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

LiDAR-camera Calibration using Intensity Variance Cost

Ryoichi Ishikawa[1], Shuyi Zhou[1], Yoshihiro Sato[1], Takeshi Oishi[1], Katsushi Ikeuchi[2]

***Abstract*— We propose an extrinsic calibration method for LiDAR-camera fusion systems using variations in intensities projected from camera images to the LiDAR point cloud. As the input, the proposed method uses a sequence of LiDAR data and camera images captured while moving the system. Once the camera motion is calculated, camera images are projected onto the point cloud. The variations in the projected intensities at each point are large in the presence of errors in the estimated motion or calibration parameters. Consequently, the extrinsic parameters are optimized for cost minimization based on the intensity variance. In addition, a suitable geometry is proposed for the calibration and verified using simulations. Our experimental results showed that the proposed method accurately performed calibrations using a camera and a sparse multi-beam LiDAR or one-dimensional LiDAR.**

I. Introduction

Multimodal sensing is essential to realize various applications in robotics and computer vision. Using various types of sensors enables the collection of rich information to understand the surrounding environment. Especially, the fusion of cameras and LiDARs is one of the popular multisensor configurations. The camera collects dense color information, whereas the LiDAR collects sparse but accurate depth information. LiDAR-camera fusion systems are widely utilized in SLAM [1], three-dimensional (3D) scanning [2], object detection [3], and depth completion [4].

Precise extrinsic calibration between sensors is necessary in LiDAR-camera fusion systems for stable performance. Since the optical centers of the sensors are different, the relative pose between them must be known to process the data in the same coordinate system. A point measured by both sensors must be from exactly the same surface point. Various approaches have been proposed for calibrating these sensors using the appearance [5], [6], motion [7], [8], and geometric information [9] from LiDAR and camera data.

However, calibration between the camera and sparse LiDAR or one-dimensional (1D) LiDAR has remained challenging. Sparse LiDAR is commonly used in systems that require real-time performance. Profiler-type systems in which 1D LiDAR scans the sides of the robot while moving forward provide dense and uniform 3D sampling (Fig. 1). Despite these advantages, the data collected using these LiDARs do not provide enough overlap regions to align the surfaces or rich texture information, even when using the reflectance of the laser. Moreover, the movement of mobile platforms is usually restricted, hindering the employment of motion-based approaches for LiDAR systems.

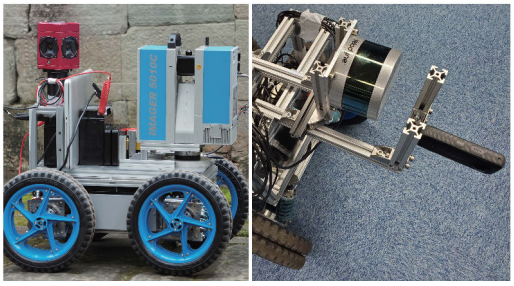

Fig. 1. Our goal is to achieve targetless LiDAR-Camera calibration of sparse LiDAR-based systems like profilers (left) or multibeam LiDARs with a small number of lasers (right) on a moving vehicle with restricted movements.

This paper presents a LiDAR-camera calibration method that can be applied to sparse LiDAR, 1D LiDAR, and motion-restricted systems. The key idea is using the intensity variance of camera images projected onto LiDAR points inspired from direct SLAM techniques [10], [11]. The proposed method assumes that the input is a sequence of LiDAR–camera scans captured while moving the system. When we project the pixel colors of camera images onto LiDAR points, the colors will be inconsistent and blurred if there are errors in the motion or calibration parameters. Therefore, we use the intensity variance as the cost, and the calibration parameters and the motion are simultaneously refined to minimize the cost.

The contributions of this paper are summarized as follows:

1) We propose a targetless LiDAR-camera calibration applicable even to 1D LiDAR using the cost function based on intensity variance.
2) We propose a simultaneous LiDAR-camera calibration and motion refinement method
3) We clarify the relationship between the calibration parameters and the geometry of the target scene.

The proposed method aims to perform offline calibration of a 3D scanning system with a moving platform. The proposed method assumes that there is no significant change in the lighting environment because the input data is measured for a short period of time ($< 40sec$).

II. Related work

A. Appearance-based method

The appearance-based method is a multimodal alignment approach that finds common points between images and 3D

[1]Ryoichi Ishikawa, Shuyi Zhou, Yoshihiro Sato, and Takeshi Oishi are with Institute of Industrial Science, The University of Tokyo, Japan {ishikawa, zhoushuyi495, yoshi, oishi}@cvl.iis.u-tokyo.ac.jp
[2]Katsushi Ikeuchi is with Microsoft, U.S.A. katsuike@microsoft.com

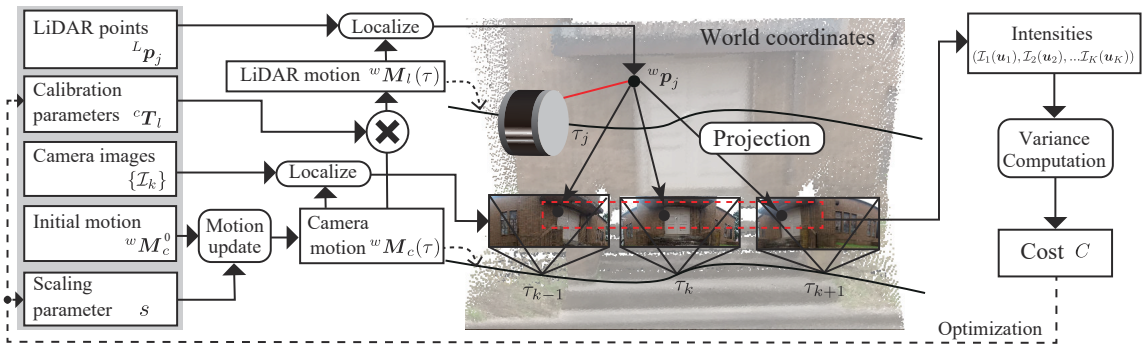

Fig. 2. Overview of cost computation. We first calculate pixel coordinates of LiDAR points projection. Then, we compute intensity variance cost using the sampled intensities of pixels.

point clouds using a target such as a checkerboard [12], circular object [13], or spherical object [14]. Methods of calibrating 1D LiDAR and a camera using targets with edges or right-angled structures have also been proposed [15]–[17]. Nontargeted methods for 2D LiDAR have been proposed as well to find commonalities in multimodal images using mutual information (MI) between (2D) grayscale-(3D) surface normal [18], MI between (2D) grayscale and (3D) reflectance [6], [19], edges [5], [20]–[22], semantic information [23], or gradients [24]. Recently, Neural network-based methods have been developed [25], [26].

B. Motion-based method

The motion-based method measures motion for each sensor and solves their relative positional posture using the differences in motion based on a method known as hand-eye calibration [27], [28]. Calibration methods for various sensor combinations have been proposed, such as the camera–inertial measurement unit (IMU) sensor system using the extended Kalman filter (EKF) method and multiple cameras [29], have been proposed. The motion-based LiDAR–camera calibration estimates LiDAR motion via LiDAR SLAM and camera motion via visual [7] or fusion SLAM [8]. The motion-based method avoids multimodal processing and is less dependent on LiDAR and camera characteristics. Conversely, the motion-based method requires LiDAR to estimate motion by itself, which may cause instability in LiDAR with a sparse scan and may not be applicable to 1D LiDAR.

C. Geometry-based method

The geometry-based method derives 3D geometrical information from camera stereo images taken at multiple sites. The 3D structure is then reconstructed from the images and aligned with a 3D point cloud [9], [30], [31]. A method that computes the extrinsic parameter between the LiDAR and the camera for texturing a dense 3D scan has been developed [30]. Chien *et al.* proposed a calibration method that iterates ego-motion estimation and extrinsic calibration by minimizing the reprojection error [9].

Although the existing geometry-based LiDAR–camera calibration methods are indirect approaches based on feature point tracking, in this work, we propose a direct geometry-based approach using consistency between motion-stereo and LiDAR scans. Instead of using intensity differences as a direct LiDAR–camera SLAM [32], we use the variance of the intensity, which allows the simultaneous correction of calibration and motion. Moreover, the proposed method does not require strong constraints on the target to be captured (target, right-angled structure, etc.), as is the case with other 2D LiDAR calibration methods [15]–[17].

III. Framework

A. Problem formulation

We assume that the inputs are sequential LiDAR points ${}^{\mathrm{L}}\boldsymbol{p}_j \in \mathbb{R}^3 (j = 1, 2, .., n_{\mathrm{p}})$ and camera images $\{\mathcal{I}_k \in \mathbb{R}^{n_{\mathrm{c}} \times W \times H}\} (k = 1, 2, .., n_{\mathrm{f}})$, where n_{p}, n_{f}, and n_{c} are the numbers of LiDAR points, camera images, and color channels, respectively. Generally, n_{c} is 1 or 3 (grayscale or RGB). The LiDAR points and camera images have time records τ_j and τ_k, respectively. ${}^{l}\boldsymbol{p}_j$ is described in the local coordinate system of the LiDAR at time τ_j. The initial values of the extrinsic matrix ${}^{c}\boldsymbol{T}_l \in \mathbb{SE}(3)$ and the camera motion ${}^{w}\boldsymbol{M}_c^0$ are also assumed to be given. The LiDAR motion ${}^{w}\boldsymbol{M}_l$ is expressed as: ${}^{w}\boldsymbol{M}_l = {}^{w}\boldsymbol{M}_c {}^{c}\boldsymbol{T}_l$.

${}^{w}\boldsymbol{M}_c(\tau)$ indicates the camera pose at time τ in the world coordinates. We assume that the camera poses between observed frames are interpolated by parametric curves or other techniques. Due to recent advancements in photogrammetry methods, we can assume that the initial camera motion ${}^{w}\boldsymbol{M}_c^0$ is accurate, except for the scaling information. Therefore, we introduce a single parameter $s \in \mathbb{R}$ to refine the camera motion as follows:

$$ {}^{w}\boldsymbol{M}_c = \begin{pmatrix} \boldsymbol{I}_{3\times3} & s\mathbf{1}_{3\times1} \\ \mathbf{0}_{1\times3} & 1 \end{pmatrix} \cdot {}^{w}\boldsymbol{M}_c^0, \tag{1} $$

where $\boldsymbol{I}$ is an identity matrix.

As shown in Fig. 2, the proposed method estimates the optimal extrinsic parameter ${}^{c}\hat{\boldsymbol{T}}_l$ and the camera motion

parameter $\hat{s}$ by minimizing the intensity variance cost C:

$$ {}^{c}\hat{\boldsymbol{T}}_l, \hat{s} = \underset{{}^{c}\boldsymbol{T}_l, s}{\arg\min}\, C. \tag{2} $$

The remainder of this section describes the procedures for calculating cost C.

B. Intensity variance cost computation

When projecting images onto LiDAR points (Fig. 2), the uv coordinates of each LiDAR point on the image plane must be determined in subpixels. These coordinates are obtained by projecting the LiDAR point onto the image plane.

1) Calculation of pixel coordinates: To obtain the image coordinates on camera images, we first locate the LiDAR points into the world coordinates with the LiDAR motion. The LiDAR point $\boldsymbol{p}_j$ in the world coordinates is

$$ {}^{w}\tilde{\boldsymbol{p}}_j = {}^{w}\boldsymbol{M}_l(\tau_j)\, {}^{l}\tilde{\boldsymbol{p}}_j, \tag{3} $$

where τ_j is the time when $\boldsymbol{p}_j$ is scanned. $\tilde{\boldsymbol{p}} = \left[\boldsymbol{p}^\top, 1\right]^\top$ is the operator to add element 1 to the end of a vector. Then, ${}^{w}\boldsymbol{p}_j$ is projected onto kth camera image by the camera pose at time τ_k, and pixel coordinates $\boldsymbol{u}_{j,k}$ are obtained as follows:

$$ \boldsymbol{u}_{j,k} = \pi({}^{c_k}\boldsymbol{p}_j), \tag{4} $$

$$ {}^{c_k}\tilde{\boldsymbol{p}}_j = {}^{w}\boldsymbol{M}_c(\tau_k)^{-1}\, {}^{w}\tilde{\boldsymbol{p}}_j, \tag{5} $$

where $\pi(\cdot)$ is the projection function derived from the camera intrinsics. The pixel coordinates of all LiDAR points for all camera images are calculated here.

2) Intensity variance computation: We assume $\mathcal{I}(\boldsymbol{u})$ returns the intensity value at pixel $\boldsymbol{u}$ in the sub-pixel. The intensity variance for the LiDAR point $\boldsymbol{p}_j$ in the color channel m is obtained from the n_f intensity values as follows:

$$ c_m(\boldsymbol{p}_j) = \frac{1}{n_\mathrm{f}} \sum_{k=1}^{n_\mathrm{f}} \alpha_k(\boldsymbol{p}_j) \left| \mathcal{I}_{k,m}(\boldsymbol{u}_{j,k}) - \mu_{j,m} \right|^2, \tag{6} $$

$$ \mu_{j,m} = \frac{1}{n_\mathrm{f}} \sum_{k=1}^{n_\mathrm{f}} \mathcal{I}_{k,m}(\boldsymbol{u}_{j,k}), \tag{7} $$

where $\mu_{j,m}$ is the average of the intensity component. The weight function α_k is used to eliminate the influence of geometric or photometric problems for robust estimation (details are provided in Section IV).

Considering the color channels, the entire cost for a point is

$$ c(\boldsymbol{p}_j) = \sum_{m=1}^{n_\mathrm{c}} c_m(\boldsymbol{p}_j). \tag{8} $$

3) Cost computation: The entire cost is the summation of the intensity variances of all LiDAR points. We introduce a weight function $\beta(\boldsymbol{p}_j)$ to improve optimization performance, which is also described in Section IV. Accordingly, the entire cost is as follows:

$$ C = \frac{1}{\sum_{j=1}^{n_\mathrm{p}} \beta(\boldsymbol{p}_j)} \sum_{j=1}^{n_\mathrm{p}} \beta(\boldsymbol{p}_j) c(\boldsymbol{p}_j), \tag{9} $$

In the remainder of the discussion, we omit the index of the color channel m in the notation for clarity.

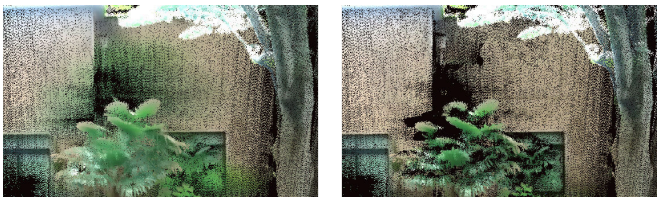

Fig. 3. Average color mapping without (left) and with (right) occlusion handling. The occlusion handling avoids the mapping of the leaf color onto the wall points.

IV. Robust optimization

In occluded areas, $\mathcal{I}(\boldsymbol{u})$ may acquire the intensity of the surface of the occluding object rather than the correct surface behind it, as shown in Fig. 3 (left). Moreover, intensity variations tend to be unreliable in and around the saturated regions.

Therefore, we introduce weighting schemes to reduce these effects for robust optimization. In this section, we define the weight functions α and β used in Eqs. 6 and 9.

A. Occlusion handling

We need to verify if a LiDAR point is occluded when viewed from a certain camera image. For this verification, the proposed method renders a depth map using a mesh model generated from a point cloud and determines if each point is on the mesh or occluded by comparing the depth to each point and the mesh surface.

First, we localize LiDAR points in the camera coordinate system using the initial camera motion and calibration parameters. Next, we generate a mesh for the point cloud [33], [34] and render a depth map from each camera position. Finally, we fill holes by updating pixel values with the smallest value in the neighboring pixels and obtain the depth map $\boldsymbol{D}_k$ for the camera frame k.

Once the depth maps are generated, we compare the depth of the LiDAR point in the camera coordinate system $||{}^{c_k}\boldsymbol{p}_j||_2$ and the depth on the depth map $\boldsymbol{D}_k(\pi({}^{c_k}\boldsymbol{p}_j))$. The weight ${}^{\mathrm{d}}w_{k,j}$ is 0 when the depth difference Δd is larger than z_f and it is 1 when Δd is smaller than z_n and is interpolated between them in other cases:

$$ {}^{\mathrm{d}}w_{k,j} = \begin{cases} 1 & (\Delta d \le z_\mathrm{n}) \\ 1 - \frac{\Delta d - z_\mathrm{n}}{z_\mathrm{f} - z_\mathrm{n}} & (z_\mathrm{n} < \Delta d \le z_\mathrm{f}) \\ 0 & (z_\mathrm{f} < \Delta d), \end{cases} \tag{10} $$

$$ \Delta d = ||{}^{c_k}\boldsymbol{p}_j||_2 - \boldsymbol{D}_k(\pi({}^{c_k}\boldsymbol{p}_j)). \tag{11} $$

Here, z_n and z_f are the heuristic parameters for tolerance determined by the LiDAR depth accuracy. Using occlusion handling, we can eliminate the effects of incorrect color mapping, as shown in Fig. 3 (right). Note that we initially assign ${}^{\mathrm{d}}w_{k,j}$ and fix it during optimization.

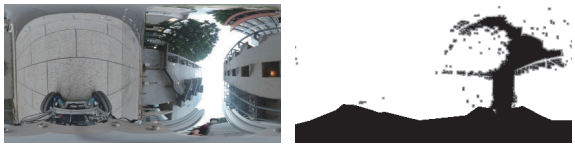

Fig. 4. Example of a weight mask. The mask image (right) is applied to the left image.

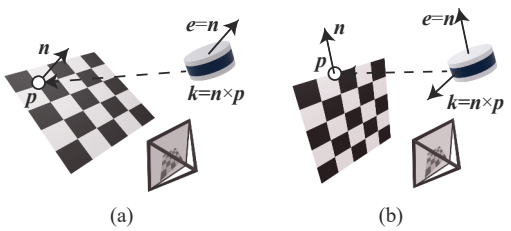

Fig. 5. Direction in which LiDAR points contribute to calibration. (a) Plane (b) Edge.

B. Masking unreliable regions

We use image masks to avoid the use of unreliable areas for variance computation. Fig. 4 shows an example of a mask image. We eliminate the saturated regions in the sky and on highly reflective surfaces. We also eliminate the static region that observes the sensors, including the mobile platform.

Given a mask image for the kth frame $\mathcal{W}_k$, the intensity value $\mathcal{I}_k(\boldsymbol{u})$ in subpixel is represented as

$$\hat{\mathcal{I}}_k(\boldsymbol{u}) = \frac{(\mathcal{I}_k \odot \mathcal{W}_k)(\boldsymbol{u})}{\mathcal{W}_k(\boldsymbol{u})}, \tag{12}$$

where $\mathcal{I} \odot \mathcal{W}$ is the image to take the Hadamard product of the image $\mathcal{I}$ and the mask $\mathcal{W}$. $\mathcal{W}_k(\boldsymbol{u})$ returns the mask value at pixel $\boldsymbol{u}$ in the sub-pixel.

C. Optimization

We incorporate the weight ${}^{\mathrm{d}}w_{k,j}$ and the mask $\mathcal{W}_k$ into the cost calculation in Eq. 6:

$$\alpha_k(\boldsymbol{p}_j) = {}^{\mathrm{d}}w_{k,j}\mathcal{W}_k(\boldsymbol{u}_{j,k}). \tag{13}$$

Since the points close to the sensor are more informative for translation estimation, we set large weights to the close points for translation optimization. Accordingly, we alternately optimize translation and rotation and assign different weights ${}^{\mathrm{r}}w$ to them. Moreover, a bright area makes a large amount of intensity change prone to noise, and the weight takes the inverse of intensity average μ_m (Eq. 7).

Consequently the weight β in Eq. 9 is defined as:

$$\beta_(\boldsymbol{p}_j) = \frac{1}{{}^{\mathrm{r}}w(\boldsymbol{p}_j)\sqrt{\sum_{m=1}^{n_c}\mu_m(\boldsymbol{p}_j)^2 + \epsilon}}, \tag{14}$$

$${}^{\mathrm{r}}w(\boldsymbol{p}_j) = \begin{cases} \sqrt{||{}^{l}\boldsymbol{p}_j||_2} & \text{(translation optimization)} \\ 1 & \text{(others)} \end{cases}, \tag{15}$$

where ϵ is a constant value for avoiding zero division. Finally, the parameters are obtained by alternatively optimizing motion, rotation, and movement in this order. The motion error tends to be large and affects extrinsic parameter optimization, whereas the motion scale is easy to converge.

V. Validation of Cost Function

In this section, we discuss the validity of the cost function and how the target scene affects the estimation of the parameters.

A. Relation between surface geometry and calibration parameters

The intensity variance cost becomes large when a LiDAR point is away from the 3D surface in the camera coordinate system. That is, if modifying a calibration parameter causes moving point $\boldsymbol{p}$ in Fig. 5 in the surface normal direction $\boldsymbol{n}$, $\boldsymbol{p}$ contributes to the calibration of that parameter. First, consider the case of translation parameter displacement. If the LiDAR moves by $\boldsymbol{e}$, as shown in Fig. 5 (a), $\boldsymbol{p}$ also moves by $\boldsymbol{e}$. Therefore, $\boldsymbol{p}$ contributes to the translation parameters in the normal direction $\boldsymbol{n}$. Next, consider the case of rotation. Considering that the LiDAR rotates at a slight angle $\Delta\theta$ around the axis $\boldsymbol{k}$, the displacement of the point $\boldsymbol{p}$ is $\Delta\theta\,(\boldsymbol{k} \times \boldsymbol{p})$, and the displacement from the object surface is $\Delta\theta\boldsymbol{n}\,(\boldsymbol{k} \times \boldsymbol{p})$. Therefore, the point $\boldsymbol{p}$ contributes to the rotation parameter in the axial direction $\boldsymbol{k} = (\boldsymbol{n} \times \boldsymbol{p})$. This contribution increases as the incident angle to the surface becomes shallower.

The occlusion edge also contributes to the calibration. The edge is the area where the local surface normals are perpendicular to the surface edge, as shown in Fig. 5 (b). An inconsistency appears only when the inner product of $\boldsymbol{p}$ displacement and $\boldsymbol{n}$ is positive.

B. Evaluation condition

We simulated a fusion system of a multibeam LiDAR with 16 laser units and a fisheye camera. The optical center of two sensors was at the same position, and the rotation axis of the LiDAR was set parallel to the x-axis of the camera. The target objects were a plane, corner, and sphere, as shown in Fig. 6. Only the sphere case has an occlusion edge. The sensor was moved in the direction of the red arrow in a constant velocity.

C. Evaluation

Fig. 7 shows the graphs of the errors and costs of each calibration parameter for the plane, corner, and sphere, respectively. As discussed in Section V-A, the plane mainly contributed to the x-axis rotation and z-axis translation (Top of Fig. 7). The minima in the y-axis rotation of the plane object scene also appeared slightly. The results obtained using the corner showed that the slope of the peak was larger for all parameters, as shown in the center of Fig. 7. For the

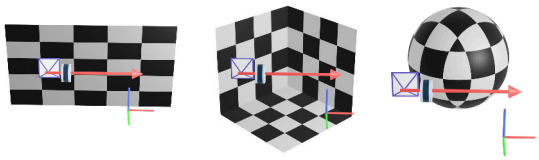

Fig. 6. Shapes used for simulation (left to right: plane, corner, and sphere) and camera and LiDAR movement. The x, y ,z axes correspond to red, blue, and green, respectively.

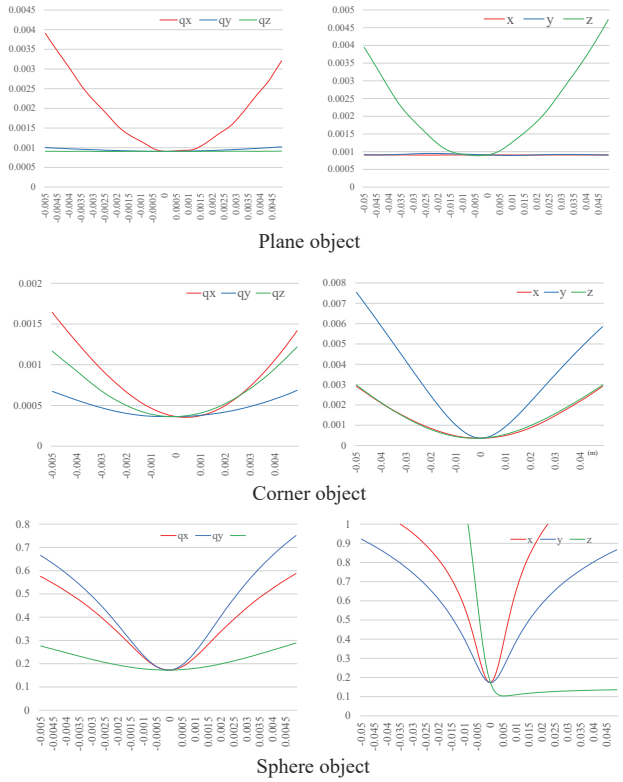

Fig. 7. Parameter error vs cost using a plane, corner, and sphere object. Left: Rotation (quaternion), Right: Translation (m).

sphere, sharp peaks due to the occlusion edges are shown in the bottom of Fig. 7.

VI. Experiment of Sparse LiDAR System

A. Sensor system and data acquisition

We evaluate our method with sparse multibeam LiDAR Velodyne VLP16 [35] and the panoramic camera insta360 One X [36] . We mounted the sensor system on a rover with the rotation axis of the LiDAR oriented to coincide with the travel direction, as shown in Fig. 1. The rover slowly moved straight while the LiDAR measured scenes on its sides.

Fig. 8 shows the target scenes for the experiments. We synchronized the camera and LiDAR by timestamps. We uniformly subsampled 30 camera frames to reduce calculation costs, and we also used subsampled LiDAR points at equal

Fig. 8. Scenes for sparse LiDAR-camera calibration

intervals and filtered out weak intensity points as unreliable points in advance.

B. Implementation

We manually set the initial extrinsic parameters, and the initial rotation and translation error in SCENE 1 and 2 was above (0.024, 0.063m) and it was above (0.073, 0.048m) in SCENES 3 and 4. We estimated an initial camera motion by using a feature-based camera tracking method and the initial absolute scale with LiDAR points corresponding to landmarks and the initial extrinsic parameters.

The optimization method was the line-search implemented in dlib [37], and the method approximated the direction of the gradient using a finite-difference method. The iteration times of the alternative parameter optimization was set to 6. We empirically set the depth values for occlusion detection (Section IV-A) as $z_n = 0.1(\mathrm{m})$, $z_f = 0.2(\mathrm{m})$ and $\epsilon = 1/255.0$.

C. Evaluation

We computed a reliable reference for evaluation using the high-resolution and colored scans taken by the Z+F Imager 5010C [38]. We aligned the sparse LiDAR points with the reference scans using an ICP algorithm. We also estimated the pose between the reference scan and the camera by using manually selected corresponding points. The high-resolution and colored scans help in identifying stable corresponding points for 2D-3D alignment.

The error metric for rotation estimation is the quaternion difference between the reference rotation $\boldsymbol{q}_{ref}$ and the estimated rotation $\boldsymbol{q}_{est}$: $||\boldsymbol{q}_{ref} - \boldsymbol{q}_{est}||_2$. The error metric for translation estimation is the L2 distance of the reference translation $\boldsymbol{t}_{ref}$ and the estimated translation $\boldsymbol{t}_{est}$: $||\boldsymbol{t}_{ref} - \boldsymbol{t}_{est}||_2$ in meters. For comparison, we used MI-based [6], an edge-based [5], and a motion-based [8] methods. Note that we also tried to apply [22], but it did not work for our sparse LiDAR dataset.

D. Results

Table I shows the error obtained using each method from the reference. The motion-based method [8] failed in any case because of the lack of rotational movements. The MI method [6] provided low accuracy due to the sparse ray

TABLE I

TRANSLATION AND ROTATION IN THE QUATERNION ERROR OF SPARSE LIDAR-CAMERA CALIBRATION

	SCENE 1		SCENE 2		SCENE 3		SCENE 4	
	Tran.(m)	Rot.	Tran.(m)	Rot.	Tran.(m)	Rot.	Tran.(m)	Rot.
Ours	**0.047**	0.0085	**0.042**	**0.0041**	**0.041**	**0.0024**	**0.029**	**0.0049**
Edge [5]	0.048	**0.0023**	0.153	0.2785	0.043	0.0511	0.043	0.0147
MI [6]	0.060	0.1216	0.130	0.0263	0.229	0.0285	0.117	0.0983
Motion-Based [8]	128.558	0.8614	10.877	1.0807	16.281	0.2616	12.817	0.3255

Fig. 9. Average color texturing of the building in SCENE 4 before (left) and after (right) optimization.

SCENE 5 SCENE 6 SCENE 7

Fig. 10. Scenes of profiler-camera calibration experiments

sampling and the low precision of the laser reflectance. The intensive outdoor sunlight also interfered with the laser reflection. The edge-based method [6] relatively worked well, but its performance was limited because the edges were not distinctive because of the small number of laser lines. The performance of the edge-based method also depends highly on the scene. The method did not work well in SCENE 2, where tree branches generated many unstable jump edges.

On the other hand, our method offered high accuracy, especially in rotation estimation. In addition, it provided high accuracy in translation in the scenes (e.g. in the SCENE 4), where the location was surrounded by static objects and had shape features. Fig. 9 shows the aligned LiDAR scans with texturing from multiple views before and after optimization in SCENE 4. After the optimization, we can see that the RGB of the camera image and the 3D scan data were aligned.

VII. EXPERIMENT OF PROFILER SYSTEM

A. Sensor system and data acquisition

We constructed a profiler (1D LiDAR) scanning system comprising Z+F imager 5010C and Ladybug 3 [39], and the rover, as shown in Fig. 1 left. Fig. 10 shows the target scenes. Capturing started simultaneously using a trigger from the imager 5010C to Ladybug 3 for temporal synchronization. For each scene, we used a 600-frame sequence with uniformly subsampled 30 frames for color variance computation.

TABLE II

ROTATION AND TRANSLATION CALIBRATION ERRORS OF PROFILER-CAMERA CALIBRATION

	Rot. error ($\times 10^{-2}$)	Tran. error ($\times 10^{-2}$ m)
SCENE 5	0.0049±0.0015	0.0617±0.0154
SCENE 6	0.0083±0.0015	0.0303±0.0063
SCENE 7	0.0121±0.0105	0.0359±0.0124

The reference was computed from the manual correspondence selection between the panorama 3D data and image and the encoder angle given by the imager 5010C after fixing the sensor body rotation. We set the initial extrinsic parameters by adding random noises to the reference in rotation with magnitude of 0.02 and a translation error of 0.05 m. We performed the experiment five times.

B. Results

Table II shows the calibration results. Our method removed the rotational calibration error well. Comparing SCENE 5 and 6, we found that the rotation estimation worked well in SCENE 5, whereas translation error is small in SCENE 6. This may be because edges on structures in SCENE 5 contributed to the rotation estimation whereas, in SCENE6, the wall close to the sensors may have contributed to the translation estimation. In SCENE 7, while various objects contribute to the accuracy, the featureless walls and ceiling may be the cause of calibration failure.

VIII. CONCLUSION

In this paper, we proposed an extrinsic calibration method for a sparse LiDAR-camera system, including a profiler-camera system, on a mobile platform using the intensity variance cost. We introduced masking and occlusion handling for robust estimation and discussed the shapes contributing to the calibration. The experimental results showed that the proposed method accurately calibrates in real scenes. For future work, we will enhance the robustness of the method by considering the suitable directions according to the observed structure or combining learning-based approaches to eliminate unreliable regions.

ACKNOWLEDGMENT

This work was partly supported by the social cooperation program "Technology for IoT sensing and analysis," sponsored by UTokyo and Air Water.

REFERENCES

[1] J. Zhang and S. Singh, "Visual-lidar odometry and mapping: Low-drift, robust, and fast," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 2174–2181.

[2] R. Ishikawa, M. Roxas, Y. Sato, T. Oishi, T. Masuda, and K. Ikeuchi, "A 3d reconstruction with high density and accuracy using laser profiler and camera fusion system on a rover," in *3D Vision (3DV), 2016 Fourth International Conference on*. IEEE, 2016, pp. 620–628.

[3] 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.

[4] A. Hirata, R. Ishikawa, M. Roxas, and T. Oishi, "Real-time dense depth estimation using semantically-guided lidar data propagation and motion stereo," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3806–3811, 2019.

[5] J. Levinson and S. Thrun, "Automatic online calibration of cameras and lasers." in *Robotics: Science and Systems*, vol. 2, 2013.

[6] 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.

[7] 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.

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

[9] H.-J. Chien, R. Klette, N. Schneider, and U. Franke, "Visual odometry driven online calibration for monocular lidar-camera systems," in *Pattern Recognition (ICPR), 2016 23rd International Conference on*. IEEE, 2016, pp. 2848–2853.

[10] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "DTAM: Dense tracking and mapping in real-time," in *International Conference on Conputer Vision*, 2011.

[11] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE transactions on pattern analysis and machine intelligence*, vol. 4, 2017.

[12] Q. Zhang and R. Pless, "Extrinsic calibration of a camera and laser range finder (improves camera calibration)," in *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, vol. 3. IEEE, 2004, pp. 2301–2306.

[13] V. Fremont, P. Bonnifait, *et al.*, "Extrinsic calibration between a multi-layer lidar and a camera," in *Multisensor Fusion and Integration for Intelligent Systems, 2008. MFI 2008. IEEE International Conference on*. IEEE, 2008, pp. 214–219.

[14] M. Pereira, V. Santos, and P. Dias, "Automatic calibration of multiple lidar sensors using a moving sphere as target," in *Robot 2015: Second Iberian Robotics Conference*. Springer, 2016, pp. 477–489.

[15] S. Sim, J. Sock, and K. Kwak, "Indirect correspondence-based robust extrinsic calibration of lidar and camera," *Sensors*, vol. 16, no. 6, p. 933, 2016.

[16] Z. Hu, Y. Li, N. Li, and B. Zhao, "Extrinsic calibration of 2-d laser rangefinder and camera from single shot based on minimal solution," *IEEE Transactions on Instrumentation and Measurement*, vol. 65, no. 4, pp. 915–929, 2016.

[17] Y. Bok, D.-G. Choi, and I. S. Kweon, "Extrinsic calibration of a camera and a 2d laser without overlap," *Robotics and Autonomous Systems*, vol. 78, pp. 17–28, 2016.

[18] Z. Taylor and J. Nieto, "A mutual information approach to automatic calibration of camera and lidar in natural environments," in *Australian Conference on Robotics and Automation*, 2012, pp. 3–5.

[19] K. Irie, M. Sugiyama, and M. Tomono, "Target-less camera-lidar extrinsic calibration using a bagged dependence estimator," in *Automation Science and Engineering (CASE), 2016 IEEE International Conference on*. IEEE, 2016, pp. 1340–1347.

[20] T. Cui, S. Ji, J. Shan, J. Gong, and K. Liu, "Line-based registration of panoramic images and lidar point clouds for mobile mapping," *Sensors*, vol. 17, no. 1, p. 70, 2016.

[21] M. Á. Muñoz-Bañón, F. A. Candelas, and F. Torres, "Targetless camera-lidar calibration in unstructured environments," *IEEE Access*, vol. 8, pp. 143 692–143 705, 2020.

[22] C. Yuan, X. Liu, X. Hong, and F. Zhang, "Pixel-level extrinsic self calibration of high resolution lidar and camera in targetless environments," 2021. [Online]. Available: https://arxiv.org/abs/2103.01627

[23] Y. Zhu, C. Li, and Y. Zhang, "Online camera-lidar calibration with sensor semantic information," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 4970–4976.

[24] Z. Taylor, J. Nieto, and D. Johnson, "Multi-Modal Sensor Calibration Using a Gradient Orientation Measure," *Journal of Field Robotics*, vol. 32, no. 5, pp. 675–695, 2015.

[25] 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.

[26] G. Zhao, J. Hu, S. You, and C.-C. J. Kuo, "Calibdnn: multimodal sensor calibration for perception using deep neural networks," in *Signal Processing, Sensor/Information Fusion, and Target Recognition XXX*, vol. 11756. SPIE, 2021, pp. 324–335.

[27] Y. C. Shiu and S. Ahmad, "Calibration of wrist-mounted robotic sensors by solving homogeneous transform equations of the form AX= XB," *ieee Transactions on Robotics and Automation*, vol. 5, no. 1, pp. 16–29, 1989.

[28] I. Fassi and G. Legnani, "Hand to sensor calibration: A geometrical interpretation of the matrix equation AX= XB," *Journal of Field Robotics*, vol. 22, no. 9, pp. 497–506, 2005.

[29] L. Heng, B. Li, and M. Pollefeys, "Camodocal: Automatic intrinsic and extrinsic calibration of a rig with multiple generic cameras and odometry," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE, 2013, pp. 1793–1800.

[30] A. Banno and K. Ikeuchi, "Omnidirectional texturing based on robust 3d registration through euclidean reconstruction from two spherical images," *Computer Vision and Image Understanding*, vol. 114, no. 4, pp. 491–499, 2010.

[31] D. P. Paudel, C. Demonceaux, A. Habed, and P. Vasseur, "2d–3d synchronous/asynchronous camera fusion for visual odometry," *Autonomous Robots*, pp. 1–15, 2019.

[32] Y.-S. Shin, Y. S. Park, and A. Kim, "Direct visual slam using sparse depth for camera-lidar system," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1–8.

[33] Z. C. Marton, R. B. Rusu, and M. Beetz, "On Fast Surface Reconstruction Methods for Large and Noisy Datasets," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Kobe, Japan, May 12-17 2009.

[34] R. Rusu and S. Cousins, "3d is here: Point cloud library (pcl)," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, May 2011, pp. 1 –4.

[35] (2023) Velodyne LiDAR Website. [Online]. Available: https://velodynelidar.com/

[36] (2023) Insta360 Website. [Online]. Available: https://www.insta360.com/

[37] D. E. King, "Dlib-ml: A machine learning toolkit," *Journal of Machine Learning Research*, vol. 10, pp. 1755–1758, 2009.

[38] (2023) Zoller + Fröhlich Website. [Online]. Available: https://www.zofre.de/en/

[39] (2023) Flir Website. [Online]. Available: https://www.flir.com/