\boldsymbol{T}_{CL}^{gt}$ is the ground-truth extrinsic matrix provided by KITTI dataset and $\boldsymbol{T}_{CL}^{p}$ is the predicted extrinsic matrix.

$$\textbf{RRMSE} = \sqrt{\Delta_{roll}^2 + \Delta_{pitch}^2 + \Delta_{yaw}^2} \tag{22}$$

$$\textbf{TRMSE} = \sqrt{\Delta_X^2 + \Delta_Y^2 + \Delta_Z^2} \tag{23}$$

As mentioned in Section I, two appearance-based methods [19], [30] are incorporated in our baselines. Since these

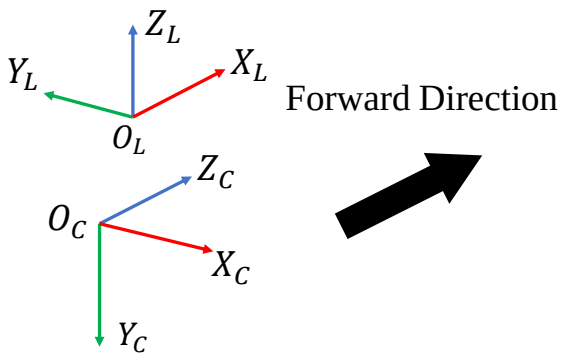


Fig. 5. The coordinate system relationship between Lidar and camera in the KITTI dataset. O_C and O_L are the coordinate origins of the camera and LiDAR, respectively. Each axis of camera and Lidar coordinate systems is annotated in the figure.

methods work for single-frame mode, we compare their best single-frame result to the multi-frame result of our method. For each of them, we evaluate its calibration results frame by frame on the dataset, and only the best single-frame result with minimum RRMSE+TRMSE is presented in Table I for comparison.

For CalibNet [24], we substitute the camera-0 images with camera-2 images as its input, since it is developed for calibrating Lidar and RGB camera. For the sake of consistency with the other groups of experiments, we assume that the transformation from camera-0 to camera-2 is already known prior to calibration. Then, we obtain the initial ($\boldsymbol{T}_{CL}^{(0)}$) and output ($\boldsymbol{T}_{CL}^{(1)}$) extrinsic matrices for CalibNet using (24) and (25), respectively. Furthermore, for self-supervision of CalibNet, sequence 11 to 17 are used for training and sequence 18, 20, 21 are used for validation.

$$\boldsymbol{T}_{CL}^{(0)} = \boldsymbol{T}_{C2,C0}\boldsymbol{T}_{C0,L}^{(0)} \tag{24}$$

$$\boldsymbol{T}_{CL}^{(1)} = \boldsymbol{T}_{C0,C2}^{-1}\boldsymbol{T}_{C2,L}^{(1)} \tag{25}$$

where $\boldsymbol{T}_{C0,C2}$ denotes the transformation from camera-0 to camera-2; $\boldsymbol{T}_{C0,L}^{(0)}$ denotes the initial transformation from Lidar to camera-0 estimated by HECalib; $\boldsymbol{T}_{C2,L}^{(1)}$ denotes the transformation from Lidar to camera-2 predicted by CalibNet.

Quantitative results are present in Table I. The initial calibration for each method (below the 2nd row) is provided by our modified HECalib introduced in Section III-B. It is illustrated that our method (the last row) significantly outperforms the baseline methods in terms of both RRMSE and TRMSE. In contrast, appearance-based approaches (3rd & 4th rows) do not effectively reduce the calibration error compared to the initial estimation (2nd row). One possible reason for this is that these methods can only converge from a small bias toward the ground-truth point. However, the initial calibration obtained by HECalib is not accurate enough for them. In comparison, the motion-based method (5th row) is globally convergent. However, it directly substitutes the HECalib constraint (1) into (7) to construct its reprojection error, resulting in a similar degeneration issue as HECalib. Consequently, its rotational errors are low while its translational errors are high, aligning with our previous analysis that the degeneration solely causes significant errors in $\boldsymbol{t}_{CL}$.

Subsequently, we also conduct ablation experiments by creating two variants of the proposed framework. The first variant eliminates the use of CA loss, while the second retains CA loss but exclusively applies point-point distance (12) for optimization. As illustrated in the final three rows of Table I, our method and both variants yield similar rotational errors. However, notable differences arise in translation. The last and the third-to-last rows of Table I jointly showcase that significant refinement of the TRMSE metric is achieved by adding CA loss. The complete form of our method (the last row) achieves the lowest TRMSE on almost all the sequences except for sequence 00.

Nevertheless, the second variant of our method, does not always perform better than the first. Its TRMSE metric is not as good as the first variant in sequences 02 and 03, illustrating the importance of the proposed point-plane distance. Regarding computational efficiency, the first variant runs faster than the second due to the absence of CA loss, and the second is more efficient than our complete version because Criterion (14) and (15) are no longer required to be validated.

Finally, we choose examples from four sequences (00, 03, 04, 07) to qualitatively demonstrate the calibration accuracy of our method. In Fig. 6, Lidar points are projected onto images using the predicted extrinsic matrix (left column) and the ground-truth (right column) one, respectively. Only slight visual differences can be recognized between two columns of corresponding images, which are marked with yellow circles.

C. Optimization Process

In this section, we analyze the trend of the final loss function L in the MADS algorithm. We plot the curve of L using the feasible points generated by MADS in Fig. 7. Feasible points refer to solutions that satisfying Constraint (19) & (20). Based on our observations, all the infeasible points depicted in Fig. 7 satisfy (20) but not (19). Therefore, all subsequent analyses concerning the infeasible points focus on (20).

Initially, L decreases rapidly, but its rate of decrease gradually slows down as the optimization process continues. The variation of L follows a step-like pattern, with noticeable drops at indices 1022, 2624, and 3603. Even towards the end of the optimization, L still shows a decreasing trend and has not fully reached convergence (although the optimization terminates due to the minimum mesh condition being met). This reflects the difficulty of optimizing L, which is mainly attributed to its discontinuity.

As expressed in (16), the final loss function L is a weighted sum of L_1 and L_2. Both L_1 and L_2 exhibit discontinuity due to variations in correspondences. The cause of the discontinuity in the CA loss is similar to that in the ICP algorithm [35]. CA correspondences can vanish or change while the relevant Euclidean distances exceed a predefined threshold δ_2. Similar patterns are observed in CA correspondences, which is an explanation to why the number of correspondences changes after optimization. The variation in CA correspondences is visually depicted by the white circles in Fig. 4 while that in CBA correspondences is marked by orange circles in Fig. 3.

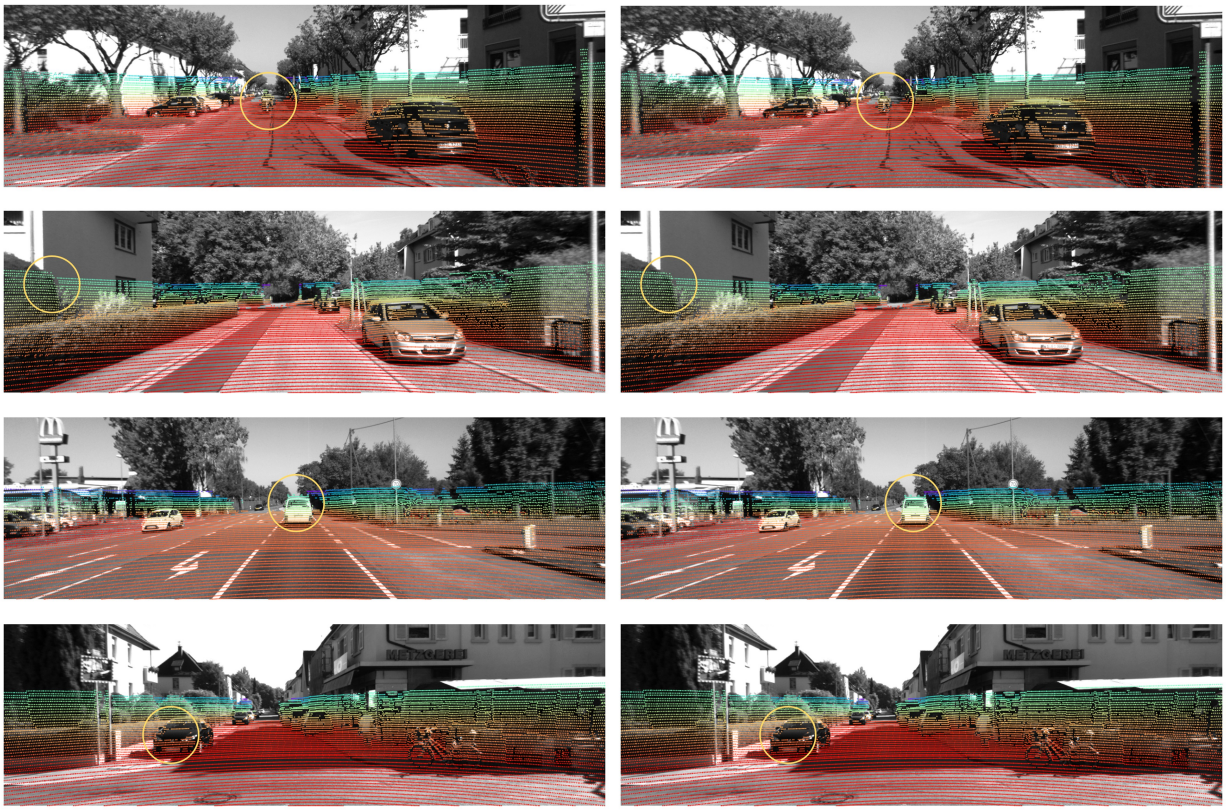


Fig. 6. **Left** column: Lidar points are projected onto images using the predicted extrinsic matrix; **Right** column: Lidar points are projected onto images using the ground-truth extrinsic matrix. From **top** to **bottom**: samples from KITTI seq 00, 03, 04, 07. Some perceptible differences are denoted with paired yellow circles.

Comparatively, the discontinuity in the CBA loss primarily stems from the reprojection operation. By minimizing (8), the position of $\boldsymbol{p}_i^{(t)}$ can undergo a shift during selection, leading to a sudden value change in the reprojection error calculated in (9). Discontinuity also happens when some $\boldsymbol{p}_i^{(t)}$ selected by (8) can not be tracked in the next iteration. Moreover, similar to CA, a subset of CBA correspondences may change or disappear when filtered by the threshold δ_1. To illustrate, we mark three changed CBA correspondences using a pair of orange circles in Fig. 3(a) and Fig. 3(b).

D. Unidimensional Analysis.

Taking the sequence 04 as an example, we show the degeneration that occurs in HECalib and the degeneration-resistance performance of our method in this section. The values of L and meeting statuses of (19) under unidimensional offsets of the optimization variable $\boldsymbol{x}$ are drawn in Fig. 8. As discussed in Section III-B, the solution of HECalib for $\boldsymbol{t}_{CL}$ becomes degenerate in this case. As shown in Fig. 8(d), 8(e) and 8(f), no infeasible points appear as the translational offset increases. According to the theory of Lie algebra, the physical translation $\boldsymbol{t}$ equals the translation vector ρ when rotation degrades to zero. Therefore, this observation implies that the meeting status of (19) does not alter with the variation of $\boldsymbol{t}_{CL}$, which aligns with our theoretical derivations in Section III-B.

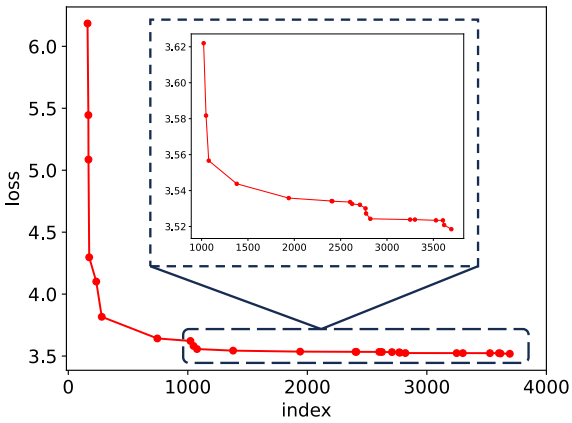
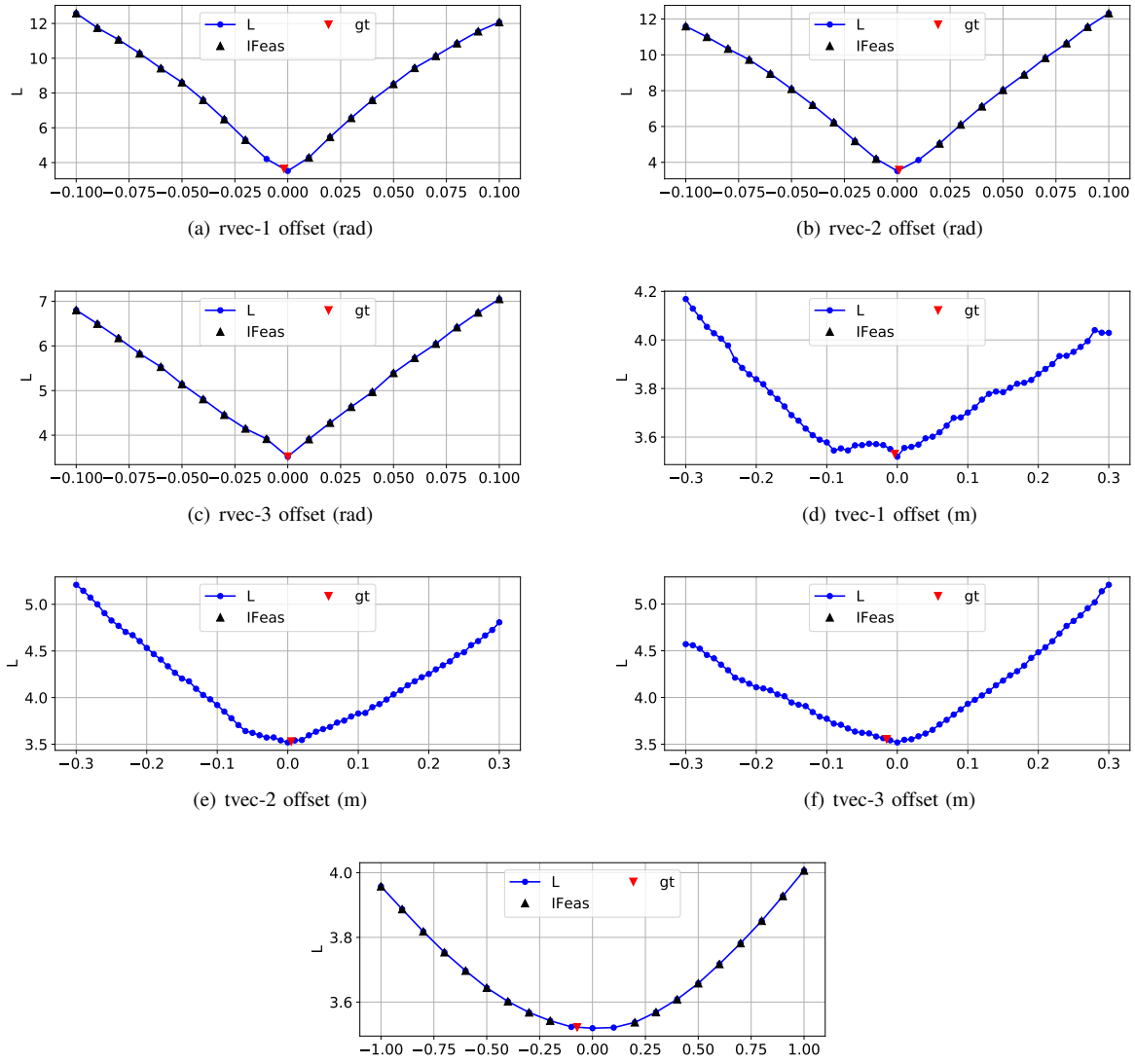


Fig. 7. Curve of the loss (16) during optimization. Part of the curve (index $\geq$ 1022) has been zoomed for better viewing. The label of the x-axis indicates the index of feasible points in the MADS algorithm.

However, our designed loss function is sensitive to translation changes in the calibration parameters. On the theoretical side, it is observed in (11) that the coefficient of $\boldsymbol{t}_{CL}$ does not degrade to zero when $\boldsymbol{R}_{ij}^{C} \approx \boldsymbol{I}$, signifying that the solution of $\boldsymbol{t}_{CL}$ is not degenerate in CBA loss. Similar derivations can be applied to CA loss if we substitute (5) into (12) & (13). On the



(a) rvec-1 offset (rad)

(b) rvec-2 offset (rad)

(c) rvec-3 offset (rad)

(d) tvec-1 offset (m)

(e) tvec-2 offset (m)

(f) tvec-3 offset (m)

(g) scale offset

Fig. 8. Values of L and meeting statuses of (19) under unidimensional offsets of the optimization variable $\boldsymbol{x}$. The zero offset point is our optimization result $\boldsymbol{x}^*$. **Blue curves** (L): the curve of (16); **Black triangles** (IFeas): infeasible points that do not satisfy (19); **Red inverted triangles** (gt): the ground-truth point

experimental side, Fig. 8(d), 8(e), and 8(f) jointly demonstrate that the ground-truth solution (red inverted triangle) is close to the zero-offset point, indicating the accurate estimation of $\boldsymbol{t}_{CL}$ through our optimization method. In addition, these three curves indicate that the optimization in translational dimensions of $\boldsymbol{x}$ is almost convex. Despite the appearance of local minima in Fig. 8(d), the shape of L still guarantees the minimum value is achieved near the ground-truth point.

On the opposite, it is demonstrated in Fig. 8(a), 8(b) and 8(c) that the optimization in rotational dimensions is much easier. The HECalib constraint (19) successfully narrowed down the search range to the vicinity of the ground-truth and the variations of the function on either side of the zero-offset are almost strictly monotonic.

Concerning the scale dimension, the ground-truth scale is unknown because we do not have the ground-truth visual 3D map, so we adopt the initial scale $s^{(0)}$ from HECalib as the zero-offset point and plot relevant curves in Fig. 8(g). The scale dimension is not directly associated with our calibration task, but its value is required by the optimization process. Fig. 8(g) demonstrates that the HECalib constraint (19) also works in the scale dimension.

In terms of the function of other constraints (18) and (20), constraint (18) is predefined to determine the initial search range and (20) is applied to prevent runtime errors in extreme cases.

V. Conclusion

In this paper, we propose a novel targetless Lidar-camera calibration method based on cross-modality structure consistency. The performance of this method in degenerate scenes is demonstrated through experiments, and its theoretical degeneration-resistance property is derived. This property enables its application in scenarios where the vehicle only moves

forward without rotation, which is common in autonomous driving applications.

Whereas, due to the discontinuity of the loss functions, it is tricky to use derivative-based algorithms in our framework. In our future research, we aim to develop a continuous and differentiable form of the loss function for calibration. Derivative-based optimization methods are generally more computationally efficient than derivative-free ones. Additionally, with the availability of gradients, joint optimization of the extrinsic matrix and camera poses is expected to yield more accurate extrinsic parameters.

REFERENCES

[1] Y. Li, A. W. Yu, T. Meng, B. Caine, J. Ngiam, D. Peng, J. Shen, Y. Lu, D. Zhou, Q. V. Le, *et al.*, "Deepfusion: Lidar-camera deep fusion for multi-modal 3d object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 17 182–17 191.

[2] L. Caltagirone, M. Bellone, L. Svensson, and M. Wahde, "Lidar–camera fusion for road detection using fully convolutional neural networks," *Robotics and Autonomous Systems*, vol. 111, pp. 125–131, 2019.

[3] Z. Chen, J. Zhang, and D. Tao, "Progressive lidar adaptation for road detection," *IEEE/CAA Journal of Automatica Sinica*, vol. 6, no. 3, pp. 693–702, 2019.

[4] Z. Yuan, Q. Wang, K. Cheng, T. Hao, and X. Yang, "Sdv-loam: Semi-direct visual-lidar odometry and mapping," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[5] W. Wang, J. Liu, C. Wang, B. Luo, and C. Zhang, "Dv-loam: Direct visual lidar odometry and mapping," *Remote Sensing*, vol. 13, no. 16, p. 3340, 2021.

[6] C.-C. Chou and C.-F. Chou, "Efficient and accurate tightly-coupled visual-lidar slam," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 9, pp. 14 509–14 523, 2021.

[7] A. Dhall, K. Chelani, V. Radhakrishnan, and K. M. Krishna, "Lidar-camera calibration using 3d-3d point correspondences," *arXiv preprint arXiv:1705.09785*, 2017.

[8] 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 correspondences," *Optics express*, vol. 28, no. 2, pp. 2122–2141, 2020.

[9] L. Zhou, Z. Li, and M. Kaess, "Automatic extrinsic calibration of a camera and a 3d lidar using line and plane correspondences," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 5562–5569.

[10] D. Tsai, S. Worrall, M. Shan, A. Lohr, and E. Nebot, "Optimising the selection of samples for robust lidar camera calibration," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2021, pp. 2631–2638.

[11] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part I 9*. Springer, 2006, pp. 430–443.

[12] K. G. Derpanis, "The harris corner detector," *York University*, vol. 2, pp. 1–2, 2004.

[13] J. W. Weingarten, G. Gruener, and R. Siegwart, "Probabilistic plane fitting in 3d and an application to robotic mapping," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, vol. 1. IEEE, 2004, pp. 927–932.

[14] J.-K. Huang and J. W. Grizzle, "Improvements to target-based 3d lidar to camera calibration," *IEEE Access*, vol. 8, pp. 134 101–134 110, 2020.

[15] C. Guindel, J. Beltrán, D. Martín, and F. García, "Automatic extrinsic calibration for lidar-stereo vehicle sensor setups," in *2017 IEEE 20th international conference on intelligent transportation systems (ITSC)*. IEEE, 2017, pp. 1–6.

[16] X. Xu, L. Zhang, J. Yang, C. Liu, Y. Xiong, M. Luo, Z. Tan, and B. Liu, "Lidar–camera calibration method based on ranging statistical characteristics and improved ransac algorithm," *Robotics and Autonomous Systems*, vol. 141, p. 103776, 2021.

[17] C. Yuan, X. Liu, X. Hong, and F. Zhang, "Pixel-level extrinsic self calibration of high resolution lidar and camera in targetless environments," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7517–7524, 2021.

[18] N. Ou, H. Cai, J. Yang, and J. Wang, "Targetless extrinsic calibration of camera and low-resolution 3d lidar," *IEEE Sensors Journal*, 2023.

[19] J. Levinson and S. Thrun, "Automatic online calibration of cameras and lasers." in *Robotics: Science and Systems*, vol. 2, no. 7. Citeseer, 2013.

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

[21] X. Liu, C. Yuan, and F. Zhang, "Targetless extrinsic calibration of multiple small fov lidars and cameras using adaptive voxelization," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–12, 2022.

[22] G. Pandey, J. R. McBride, S. Savarese, and R. M. Eustice, "Automatic extrinsic calibration of vision and lidar by maximizing mutual information," *Journal of Field Robotics*, vol. 32, no. 5, pp. 696–722, 2015.

[23] P. Jiang, P. Osteen, and S. Saripalli, "Semcal: Semantic lidar-camera calibration using neural mutual information estimator," in *2021 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*. IEEE, 2021, pp. 1–7.

[24] 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 International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1110–1117.

[25] X. Lv, B. Wang, Z. Dou, D. Ye, and S. Wang, "Lccnet: Lidar and camera self-calibration using cost volume network," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 2894–2901.

[26] 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, 2022.

[27] R. Y. Tsai, R. K. Lenz, *et al.*, "A new technique for fully autonomous and efficient 3 d robotics hand/eye calibration," *IEEE Transactions on robotics and automation*, vol. 5, no. 3, pp. 345–358, 1989.

[28] F. C. Park and B. J. Martin, "Robot sensor calibration: solving ax= xb on the euclidean group," *IEEE Transactions on Robotics and Automation*, vol. 10, no. 5, pp. 717–721, 1994.

[29] R. Ishikawa, T. Oishi, and K. Ikeuchi, "Lidar and camera calibration using motions estimated by sensor fusion odometry," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 7342–7349.

[30] C. Park, P. Moghadam, S. Kim, S. Sridharan, and C. Fookes, "Spatiotemporal camera-lidar calibration: A targetless and structureless approach," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1556–1563, 2020.

[31] Z. Taylor and J. Nieto, "Motion-based calibration of multimodal sensor extrinsics and timing offset estimation," *IEEE Transactions on Robotics*, vol. 32, no. 5, pp. 1215–1229, 2016.

[32] H. Xu, G. Lan, S. Wu, and Q. Hao, "Online intelligent calibration of cameras and lidars for autonomous driving systems," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 3913–3920.

[33] J. Castorena, G. V. Puskorius, and G. Pandey, "Motion guided lidar-camera self-calibration and accelerated depth upsampling for autonomous vehicles," *Journal of Intelligent & Robotic Systems*, vol. 100, pp. 1129–1138, 2020.

[34] I. Fassi and G. Legnani, "Hand to sensor calibration: A geometrical interpretation of the matrix equation ax= xb," *Journal of Robotic Systems*, vol. 22, no. 9, pp. 497–506, 2005.

[35] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 13, no. 04, pp. 376–380, 1991.

[36] J. Zhang and S. Singh, "Loam: Lidar odometry and mapping in real-time." in *Robotics: Science and Systems*, vol. 2, no. 9. Berkeley, CA, 2014, pp. 1–9.

[37] H. Wang, C. Wang, C. Chen, and L. Xie, "F-loam : Fast lidar odometry and mapping," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[38] P. Dellenbach, J.-E. Deschaud, B. Jacquet, and F. Goulette, "Ct-icp: Real-time elastic lidar odometry with loop closure," 2021.

[39] G. Kim and A. Kim, "Scan context: Egocentric spatial descriptor for place recognition within 3d point cloud map," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 4802–4809.

[40] G. Kim, S. Choi, and A. Kim, "Scan context++: Structural place recognition robust to rotation and lateral variations in urban environments," *IEEE Transactions on Robotics*, vol. 38, no. 3, pp. 1856–1874, 2021.

[41] D. Zhou, Y. Dai, and H. Li, "Ground-plane-based absolute scale estimation for monocular visual odometry," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 2, pp. 791–802, 2019.

[42] H. Cai, N. Ou, and J. Wang, "Visual-lidar odometry and mapping with monocular scale correction and visual bootstrapping," *arXiv preprint arXiv:2304.08978v2*, 2023.

[43] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 3, pp. 611–625, 2017.

[44] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE transactions on robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[45] P. Moulon, P. Monasse, and R. Marlet, "Global fusion of relative motions for robust, accurate and scalable structure from motion," in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 3248–3255.

[46] S. Agarwal, N. Snavely, S. M. Seitz, and R. Szeliski, "Bundle adjustment in the large," in *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part II 11*. Springer, 2010, pp. 29–42.

[47] P. Moulon, P. Monasse, and R. Marlet, "Adaptive structure from motion with a contrario model estimation," in *Proceedings of the Asian Computer Vision Conference (ACCV 2012)*. Springer Berlin Heidelberg, 2012, pp. 257–270.

[48] J. L. Schonberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4104–4113.

[49] 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," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 10 197–10 202.

[50] L. Li, H. Li, X. Liu, D. He, Z. Miao, F. Kong, R. Li, Z. Liu, and F. Zhang, "Joint intrinsic and extrinsic lidar-camera calibration in targetless environments using plane-constrained bundle adjustment," *arXiv preprint arXiv:2308.12629*, 2023.

[51] W. Zhen, Y. Hu, J. Liu, and S. Scherer, "A joint optimization approach of lidar-camera fusion for accurate dense 3-d reconstructions," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3585–3592, 2019.

[52] D. Cernea, "OpenMVS: Multi-view stereo reconstruction library," 2020. [Online]. Available: https://cdcseacave.github.io/openMVS

[53] B. Nagy, L. Kovács, and C. Benedek, "Sfm and semantic information based online targetless camera-lidar self-calibration," in *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2019, pp. 1317–1321.

[54] D. Tu, B. Wang, H. Cui, Y. Liu, and S. Shen, "Multi-camera-lidar auto-calibration by joint structure-from-motion," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 2242–2249.

[55] P. Moulon, P. Monasse, and R. Marlet, "Adaptive structure from motion with a contrario model estimation," in *Proceedings of the Asian Computer Vision Conference (ACCV 2012)*. Springer Berlin Heidelberg, 2012, pp. 257–270.

[56] ——, "Global fusion of relative motions for robust, accurate and scalable structure from motion," in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 3248–3255.

[57] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g 2 o: A general framework for graph optimization," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 3607–3613.

[58] S. Agarwal, K. Mierle, and T. C. S. Team, "Ceres Solver," 3 2022. [Online]. Available: https://github.com/ceres-solver/ceres-solver

[59] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustmenta modern synthesis," in *Vision Algorithms: Theory and Practice: International Workshop on Vision Algorithms Corfu, Greece, September 21–22, 1999 Proceedings*. Springer, 2000, pp. 298–372.

[60] Y. Eldar, M. Lindenbaum, M. Porat, and Y. Y. Zeevi, "The farthest point strategy for progressive image sampling," *IEEE Transactions on Image Processing*, vol. 6, no. 9, pp. 1305–1315, 1997.

[61] Q.-Y. Zhou, J. Park, and V. Koltun, "Open3D: A modern library for 3D data processing," *arXiv:1801.09847*, 2018.

[62] M. A. Abramson, C. Audet, J. E. Dennis Jr, and S. L. Digabel, "Orthomads: A deterministic mads instance with orthogonal directions," *SIAM Journal on Optimization*, vol. 20, no. 2, pp. 948–966, 2009.

[63] C. Audet, S. L. Digabel, V. R. Montplaisir, and C. Tribes, "Nomad version 4: Nonlinear optimization with the mads algorithm," *arXiv preprint arXiv:2104.11627*, 2021.

[64] C. Audet, V. Béchard, and S. L. Digabel, "Nonsmooth optimization through mesh adaptive direct search and variable neighborhood search," *Journal of Global Optimization*, vol. 41, pp. 299–318, 2008.

[65] D. Gálvez-López and J. D. Tardós, "Bags of binary words for fast place recognition in image sequences," *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 1188–1197, October 2012.

[66] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 2012, pp. 3354–3361.

VI. Biography Section

Ni Ou Ni Ou received a Bachelor's degree in Electrical Engineering and Its Automation from China University of Mining Technology, Xuzhou, China, in 2020. He is currently pursuing a Ph.D. degree in the School of Automation at Beijing Institute of Technology, Beijing, China. His research focuses on SLAM systems, robotic sensor calibration, and point cloud registration.

Hanyu Cai Hanyu Cai obtained his Bachelor's Degree from Chongqing University, Chongqing, China, in 2021. He is currently pursuing a master's degree at Beijing Institute of Technology, Beijing. His research focuses on Structure from Motion and visual-inertial SLAM systems.

Junzheng Wang Junzheng Wang received his Ph.D. degree in control science and engineering from Beijing Institute of Technology, Beijing, China, in 1994. He currently serves as the Deputy Director of the State Key Laboratory of Intelligent Control and Decision of Complex Systems, and the director of the Key Laboratory of Servo Motion System Drive. He is also a senior member of the Chinese Mechanical Engineering Society and the Chinese Society for Measurement. His research expertises include motion control, electric hydraulic servo systems, and robotic perception.