IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 28, 2019, accepted August 22, 2019, date of publication August 27, 2019, date of current version September 9, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2937909

Extrinsic Calibration and Odometry for Camera-LiDAR Systems

CHENGHAO SHI, KAIHONG HUANG, QINGHUA YU, JUNHAO XIAO, (Member, IEEE), HUIMIN LU, AND CHENGGANG XIE

Robotics Research Center, College of Intelligence Science and Technology, National University of Defense Technology, Changsha 410073, China

Corresponding author: Junhao Xiao (junhao.xiao@ieee.org)

This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFC0803300, and in part by the National Science Foundation of China under Grant U1813205 and Grant 61773393.

ABSTRACT Most autonomous mobile robots are often equipped with monocular cameras and 3D LiDARs to perform vital tasks such as localization and mapping. In this paper, we present a two-stage extrinsic calibration method as well as a hybrid-residual-based odometry approach for such camera-LiDAR systems. Our extrinsic calibration method can estimate the relative transformation between the camera and the LiDAR with high accuracy, allowing us to better register the image and the point cloud data. After the calibration, our hybrid-residual-based odometry can be used to provide real-time, accurate odometry estimates. Our approach exploits both direct and indirect image features. The sensor motions are estimated by jointly minimizing reprojection residuals and photometric residuals in a nonlinear optimization procedure. Experiments are conducted to show the accuracy and robustness of our extrinsic calibration and odometry algorithms using both public and self-owned real-world datasets. The results suggest that our calibration method can provide accurate extrinsic parameters estimation without using initial values, and our odometry approach can achieve competitive estimation accuracy and robustness.

INDEX TERMS Camera-LiDAR system, hybrid-residual, extrinsic calibration, odometry.

I. INTRODUCTION

Simultaneous localization and mapping (SLAM) is a vital task of most autonomous mobile robots, and LiDAR and camera are the two most widely used sensors for such tasks.

SLAM algorithms using camera images are often referred as visual SLAM, they use abundant texture information to estimate ego-motion, either by using image features [1] or by using direct pixel intensities [2], [3]. Recent methods are capable of running at real-time with relatively high accuracy and using a monocular camera at a low cost. However, monocular visual SLAM suffers inherently from the scale-ambiguity problem.

SLAM algorithms using LiDAR point clouds are based on point-cloud registration using algorithms such as ICP [4] or NDT [5]. As the LiDAR can provide direct 3D point-measurements, LiDAR-based algorithms are often superior in mapping accuracy. However, due to the limited number of laser beams in the LiDAR, the point cloud are rather sparse in the vertical direction. This can pose difficulties in the registration.

Camera-LiDAR odometry tries to exploit both the laser point cloud data and the camera image information for estimating the ego-motion. A combination of accurate but sparse spatial measurements from laser scans and dense appearance information from camera images have the potential to complement each other for the task of motion estimation. Such a system has shown remarkable potential in both mapping accuracy and robustness in recent studies [6]–[8].

In this paper, we focus on camera-LiDAR systems and address at first the extrinsic calibration problem, and then, the odometry problem.

We study the camera-LiDAR extrinsic calibration problem so that the LiDAR and camera data can be accurately registered under a common reference frame, thereby laying a good foundation for camera-LiDAR odometry. We propose a two-stage calibration method inspired by the works of Huang and Stachniss [9] and Pandey *et al.* [10]. The first calibration stage aims to obtain a proper initial guess of the extrinsic parameters by exploiting constraints between the motions

The associate editor coordinating the review of this article and approving it for publication was Aysegul Ucar.

of individual sensors. The second calibration stage further refines the result by registering the LiDAR reflectivity information to the image intensity information using a metric call mutual information, assuming the camera and LiDAR have a sufficiently overlapped field of view (FOV). Experimental results show that our method can achieve accurate calibration results.

In the second part of this paper, we study the camera-LiDAR odometry problem. In our previous work [11], we proposed an odometry framework for RGB-Depth sensors which are similar to camera-LiDAR systems but can provide much denser close-range depth information (e.g., the Kinect sensor). For RGB-Depth sensors, our previous work jointly minimizes the reprojection residuals, the photometric residuals and the depth residuals altogether in a novel hybrid-residual-based optimization model. By exploiting multiple types of information, the robustness and accuracy of odometry estimates are greatly enhanced. However, in case of camera-LiDAR system, such an approach cannot be adapted easily as the depth information provided by LiDAR is much sparser than the image pixels, rendering the depth residuals and 3D landmark features rarely sufficient. In this paper, we address these issues and propose camera-LiDAR odometry which exploits both direct and indirect image features (assuming the camera and LiDAR have a sufficiently overlapped FOV). By performing depth-interpolation, our approach can obtain sufficient landmark features and provide accurate ego-motion estimates. Experiments are conducted to evaluate the accuracy and robustness of our odometry algorithms using both public and self-owned real-world datasets. The results suggest that our odometry approach can achieve competitive estimation accuracy and robustness.

The main contributions of this paper are as follows:

- An accurate two-stage camera-LiDAR extrinsic calibration approach;
- A novel hybrid-residual-based odometry method for camera-LiDAR systems;
- A novel occlusion-filtering approach for improving accuracy in both extrinsic calibration and odometry.

II. RELATED WORK

In this section, we provide an overview of related work for extrinsic calibration and odometry problems regarding camera-LiDAR systems.

A. EXTRINSIC CAMERA-LiDAR CALIBRATION

Previous extrinsic calibration methods can be categorized into two categories: the target-based and target-less methods.

Target-based methods involve using fiducial objects with known geometry to perform the calibration [12]–[14]. Such kind of approach can often provide accurate calibration results. However, due to the requirement of fiducial objects and human interventions, these calibration methods cannot be done on-the-fly or in-field easily.

Many works are focused on target-less calibration, including motion-based and feature-based calibration methods. Motion-based calibration methods [9], [15], [16] estimate the extrinsic parameters by using a series of relative pose measurements of the sensors, while feature-based calibration methods [10], [17]–[21] extract features from the camera and lidar data and then estimate the parameters by minimizing the reprojection error of the corresponding features.

Among the motion-based methods, Shiu and Ahmad [15] solve a set of homogeneous transform equations in the form of $\mathbf{AX} = \mathbf{XB}$, where $(\mathbf{A}, \mathbf{B})$ represents the relative motions of two modalities and $\mathbf{X}$ is the six Degree of Freedom (DOF) relative transformation parameters between the modalities. Huang and Stachniss [9] estimate not only the extrinsic parameters but also the motion errors by using the Gauss-Helmert paradigm, thereby improving the calibration accuracy and robustness significantly.

Among the feature-based methods, Liao *et al.* [17] and Moghadam *et al.* [18] use 3D line features from the point clouds and corresponding 2D line segments from camera images when exploiting the statistical dependence of the sensor data. Boughorbal *et al.* [19] and Williams *et al.* [21] estimate the extrinsic parameters by using a χ^2 test to maximize the correlation between the sensor data. Napier *et al.* [20] optimize a correlation measure between the laser reflectivity and grayscale image values to calibrate a 2D push-broom lidar and a camera system. Pandey *et al.* [10] address the camera-LiDAR calibration problem by maximizing the mutual information between the sensor measured surface intensities. Such a method can achieve accurate calibration results if provided sufficient data and a proper initial guess.

According to our experience, feature-based calibration approaches can often provide satisfying results if a proper initial guess is provided. Motion-based calibration approaches, on the other hand, are often not accurate enough for proper point-cloud-to-image registration, but it can tolerate a noisy initial guess and even work without using one. Therefore, we propose to combine the advantages of both methods by using a two-stage calibration method where a motion-based approach serves as an initial calibration, and a mutual-information-based approach further refines the estimate. The work of Liao *et al.* [17] has a similar idea, but they rely on a line-feature based method as the second stage, which has a strong dependence on the structured environment and is likely to fail in an unstructured scene. Our approach does not has such limitation and can work with nature scenes.

B. CAMERA-LiDAR ODOMETRY

There are direct and indirect methods for camera-LiDAR odometry problems. The indirect approaches, also known as feature-based, extract image keypoints from the image and their depth information from the LiDAR point cloud. A set of such keypoints are then used to estimate the sensor motions. For example, Zhang *et al.* [8] proposed a method utilizing keypoints both with and without depth measurements. The motion is estimated by minimizing the reprojection errors of keypoints. Andreasson *et al.* [7], [22] utilize all the keypoints as well, but they address the missing depth problem

with vision-based depth interpolation. Direct methods are a kind of methods which do not extract image features but use pixel intensity values directly. Shin *et al.* [6] use the projected laser points to perform a multi-frame photometric optimization the same as the DSO visual-SLAM odometry [2]. Della Corte *et al.* [23] proposed a multi-cue photometric point cloud registration approach which considers color, depth, and normal information, however, it is designed for RBG-Depth sensors.

III. THE TWO-STAGE CALIBRATION FRAMEWORK FOR LiDAR AND CAMERA

In this section, we present our two-Stage extrinsic calibration method for camera-LiDAR systems. Our calibration approach combines a motion-based approach and a mutual-information-based approach. Our motivation is that, on the one hand, the mutual-information-based approach can provide more accurate calibration results but require a proper initial guess of the extrinsic calibration. On the other hand, the motion-based approach can work without an initial guess, but its estimate has limited accuracy. Thus, the combination of the two approaches can yield advantages.

In the following discussion, we assume the camera and LiDAR to be time-synchronized (e.g., by using hardware trigger) and the intrinsic calibration parameters of the monocular camera (i.e., focal length, camera center, distortion coefficients of the lens) are pre-calibrated. The LiDAR coordinate system is originated at the LiDAR optical center with its x-axis pointing to the forward, y-axis pointing left, and z-axis pointing upward coinciding with the LiDAR principal axis. The camera coordinate system is originated at the camera optical center with its x-axis pointing to the downward, y-axis pointing left, and z-axis pointing forward coinciding with the camera principal axis. The world coordinate system coincides with the camera coordinate system at the starting position. The extrinsic parameters used in this paper are the rotations and translations of the LiDAR coordinate frame relative to the camera coordinate frame, which are defined in the camera coordinate frame.

A. MOTION-BASED FIRST STAGE CALIBRATION

The first calibration stage of our approach is a motion-based method which exploits the constraint equation $\mathsf{AX} = \mathsf{XB}$ between the ego-motions/trajectories (i.e., A, B) of individual sensors to estimate a coarse initial guess of the extrinsic parameters (i.e., X). We model this problem with the Guass-Helmert least-squares formulation, which is described in this section.

To perform the motion-based calibration, one needs to instruct the robot to perform six-DOF motions and obtain ego-centric trajectories of both the LiDAR and the camera. We denote the trajectories of the two sensors as:

$$\boldsymbol{l}_i \stackrel{\text{def}}{=} \begin{bmatrix} \boldsymbol{r}_{ai} \\ \boldsymbol{r}_{ci} \\ \boldsymbol{t}_{ai} \\ \boldsymbol{t}_{ci} \end{bmatrix}, \quad i = 1, \cdots, N \tag{1}$$

where $\boldsymbol{r}_{ai}$ and $\boldsymbol{t}_{ai}$ are the rotations and translations of the LiDAR trajectories; $\boldsymbol{r}_{ci}$ and $\boldsymbol{t}_{ci}$ are the rotations and translations of the camera trajectories. The notation $\boldsymbol{r}$ represents a angle-axis vector of a rotation matrix, and $\mathsf{R}(\cdot)$ represents the rotation matrix of the corresponding angle-axis vector. We also denote the unknown extrinsic parameters as:

$$\boldsymbol{x} \stackrel{\text{def}}{=} \begin{bmatrix} \boldsymbol{\eta} \\ \boldsymbol{\xi} \end{bmatrix} \tag{2}$$

where $\boldsymbol{\eta}$ is the angle-axis vector of the rotation parameter and $\boldsymbol{\xi}$ is the translation parameter.

Base on the constraint equation $\mathsf{AX} = \mathsf{XB}$, we define two error functions which relate the sensor trajectories data $\boldsymbol{l}_i$ to the unknown extrinsic parameters $\boldsymbol{x}$:

$$\boldsymbol{g}_t(\boldsymbol{x}, \boldsymbol{l}_i) \stackrel{\text{def}}{=} [\mathsf{R}(\boldsymbol{r}_{ai}) - \mathsf{I}_3]\boldsymbol{\xi} + \boldsymbol{t}_{ai} - \mathsf{R}(\boldsymbol{\eta})\boldsymbol{t}_{ci}, \tag{3}$$

$$\boldsymbol{g}_r(\boldsymbol{x}, \boldsymbol{l}_i) \stackrel{\text{def}}{=} \boldsymbol{r}_{ai} - \mathsf{R}(\boldsymbol{\eta})\boldsymbol{r}_{ci}. \tag{4}$$

In the noise-free case, a true solution $\boldsymbol{x}^*$ will fulfill

$$\boldsymbol{g}(\boldsymbol{x}^*, \boldsymbol{l}_i) \stackrel{\text{def}}{=} \begin{bmatrix} \boldsymbol{g}_t(\boldsymbol{x}^*, \boldsymbol{l}_i) \\ \boldsymbol{g}_r(\boldsymbol{x}^*, \boldsymbol{l}_i) \end{bmatrix} = \boldsymbol{0}. \quad \forall i. \tag{5}$$

However, this is often not the case and there will be inevitable measurement errors, we therefore denote the unknown trajectory errors as $\boldsymbol{\epsilon}_i$:

$$\boldsymbol{\epsilon}_i = \boldsymbol{l}_i - \boldsymbol{l}_i^0, \tag{6}$$

with $\boldsymbol{l}_i^0$ being the raw trajectory data.

Then, our task is to estimate both the extrinsic parameters $\boldsymbol{x}$ and the trajectory errors $\boldsymbol{\epsilon}_i$ by solving

$$\begin{aligned} &\underset{\boldsymbol{x}, \{\boldsymbol{\epsilon}_i\}}{\text{argmin}} \quad \sum_i^N \|\boldsymbol{\epsilon}_i\|_{\mathsf{W}_i}^2 \\ &\text{subject to} \quad \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{\epsilon}_i + \boldsymbol{l}_i^0) = \boldsymbol{0}, \quad \forall i, \end{aligned} \tag{7}$$

assuming $\boldsymbol{\epsilon}_i$ follows a normal distribution and $\mathsf{W}_i \stackrel{\text{def}}{=} \boldsymbol{\Sigma}_{l_i l_i}^{-1}$ is its inverse covariance matrix.

We refer to Equation (7) as a Guass-Helmert least-squares problem and it can be solved by iterating between linearizing the model and adjusting the parameters $\boldsymbol{x}$ and $\boldsymbol{\epsilon}_i$. To be more specific, assume that in the k-th iteration, the corrected measurements $\boldsymbol{l}_i^k$ as well as the estimated parameters $\boldsymbol{x}^k$ will be updated by

$$\boldsymbol{l}_i^{k+1} = \Delta\boldsymbol{l}_i + \boldsymbol{l}_i^k \quad \text{and} \quad \boldsymbol{x}^{k+1} = \Delta\boldsymbol{x} + \boldsymbol{x}^k. \tag{8}$$

We first linearize the non-linear constraint equation $\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{l}) = \boldsymbol{0}$ around $(\boldsymbol{x}^k, \boldsymbol{l}_i^k)$ by:

$$\boldsymbol{g}(\boldsymbol{x}^k, \boldsymbol{l}_i^k) + \mathsf{J}_i^k \Delta\boldsymbol{x} + \mathsf{L}_i^k \Delta\boldsymbol{l}_i = \boldsymbol{0}, \tag{9}$$

where J_i^k are the Jacobians of $\boldsymbol{g}$ with respect to $\boldsymbol{x}$ and L_i^k are the Jacobians of $\boldsymbol{g}$ with respect to $\boldsymbol{l}$. Then the Equation (7) becomes

$$\begin{aligned} &\underset{\Delta\boldsymbol{x}, \{\Delta\boldsymbol{l}_i\}}{\text{argmin}} \quad \sum_i \|\Delta\boldsymbol{l}_i + \boldsymbol{\epsilon}_i^k\|_{\mathsf{W}_i}^2 \\ &\text{s.t.} \quad \boldsymbol{g}(\boldsymbol{x}^k, \boldsymbol{l}_i^k) + \mathsf{J}_i^k \Delta\boldsymbol{x} + \mathsf{L}_i^k \Delta\boldsymbol{l}_i = \boldsymbol{0}, \quad \forall i, \end{aligned} \tag{10}$$

in which $\boldsymbol{\epsilon}_i^k \stackrel{\text{def}}{=} \boldsymbol{l}_i^k - \boldsymbol{l}_i^0$ are the corresponding k-th iteration trajectory errors.

Equation (10) can be solved using the method of Lagrange multipliers and its solution is:

$$\Delta\boldsymbol{x} = \Big(\sum_i \mathbf{J}_i^\mathsf{T} \Lambda_i \mathbf{J}_i\Big)^{-1} \sum_i \mathbf{J}_i^\mathsf{T} \Lambda_i \boldsymbol{c}_i, \tag{11a}$$

$$\Delta\boldsymbol{l}_i = \boldsymbol{\Sigma}_{l_i l_i} \mathbf{L}_i^\mathsf{T} \Lambda_i (\boldsymbol{c}_i - \mathbf{J}_i \Delta\boldsymbol{x}) - \boldsymbol{\epsilon}_i^k, \tag{11b}$$

$$\text{with } \Lambda_i \stackrel{\text{def}}{=} (\mathbf{L}_i \boldsymbol{\Sigma}_{l_i l_i} \mathbf{L}_i^\mathsf{T})^{-1}, \tag{11c}$$

$$\text{and } \quad \boldsymbol{c}_i \stackrel{\text{def}}{=} -\boldsymbol{g}(\boldsymbol{x}^k, \boldsymbol{l}_i^k) + \mathbf{L}_i \boldsymbol{\epsilon}_i^k. \tag{11d}$$

Using Equation (11), we can obtain the estimate $(\boldsymbol{x}^{k+1}, \boldsymbol{l}_i^{k+1})$ and repeat the process until convergence.

Note that motion-based algorithm will degenerate if the trajectory measurements do not contain full 3D rotation [9]. Thus, one should ensure that the provided sensor trajectories contain sufficient movements in all six-DOF.

When working with trajectories of a monocular camera, their scale factors are often unknown due to the inherent scale-ambiguity problem. We propose to determine such scale factors (denoted as λ) by using the fact that the norm of a translation is the same for both camera and LiDAR in case of pure translations, i.e.

$$\|\lambda \boldsymbol{t}_{ci}\| = \|\boldsymbol{t}_{ai}\|, \quad \text{if } \|\boldsymbol{r}_{ci}\| = 0. \tag{12}$$

Thus, one can first instruct the robot to perform pure translation (which can be carried out easily for ground vehicles in most cases), then use such trajectories to estimate the scale parameter of the camera by solving

$$\underset{\lambda}{\text{argmin}} \sum_{i=1}^{M} \left(\|\lambda \boldsymbol{t}_{ci}\| - \|\boldsymbol{t}_{ai}\|\right)^2, \quad i = 1, \cdots, M, \tag{13}$$

where M is the number of motion measurement segments which contain no rotations, and $\|\cdot\|$ represents 2-norm. We can solve λ by making the first derivative of the Equation (13) equals to zero, i.e.

$$2\sum_{i=1}^{M} \|\boldsymbol{t}_{ci}\| \left(\lambda \|\boldsymbol{t}_{ci}\| - \|\boldsymbol{t}_{ai}\|\right) = 0. \quad i = 1, \cdots, M. \tag{14}$$

The solution to λ reads as:

$$\lambda = \frac{\sum_{i=1}^{M} \|\boldsymbol{t}_{ci}\| \, \|\boldsymbol{t}_{ai}\|}{\sum_{i=1}^{M} \|\boldsymbol{t}_{ci}\|^2}. \quad i = 1, \cdots, M. \tag{15}$$

The scale-corrected camera trajectory is now $(\boldsymbol{r}_{ci}, \boldsymbol{t}'_{ci})$ with

$$\boldsymbol{t}'_{ci} \stackrel{\text{def}}{=} \lambda \boldsymbol{t}_{ci}. \tag{16}$$

Note that λ needs to be calculated for each camera trajectory.

B. MUTUAL-INFORMATION-BASED SECOND STAGE CALIBRATION

The second calibration stage of our approach is a mutual-information-based method which further refines the extrinsic parameters by registering the LiDAR reflectivity information to the camera image intensity information using a metric call mutual information.

For a scene point co-observed by the LiDAR and camera, one can obtain three types of measurement: its range, reflectivity, and image intensity. Both the range and reflectivity information are provided by the LiDAR, and the reflectivity measures how much percentage of the infra-red pulse is reflected back to the LiDAR receiver by the scene point. Such kind of reflectivity information is often similar or well-aligned to the image intensity information in a real-world environment, and hence can be used to calibrate the two sensors.

To perform mutual-information-based calibration, we rely on a set of scene points co-observed by the LiDAR and camera. Assuming an initial extrinsic parameters $(\boldsymbol{\eta}, \boldsymbol{\xi})$ for stage one calibration are obtained, a 3D LiDAR point $\boldsymbol{s}$ can be projected to the camera image plane at the pixel location of

$$\boldsymbol{c} \stackrel{\text{def}}{=} \pi(\mathbf{K}(\mathbf{R}(\boldsymbol{\eta})\boldsymbol{s} + \boldsymbol{\xi})), \tag{17}$$

where $\mathbf{K}$ is the camera matrix defined by camera intrinsic parameters, and $\pi(\cdot)$ is the Euclidean normalization function that transforms homogeneous coordinates into (inhomogeneous) pixel coordinates, i.e.

$$\pi\left(\begin{bmatrix} u \\ v \\ w \end{bmatrix}\right) \stackrel{\text{def}}{=} \begin{bmatrix} u/w \\ v/w \end{bmatrix}. \tag{18}$$

If the projected 2D point $\boldsymbol{c}$ lies inside the camera image, then we refere to the scene point $\boldsymbol{s}$ as a co-observed point. Such a set of co-observed points are used in our second calibration stage. We denote their LiDAR reflectivity values as random variable S and their image intensities as random variable C. The mutual information between S and C is a measure of statistical dependence (or correlation) occurring between the two random variables, which we define as the entropy:

$$\text{MI}(S, C) \stackrel{\text{def}}{=} \text{H}(S) + \text{H}(C) - \text{H}(S, C), \tag{19}$$

with

$$\text{H}(S) \stackrel{\text{def}}{=} -\sum_{S_i \in S} p_S(S_i) \log p_S(S_i), \tag{20}$$

$$\text{H}(C) \stackrel{\text{def}}{=} -\sum_{C_i \in C} p_C(C_i) \log p_C(C_i), \tag{21}$$

$$\text{H}(S, C) \stackrel{\text{def}}{=} -\sum_{S_i \in S} \sum_{C_i \in C} p_{SC}(S_i, C_i) \log p_{SC}(S_i, C_i), \tag{22}$$

and

$$p_S(S = S_i) \stackrel{\text{def}}{=} \frac{1}{n} \sum_{j=1}^{n} G_\omega(S - S_j), \tag{23}$$

$$p_C(C = C_i) \stackrel{\text{def}}{=} \frac{1}{n} \sum_{j=1}^{n} G_\omega(C - C_j), \tag{24}$$

$$p_{SC}(S = S_i, C = C_i) \stackrel{\text{def}}{=} \frac{1}{n} \sum_{j=1}^{n} G_\Omega\left(\begin{bmatrix} S \\ C \end{bmatrix} - \begin{bmatrix} S_j \\ C_j \end{bmatrix}\right). \tag{25}$$

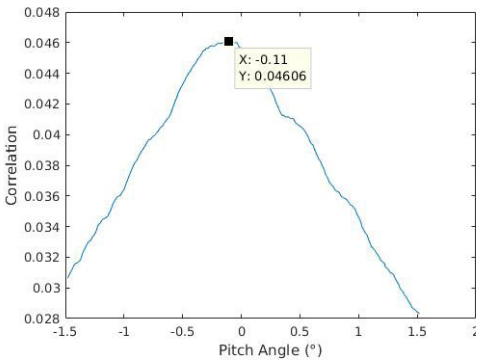


FIGURE 1. The mutual-information between the LiDAR reflectivity and camera intensity as a function of one of the extrinsic parameters, pitch angle. All the other parameters are fixed at their ground truth. The maximum mutual-information is observed at the true pitch angle -0.11°.

Here, $\mathrm{H}(\cdot)$ represent the amount of uncertainty. S_i, C_i represent the observations of the random variables, n is the total number of co-observed scene points. $p(\cdot)$ represent marginal and joint probabilities of the random variables, which are kernel density estimates by using Gaussian kernels $G_\omega(\cdot)$ and $G_\Omega(\cdot)$ with a bandwidth of ω and Ω respectively. Ω is determined by Silverman's rule of thumb [24]:

$$\Omega = 1.06 n^{\frac{1}{5}} \begin{bmatrix} \sigma_S & 0 \\ 0 & \sigma_C \end{bmatrix}. \tag{26}$$

Under the correct extrinsic parameters, the mutual-information between the LiDAR reflectivity information and the image intensity information should be maximized [10], i.e.

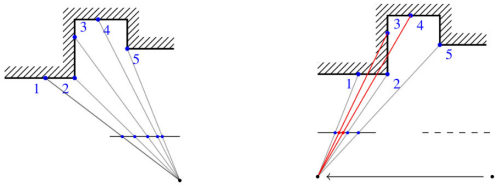
$$\boldsymbol{x}^* = \underset{\boldsymbol{x}}{\operatorname{argmax}}\, \mathrm{MI}(S, C; \boldsymbol{x}). \tag{27}$$

Fig. 1 depicted an experimental result to support this argument. Here, the mutual-information is calculated under different extrinsic parameter (i.e. pitch angle), it reaches the maximum value at the correct pitch angle.

Although the objective function in Equation (27) is not a direct function of the calibration parameters, it can still be solved numerically with a gradient-descent approach the same as in [25].

C. OCCLUSION FILTERING FOR ACCURACY IMPROVEMENTS

Throughout this paper, we assume each triplet of range, reflectivity, and image intensity values reported by the camera-LiDAR system are originated from one co-observed scene point. However, such assumption will be violated if the scene point is occluded due to a viewpoint difference between the LiDAR and camera, which often happens for a point located at object borders. In that case, the reflectivity and image intensity values are originated from different scene points and should be removed from the estimation process. Otherwise, the estimation accuracy will deteriorate, and the estimation result will contain systematic errors.



(a) camera: 1 2 3 4 5 (b) LiDAR: 1 3 4 2 5

FIGURE 2. The pixels order of projected 3D points will change if they are occluded. In the camera view, we label the 3D points from left to right as (1, 2, 3, 4, 5). In the LiDAR view, point 3 and 4 are occluded, leading to a different pixel order, which is now (1, 3, 4, 2, 5) from left to right. We exploit such pixel order changes to perform occlusion detection for sparse 3D point clouds.

To overcome the occlusion problem, we propose a novel occlusion filtering method to predict which scene point in a sparse point cloud will be occluded due to sensor viewpoint differences. The key observation of our approach is that whenever parts of a point cloud are occluded in one sensor view, the relative pixel order of the projected point cloud in another sensor view will be different.

Fig. 2 illustrated an example. Assume in the camera view, and there are five scene points labeled as (1, 2, 3, 4, 5) from left to right. Point 3 and 4 are occluded in the LiDAR view due to a translational viewpoint difference. In this case, the projected pixel order of the same point-set becomes (1, 3, 4, 2, 5) in the LiDAR view, again from left to right. In other words, the order of points (3, 4) and point 2 is swapped in the two sensor views, and we could exploit such changes to detect occlusion.

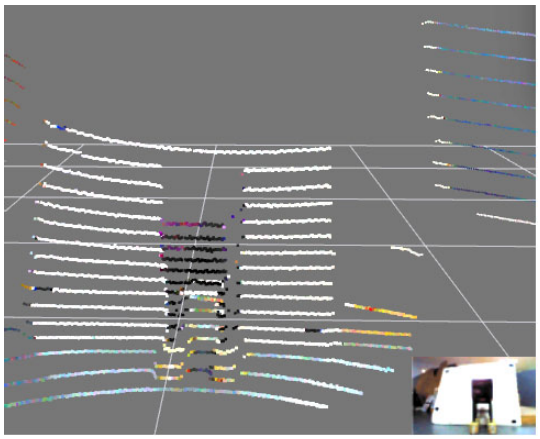
We generalized this idea and developed an algorithm for occlusion filtering for sparse 3D point clouds. Fig. 3 shows an example result on real-world data. The detected occluded points are marked red in Fig. 3a. As shown, the corresponding points in Fig. 3b are drawn incorrect color. Such points should therefore be removed from the mutual-information-based calibration.

Note that the number of occluded points is likely to increase when the observed object gets closer, thus the occlusion filtering becomes more necessary in an indoor environment.

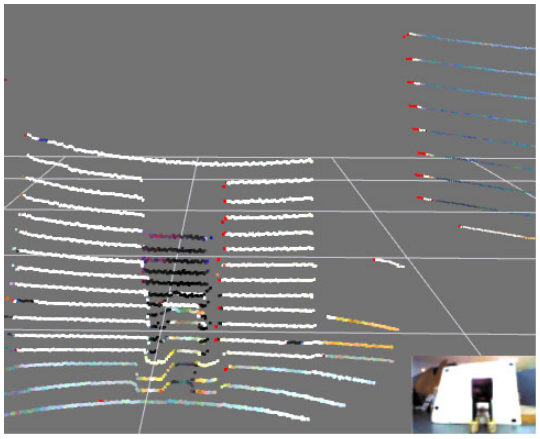
IV. HYBRID-RESIDUAL-BASED ODOMETRY FOR CAMERA-LiDAR SYSTEM

After the calibration, we propose a hybrid-residual-based camera-LiDAR odometry method for accurate real-time motion estimation.

The motion estimation problem of the camera-LiDAR system is equivalent to the problem of estimating the camera motion using both the image and point cloud data. Our approach relies on two types of the landmark: the photometric landmarks and the reprojection landmarks. Both types of the landmark are a set of pixels extracted from the camera image base on certain criterions (Sec. IV-A). We assign each of the landmarks with a depth value either directly from a LiDAR range readings or through a depth interpolation



(a) before detection



(b) after detection

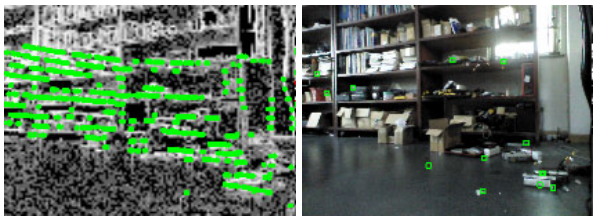
FIGURE 3. An occlusion detection result on real-world data, image (a) shows a result that after projecting an image onto point cloud. Image (b) shows the occlusion detection result, and red points indicate the occlude points which are detected by our method. These points are clearly drawn incorrect color information, as shown in the image (a). The bottom-right of each image shows the image being projected.

algorithm (Sec. IV-B). Each landmark introduces a photometric or a reprojection residual term in our motion estimation model (Sec. IV-C).

A. LANDMARK EXTRACTION

For photometric landmarks, we choose pixels which have the highest gradient magnitude values because they are considered to be more salient and robust to illumination changes. To extract these photometric landmarks, we first compute the gradient of each pixel by applying the Sobel filter to the grayscale camera image. We then construct a histogram of all gradient values to determine a threshold, and any pixel with a gradient value exceeds the threshold are selected. Fig. 4a illustrated an example of extracted photometric landmarks (marked green) on top of a gradient magnitude image.

For reprojection landmarks, we use the Oriented FAST and rotated BRIEF (ORB) feature detector and descriptor [26] to perform keypoint extraction and matching. During the keypoint extraction, the image is divided into grids, and the ORB feature detector is applied to each grid. In this way, keypoint is extracted homogeneously in the image, and the number of features in each grid is similar. Fig. 4b illustrated an example of extracted reprojection landmarks (green mark) on top of a color image.



(a) photometric landmarks (b) reprojection landmarks

FIGURE 4. An example of photometric and reprojection landmarks.

B. DEPTH INTERPOLATION

After a pixel or keypoint is selected as a landmark, it has to be assigned with a depth value in order to be useful for the camera motion estimation. This can be done easily if the pixel/keypoint correspond to a LiDAR point, as the depth value will be directly available from the LiDAR range reading after the coordinate transformation. However, because the range measurements from LiDAR are much sparser than the image pixels, most of the landmarks will not correspond to a LiDAR point and hence has to be discarded if no special treatment is applied, rendering the camera motion estimation process unable to proceed. To address this problem, we propose to approximate the missing depth values with interpolated ones, using a method described in this section.

First, let us denote a depthless pixel or keypoint as

$$\boldsymbol{c}_i = (r_i, g_i, b_i, u_i, v_i), \tag{28}$$

where r_i, g_i, b_i are the three RGB values of the pixel and u_i, v_i are the pixel coordinates. Then, we denote a LiDAR-point-projected pixel as

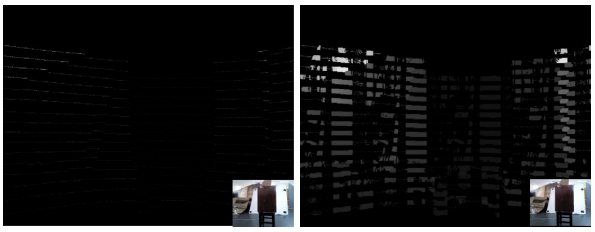
$$\boldsymbol{c}_j = (r_j, g_j, b_j, u_j, v_j, d_j), \tag{29}$$

where d_j is the known depth of the LiDAR point. Our task is to estimate the depth value d of a pixel $\boldsymbol{c}_i$ given a set of pixels $\boldsymbol{c}_j$.

One can use the nearest-neighbor interpolation method to select a $\boldsymbol{c}_j$ closest to the $\boldsymbol{c}_i$ in terms of coordinate distances. However, a more reasonable method should also consider the color similarity, as the approximation of depth is related to not only pixel distance but also color. Therefore, we formulate this as a Maximum Likelihood Estimate (MLE) problem with the likelihood function defined as

$$p(\boldsymbol{c}_j, \boldsymbol{c}_i) = e^{-\frac{(u_i-u_j)^2+(v_i-v_j)^2}{\sigma_P^2} - \frac{(r_i-r_j)^2+(g_i-g_j)^2+(b_i-b_j)^2}{\sigma_C^2}} \tag{30}$$

where σ_C^2 is the variance for color differences, σ_P^2 is the variance of coordinate differences which is calculated from



(a) before interpolation (b) after interpolation

FIGURE 5. An example depth image before and after the interpolation. The pixel is drawn as a gray value according to its depth value. The bottom-right of each image shows the corresponding color image.

a distance matrix. Each pixel can generate its distance matrix which has a size the same as the image. Each item in the matrix refers to the distance between that pixel and all other pixels. The mean square deviation of the distance matrix is used as the variance of the pixel point. Pixel color can be regarded as three-DOF coordinates, hence the variances of pixel color are calculated using a similar process.

Fig. 5 shows an example depth image before and after the interpolation. Please note that the depth interpolation is only performed for the pixels which are within a 5 pixel distance of those pixels with depth measurements. Cause the depth interpolation is meaningless when two pixels are too far apart. Compared to other method such as [27], our approach does not require solving high-dimensional linear equations, making the interpolation process less time-consuming and can be run in real-time.

C. OPTIMIZATION MODEL

Once the photometric and reprojection landmarks are extracted and assigned with depth values, they are used to estimate the camera motion.

We model such problem as a MLE problem:

$$\underset{\boldsymbol{\phi}}{\operatorname{argmax}} \prod_{n=1}^{2} p((\boldsymbol{a}_{on}, \boldsymbol{a}_{wn})|\boldsymbol{\phi}), \tag{31}$$

where $\boldsymbol{\phi}$ is the six-DOF motion parameters of the camera frame with respect to the world coordinate system, $p((\boldsymbol{a}_{on}, \boldsymbol{a}_{wn}))$ is the maximum likelihood probability, $\boldsymbol{a}_{on}$ is the landmark in the world coordinate, $\boldsymbol{a}_{wn}$ is the corresponding landmark in the camera coordinate.

Under the Gaussian noise assumption, the MLE problem in Equation (31) is equivalent to a weighted nonlinear least squares problem which contains both reprojection residuals and photometric residuals:

$$\underset{\boldsymbol{\phi}}{\operatorname{argmin}} \left(\sum_{i=1}^{N_{rep}} \frac{e_{rep,i}^2}{\sigma_{rep,i}^2} + \sum_{j=1}^{N_{pho}} \frac{e_{pho,j}^2}{\sigma_{pho,j}^2} \right), \tag{32}$$

where $e_{rep,i}$ is the reprojection residual, $e_{pho,j}$ is the photometric residual, $\sigma_{rep,i}^2$ and $\sigma_{pho,j}^2$ are the observation noise covariance, which are estimated online based on the distribution of observations.

The reprojection residual $e_{rep,i}$ is defined as the coordinate difference between a detected reprojection landmark and its corresponding reprojected point in the image

$$e_{rep,i} \overset{\text{def}}{=} \left\| [u_{oi}, v_{oi}]^{\mathsf{T}} - [u_{wi}, v_{wi}]^{\mathsf{T}} \right\|, \quad i = 1, \cdots, N_{rep}, \tag{33}$$

where $[u_{oi}, v_{oi}]^{\mathsf{T}}$ are the pixel coordinates of observed keypoint, and $[u_{wi}, v_{wi}]^{\mathsf{T}}$ are the pixel coordinates of corresponding reprojected point which is determined by keypoint extraction and matching.

The photometric residuals $e_{pho,j}$ is defined as the pixel intensity distance of a photometric landmark and its reprojected pixel in the current image

$$e_{pho,j} \overset{\text{def}}{=} \mathcal{I}([u_{oj}, v_{oj}]^{\mathsf{T}}) - \mathcal{I}(\boldsymbol{a}_{wj}), \quad j = 1, \cdots, N_{pho}, \tag{34}$$

where $\mathcal{I}(\boldsymbol{a}_{wj})$ is the intensity value of the photometric landmark when it is first extracted, and $\mathcal{I}([u_{oj}, v_{oj}]^{\mathsf{T}})$ is the intensity value of corresponding pixel in the image.

The nonlinear optimization problem in Equation (32) is solved with a similar approach as in [11]. To increase the robustness with respect to outliers, residuals larger than a certain threshold is considered as an outlier and removed from the optimization.

V. EXPERIMENTS AND RESULTS

In this paper, we propose a two-stage extrinsic calibration method as well as a hybrid-residual-based odometry approach for camera-LiDAR systems. We conduct experiments to show that (i) our two-stage calibration approach can provide accurate extrinsic parameter estimates, without requiring an initial guess, (ii) our occlusion detection approach can improve the overall calibration accuracy, (iii) our hybrid-residual-based odometry can achieve competitive localization accuracy even with fast motion.

The experiments consist of two parts. In the first part, we perform extrinsic calibration using real-world data, and in the second part, we evaluate the proposed camera-LiDAR odometry approach and make comparisons to other state-of-the-art algorithms.

The main robotic platform used in the experiments is a self-developed tracked robot equipped with a camera-LiDAR system, as shown in Fig. 6. All our methods are implemented in C++ and run on this platform with a 2.2 GHz eight-core CPU processor.

A. EXTRINSIC CALIBRATION ON REAL ROBOT

To validate our calibration method, we perform extrinsic calibration on our own robotic platform.

We first acquired a set of sensor trajectories to estimate an initial guess for the extrinsic parameters, using our stage-one calibration algorithm. We instructed the robot to perform a set of movements, and at the same time, estimate the ego-motion of both the LiDAR and the camera using a LiDAR odometry algorithm named LOAM [28] and visual odometry named ORB-SLAM [1].

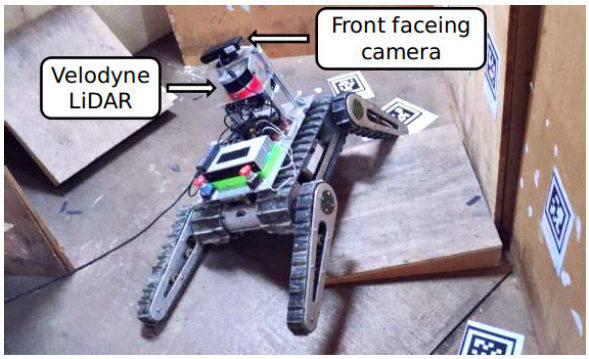


FIGURE 6. Our experimental robot platform which is equipped with a camera-LiDAR system.

TABLE 1. Extrinsic calibration results on real robot.

	Translation (m)			Mean Difference (m)
	x	y	z	
reference	0.0353	0.0372	0.1036	—
first-stage	0.0674	0.0408	0.0821	0.0191
second-stage	0.0351	0.0302	0.1031	**0.0026**
Mean difference in translation is the mean value of each axis difference.				
	Rotation $(^\circ)$			Mean Difference $(^\circ)$
	$roll$	$pitch$	yaw	
reference	-0.3012	0.1132	0.2031	—
first-stage	0.0312	-0.0312	0.0314	0.3265
second-stage	-0.2449	0.1302	0.1567	**0.0620**
Mean difference in rotation is the difference of angle parameter of the angle-axis vector.				

The performed robot movements contain both pure translations and full six-DOF motions. The pure translations are used for recovering the scale factor of the estimated camera trajectories, using the method described in Sec. III-A. The six-DOF motions ensure the motion-based calibration algorithm does not degenerate during the estimation, allowing all six extrinsic parameters to be recovered.

In the first-stage motion-based calibration, we used a total of 31 pairs of motion data to obtain the initial guess. In the second-stage mutual-information-based calibration, we used 20 pairs of selected distinctive sensor data to refine the estimate. Both the intermediate and final extrinsic parameters are shown in TABLE 1. The rotational parameters are represented as Euler angles, i.e., the roll, pitch and yaw angles.

As this is a real-world experiment, the ground truth of extrinsic parameters is unfortunately not available. In order to perform a quantitative evaluation, we obtained a reference value by manually adjusting the six-DOF extrinsic parameters until the point cloud projections onto the camera images are visually plausible as much as possible (as depicted in Fig. 7). The resulting reference values are also listed in TABLE 1 and serve as the ground-truth for the following evaluations.

Compared to the reference values, the first-stage calibration result has a translation difference about 2 *cm* and a rotation difference about 0.3°. Such differences are further reduced by our second-stage calibration algorithm and lead to a final difference of 0.2 *cm* in translation and 0.06° in rotation. Thus, we conclude our two-staged calibration algorithm can perform accurate calibration on real robots.

Furthermore, the computation time of the whole calibration pipeline is around 90 second, including the scale estimation and the two-stage calibration, which is acceptable for practical in-field operations of robots.

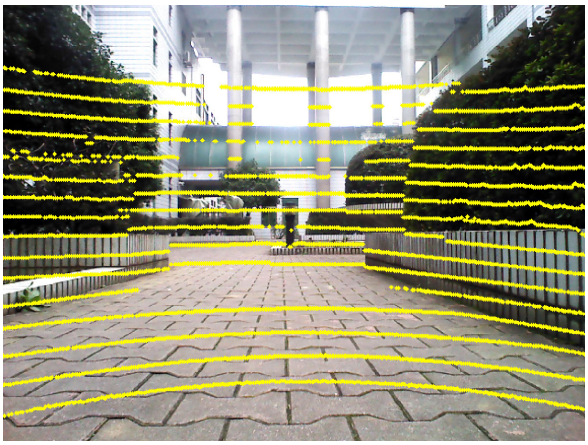
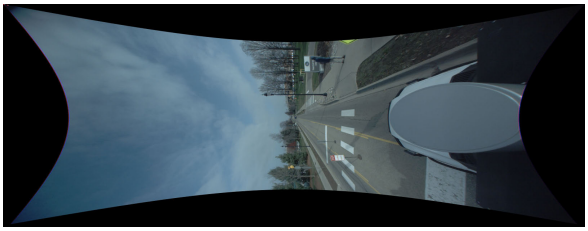
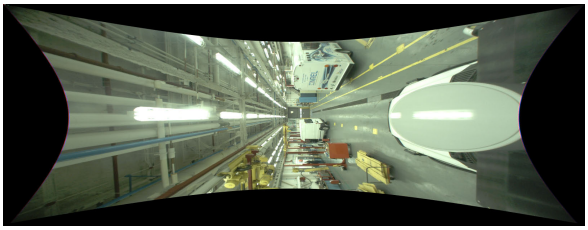


FIGURE 7. Point cloud projections onto the camera images using manually adjusted extrinsic parameters.



(a) outdoor example

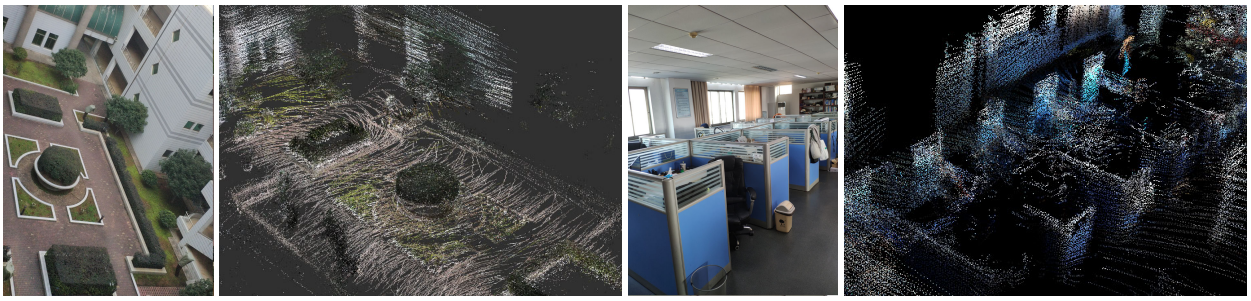


(b) indoor example

FIGURE 8. Ford Campus dataset example (a) outdoor scene (b) indoor scene.

B. EXTRINSIC CALIBRATION WITH FORD CAMPUS DATASET

To further validate our calibration algorithm, we also apply our algorithm/pipeline to the Ford Campus dataset [29], which is a dataset collected using a ground vehicle equipped with a Point Grey Ladybug3 omnidirectional camera system and a Velodyne 3D LiDAR (examples shown in Fig. 8). We estimate the extrinsic parameters between the LiDAR and



(a) (b) (c) (d)

FIGURE 9. The maps generated by registering the point clouds based on our odometry in (a)-(b) a courtyard, (c)-(d) an office. The camera-LiDAR system is placed on NuBot platform.

TABLE 2. Calibration results on Ford Campus dataset.

	Translation (m)			Mean Error (m)
	x	y	z	
Ground Truth	0.4150	0.0129	0.2998	—
Estimated	0.4185	0.0311	0.3021	**0.0095**
Estimated(no occl. filtering)	0.4185	0.0308	0.3075	0.0097
Mean error in translation is the mean value of each axis error.				
	Rotation $(^\circ)$			Mean Error $(^\circ)$
	$roll$	$pitch$	yaw	
Ground Truth	89.9406	-0.1068	-89.7813	—
Estimated	90.1219	-0.1364	-89.7147	**0.0495**
Estimated(no occl. filtering)	90.0434	-0.1914	-90.0473	0.1640
Mean error in rotation is the error of angle parameter of the angle-axis vector.				

TABLE 3. Translational drifts of each estimated trajectory.

Environment	Trial No.	End Point Error (m)			Error (%)
		x	y	z	
Courtyard	01	-1.209	0.856	0.476	2.47
	02	1.300	1.009	-0.628	2.68
	03	-1.097	-0.965	-0.635	2.52
Office	01	-0.1385	-0.2702	0.6388	2.51
	02	0.2001	-0.1178	-0.7207	2.68
	03	0.1639	-0.1295	0.6433	2.39

TABLE 4. Comparison on translational drifts of three methods.

Environment	LOAM [28] (a quarter of the laser points)	ORB-SLAM [1]	Ours
Office	13.3%	3.89%	**2.52%**
Courtyard	14.1%	7.40%	**2.56%**

the forward-looking camera, then compare the result to the provided parameters.

The calibration result is shown in TABLE 2. Compared to the ground-truth, our estimated parameters have a translational error of 0.95 *cm* and a rotational error of 0.0495°.

To study the influence of our occlusion filtering algorithm, we also evaluated the calibration results estimated with the occlusion filtering disabled. As can be seen in TABLE 2, the calibration accuracy deteriorates when the filtering is disabled, especially for the rotation estimation. Such a result suggests that our occlusion detection approach can improve the overall calibration accuracy.

C. ODOMETRY WITH OWN DATASET

In this experiment, we evaluate our odometry approach using datasets collected with our robotic platform.

The dataset contains both indoor and outdoor environments, as shown in Fig. 9a and Fig. 9c. During the experiment, we instruct the robot to strictly follow a set of predefined trajectories, whose geometry are known and manually measured. All the trajectories are loop-closed, and the robot always returns to the starting point, therefore we can easily calculate the accumulated drift of the odometry.

We carried out three trials for each scene, and the translational drifts/errors of each trial are listed in TABLE 3. The average error of the indoor trials is 2.52%, and the average error of the outdoor trials is 2.56%, as shown in the rightmost column of TABLE 4.

TABLE 4 also shows the average translational errors of i) the LiDAR-based approach LOAM [28] and ii) the camera-based approach ORB-SLAM [1] on this dataset for comparison. As can be seen in the TABLE 4, our method outperforms the other two methods which rely on a single sensor modality.

The resulting 3D maps generated by our approach are shown in Fig. 9b and Fig. 9d. In the experiment, all computation is performed in real-time, and the camera-LiDAR extrinsic parameters are determined by our two-stage calibration method as mentioned in the previous section.

It is worth mentioning that our odometry approach is able to handle (relatively) fast motion. During the experiment, our algorithm remains accurate even when the robot moves with relatively high speed (around 0.8 m/s) and making large turns. Take the corner shown in Fig. 10 as an example. The map generated by LOAM (i.e. Fig. 10a) is significantly misaligned during a 90° rotation whereas the result of our approach (i.e. Fig. 10b) is not affected by such motion.

(a) LOAM (b) Ours

FIGURE 10. **(a) The LOAM failed when passing the corner (the point cloud is colored). (b) Our approach has almost no drift in the same corner, which proves that fusing image information improves the odometry robustness.**

TABLE 5. **Comparison on averaging translational error of the KITTI odometry dataset.**

Seq No.	LOAM [28] (a quarter of the laser points)	ORB-SLAM [1]	DEMO [8]	Ours
01	14.53%	×	1.87%	2.30%
03	5.61%	1.45%	0.99%	1.06%
10	4.79%	2.21%	1.14%	1.62%

D. ODOMETRY WITH THE KITTI DATASET

In this experiment, we make further comparisons between our odometry approach and three state-of-the-art approaches using the KITTI odometry dataset [30]. The three methods are i) LiDAR-only odometry LOAM, ii) camera-only odometry ORB-SLAM and iii) a state-of-the-art camera-LiDAR odometry named DEMO [8].

We chose the sequence 01, 03 and 10 for a fair comparison, as these sequences do not contain loops in their path and hence once can ignore the effect of loop-closing. The respective translational error of each odometry method is depicted in TABLE 5. Judging from the result, our approach again outperforms the single modality methods (i.e., LOAM and ORB-SLAM), and can achieve competitive estimation accuracy with respect to the state-of-the-art camera-LiDAR odometry DEMO.

Notice that the ORB-SLAM lost track in the sequence 01, which is not unexpected since sequence 01 is a challenging highway sequence which lacks both structural and textual features. Our approach, on the other hand, is not affected.

VI. CONCLUSION

In this paper, we focus on camera-LiDAR systems and propose a two-stage extrinsic calibration method as well as a hybrid-residual-based odometry approach. To better register the image and the point cloud data, our calibration approach combines a motion-based approach and a mutual-information-based approach, where the motion-based approach (i.e., the first stage) estimates an initial guess of the extrinsic parameters, and the mutual-information-based approach (i.e., the second stage) refines the result to a high accuracy. We also presented a novel occlusion filtering algorithm to remove occluded data points, which is useful for improving calibration accuracy and robustness. We also proposed real-time hybrid-residual-based odometry which exploits both photometric and reprojection image features. The point cloud sparsity problem is resolved with our color-related depth-interpolation. We evaluated our calibration and odometry approach on both our own robotic platform as well as the real-world public dataset. The experimental results suggest that our calibration method can provide accurate extrinsic parameters estimation, and our odometry approach can achieve competitive estimation accuracy and robustness.

REFERENCES

[1] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1255–1262, Oct. 2017.

[2] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 3, pp. 611–625, Mar. 2016.

[3] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast semi-direct monocular visual odometry," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May/Jun. 2014, pp. 15–22.

[4] Y. Chen and G. Medioni, "Object modelling by registration of multiple range images," *Image Vis. Comput.*, vol. 10, no. 3, pp. 145–155, Apr. 1992.

[5] M. Magnusson, "The three-dimensional normal-distributions transform—An efficient representation for registration, surface analysis, and loop detection," Ph.D. dissertation, Örebro Univ., Örebro, Sweden, 2009.

[6] Y.-S. Shin, Y. S. Park, and A. Kim, "Direct visual SLAM using sparse depth for camera-LiDAR system," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 1–8.

[7] H. Andreasson, R. Triebel, and A. Lilienthal, "Vision-based interpolation of 3D laser scans," in *Proc. IEEE Int. Conf. Auto. Robots Agents (ICARA)*, Dec. 2006, pp. 455–460.

[8] J. Zhang, M. Kaess, and S. Singh, "Real-time depth enhanced monocular odometry," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2014, pp. 4973–4980.

[9] K. Huang and C. Stachniss, "Extrinsic multi-sensor calibration for mobile robots using the Gauss–Helmert model," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 1490–1496.

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

[11] Q. Yu, J. Xiao, H. Lu, and Z. Zheng, "Hybrid-residual-based RGBD visual odometry," *IEEE Access*, vol. 6, pp. 28540–28551, 2018.

[12] Y. Park, S. Yun, C. S. Won, K. Cho, K. Um, and S. Sim, "Calibration between color camera and 3D LIDAR instruments with a polygonal planar board," *Sensors*, vol. 14, no. 3, pp. 5333–5353, 2014.

[13] J.-E. Ha, "Extrinsic calibration of a camera and laser range finder using a new calibration structure of a plane with a triangular hole," *Int. J. Control, Autom., Syst.*, vol. 10, no. 6, pp. 1240–1244, 2013.

[14] F. Vasconcelos, J. P. Barreto, and U. Nunes, "A minimal solution for the extrinsic calibration of a camera and a laser-rangefinder," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2097–2107, Nov. 2012.

[15] Y. C. Shiu and S. Ahmad, "Calibration of wrist-mounted robotic sensors by solving homogeneous transform equations of the form AX=XB," *IEEE Trans. Robot. Autom.*, vol. 5, no. 1, pp. 16–29, Feb. 1989.

[16] K. H. Strobl and G. Hirzinger, "Optimal hand-eye calibration," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2006, pp. 4647–4653.

[17] Q. Liao, M. Liu, L. Tai, and H. Ye, "Extrinsic calibration of 3D range finder and camera without auxiliary object or human intervention," 2017, *arXiv:1703.04391*. [Online]. Available: https://arxiv.org/abs/1703.04391

[18] P. Moghadam, M. Bosse, and R. Zlot, "Line-based extrinsic calibration of range and image sensors," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2013, pp. 3685–3691.

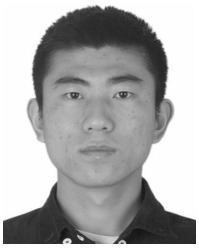
[19] F. Boughorbal, D. L. Page, C. Dumont, and M. A. Abidi, "Registration and integration of multisensor data for photo-realistic scene reconstruction," in *Proc. 28th AIPR Workshop, 3D Vis. Data Explor. Decis. Making*, vol. 3905, 2000, pp. 74–85.

[20] A. Napier, P. Corke, and P. Newman, "Cross-calibration of push-broom 2D LIDARs and cameras in natural scenes," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2013, pp. 3679–3684.

[21] N. Williams, K.-L. Low, C. Hantak, M. Pollefeys, and A. Lastra, "Automatic image alignment for 3D environment modeling," in *Proc. 17th Brazilian Symp. Comput. Graph. Image Process.*, 2004, pp. 388–395.
[22] H. Andreasson and A. J. Lilienthal, "6D scan registration using depth-interpolated local image features," *J. Robot. Auto. Syst.*, vol. 58, no. 2, pp. 157–165, 2010.
[23] B. D. Corte, I. Bogoslavskyi, C. Stachniss, and G. Grisetti, "A general framework for flexible multi-cue photometric point cloud registration," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 1–8.
[24] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*. Evanston, IL, USA: Routledge, 2018.
[25] J. Barzilai and J. M. Borwein, "Two-point step size gradient methods," *IMA J. Numer. Anal.*, vol. 8, no. 1, pp. 141–148, 1988.
[26] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Nov. 2011, pp. 2564–2571.
[27] I. Ashraf, S. Hur, and Y. Park, "An investigation of interpolation techniques to generate 2D intensity image from LiDAR data," *IEEE Access*, vol. 5, pp. 8250–8260, 2017.
[28] J. Zhang and S. Singh, "LOAM: Lidar odometry and mapping in realtime," in *Proc. Robot., Sci. Syst. Conf. (RSS)*, vol. 2, Jul. 2014, p. 9.
[29] G. Pandey, J. R. McBride, and R. M. Eustice, "Ford campus vision and lidar data set," *Int. J. Robot. Res.*, vol. 30, no. 13, pp. 1543–1552, 2011.
[30] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1231–1237, 2013.

CHENGHAO SHI received the B.Eng. degree from the Nanjing University of Aeronautics and Astronautics (NUAA), in 2017. He is currently pursuing the master's degree with the National University of Defense Technology (NUDT). His research interest includes localization and mobile robots.

KAIHONG HUANG received the B.E. degree from the University of Electronic Science and Technology of China, in 2011, the M.E. degree from the National University of Defense Technology, China, in 2013, and the Ph.D. degree from the University of Bonn, Germany, in 2018. His research interests include robotics perception and state estimation, simultaneous localization and mapping, and sensor calibration.



QINGHUA YU received the B.Eng., M.Eng., and Ph.D. degrees from the National University of Defense Technology (NUDT), China, in 2011, 2013, and 2018, respectively. His research interests include robot vision and mobile robots.

JUNHAO XIAO received the B.Eng. degree from the National University of Defense Technology (NUDT), China, in 2007, and the Ph.D. degree from the Department of Informatics, Institute of Technical Aspects of Multimodal Systems (TAMS), University of Hamburg, Germany, in 2013. In 2013, he joined the Department of Automation, NUDT, where he is currently an Assistant Professor. His research interests include mobile robotics, especially on localization, mapping, robot soccer, and robot rescue.

HUIMIN LU received the B.Eng., M.Eng., and Ph.D. degrees from the National University of Defense Technology (NUDT), China, in 2003, 2005, and 2010, respectively. In 2010, he joined the Department of Automation, NUDT, where he is currently an Associate Professor. His research interests include mobile robotics, especially on omnidirectional vision, visual SLAM, robot soccer, and robot rescue.

CHENGGANG XIE received the B.Eng. and M.Eng. degrees from the National University of Defense Technology (NUDT), China, in 1989 and 1994, respectively, and the Ph.D. degree from Central South University, China, in 2008. In 1989, he joined the Department of Automation, NUDT, where he is currently a Senior Engineer. His research interests include mobile robotics, especially on localization, mapping, and robot vision.