Intensity Image-Based LiDAR Fiducial Marker System

Yibo Liu, Hunter Schofield, and Jinjun Shan, Senior Member, IEEE

Abstract—The fiducial marker system for LiDAR is crucial for the robotic application but it is still rare to date. In this letter, an Intensity Image-based LiDAR Fiducial Marker (IILFM) system is developed. This system only requires an unstructured point cloud with intensity as the input and it has no restriction on marker placement and shape. A marker detection method that locates the predefined 3D fiducials in the point cloud through the intensity image is introduced. Then, an approach that utilizes the detected 3D fiducials to estimate the LiDAR 6-DOF pose that describes the transmission from the world coordinate system to the LiDAR coordinate system is developed. Moreover, all these processes run in real-time (approx 40 Hz on Livox Mid-40 and approx 143 Hz on VLP-16). Qualitative and quantitative experiments are conducted to demonstrate that the proposed system has similar convenience and accuracy as the conventional visual fiducial marker system.

Index Terms—Computer vision for automation, range sensing, object detection, segmentation and categorization.

I. INTRODUCTION

CAMERAS and LiDARs are two types of vital sensors found in autonomous driving and robotic applications [1], [2]. Research on the Visual Fiducial Marker (VFM) system, the fiducial marker system developed for the camera, has a long history and lavish experience and research achievements have been accumulated [3]–[7]. The VFM provides the environment with controllable artificial features such that the feature extraction and matching become simpler and more reliable. The VFM systems have been applied in Augmented Reality (AR) [8], human-robot interaction [3], [4], multi-sensor calibration [9] and Simultaneous Localization and Mapping (SLAM) [10].

Development of a LiDAR fiducial marker system also plays a key role in LiDAR applications since feature extraction and matching are fundamental requirements in LiDAR applications [9], [11], [12]. In contrast with the VFM system, the fiducial marker system for LiDAR is rare. To the best of our knowledge, up to date the only fiducial marker system for LiDAR that has similar functionality as the VFM is LiDARTag [13]. However, LiDARTag has a spatial restriction on marker placement on account of the fact that it employs conventional single-linkage clustering algorithm to find the clustering of the marker in the point cloud [13]. Consequently, to satisfy the spatial restriction on marker placement, it is required to add an extra 3D object to the environment when adding a LiDARTag. This negatively affects the spatial environment.

Reviewing the VFM systems, it is found that there are no restrictions on marker placement [5]–[7]. Namely, the marker can be placed anywhere without affecting the environment in terms of space. According to [13], the abandonment of this free-placement virtue is owed to the gap between the structured image and unstructured point cloud. Nevertheless, this gap is not insurmountable. In particular, there is a notable hot trend in LiDAR-based 3D object detection/segmentation [1], [14]: Neural Networks that are originally developed for 2D object detection/segmentation can be utilized to detect/segment the objects in the range/intensity image(s) of the 3D LiDAR point cloud. This indicates that the range/intensity image(s) generated from the LiDAR point cloud can be a pathway to transfer the research accomplishments on the 2D image to the 3D point cloud. Following this inspiration and also considering that the black-and-white marker is explicitly visible in the point cloud rendered by the intensity, we propose an Intensity Image-based LiDAR Fiducial Marker (IILFM) system in this letter. The ample flexibility of the IILFM system is illustrated in Fig. 1. Fig. 2 shows a comprehensive comparison between the proposed system and the LiDARTag system [13] from convenience of usage, extensibility, and user-friendly design perspectives. The contributions of this letter are:

- The development of a novel fiducial marker system for the LiDAR, the IILFM system. Unlike LiDARTag [13] which requires extra 3D objects to be added to the environment, the usage of the IILFM is as convenient and easy as the VFM systems [5], [6]. Namely, the users can produce the marker by printing the VFM on regular Letter size letter with a regular printer and then place the marker anywhere they like.
- The proposal of a novel marker detection method to detect the 3D fiducials through the intensity image. Thanks to this, the VFM systems proposed in the past, present, and even future can be easily embedded into the IILFM system. This is a superiority of the proposed system over LiDARTag [13].
Fig. 1. The proposed system has ample flexibility. Different visual fiducial marker systems (AprilTag [5], ArUco [6] and CCTag [7], etc.) can be easily embedded. There is no restriction on marker placement. The system is applicable for both solid-state LiDAR (a, b) and mechanical LiDAR (c). An AR demo using the proposed system is shown in (d): the teapot point cloud is transmitted to the location of the marker in the LiDAR’s point cloud based on the pose provided by the IILFM system.

- The introduction of a pose estimation approach for the LiDAR via the proposed IILFM, which has similar accuracy as the VFM-based pose estimation for the camera. In addition, dissimilar to the VFM system, the proposed pose estimation is free of the rotation ambiguity problem [15], [16].
- The release of the open-source implementation based on ROS and C++. Please refer to https://github.com/York-SDCNLab/IILFM. The proposed system adopts many user-friendly designs, and thus, users who are familiar with VFM systems can quickly get started with the proposed system.

The letter is organized as follows. Section II introduces the related work. Section III presents the pipeline for detecting 3D fiducials in the point cloud through the intensity image. Section IV illustrates the method to estimate the LiDAR’s pose with respect to the world coordinate system. Section V provides the qualitative and quantitative experimental evaluations regarding marker detection and pose estimation. Section VI gives the conclusion and future work.

II. RELATED WORK

The previous research on the VFM system mainly focuses on the following four aspects:

1) A higher detection rate. The line segmentation methods are constantly being upgraded along with the VFM systems [3], [4]. Apart from line segmentation, non-square shape detection methods, such as ellipse detection algorithms [7], [17], [18], have also sprung up over the years and become beneficial accumulations. The objective of these improvements is to boost the detection of line segments, candidate quads [5], [6], or circles [7], [17], [18], thus the marker detection rate under varying ambient light is improved.

2) A lower false positive rate. The coding/decoding systems of the latest VFM systems [5], [6] are upgraded with regard to the old generations [3], [4], such that the identification of a candidate marker is more reliable and the false positive cases are reduced when decoding.

3) A lower computational time. Again, through the amelioration of methods for graphic segmentation and algorithms adopted in coding/decoding systems, the VFM systems are accelerated. For instance, the speed of the third generation of AprilTag [5] is almost 5 times faster than that of the second generation [3].

4) Resolving the rotation ambiguity problem. The rotational ambiguity means a planar marker could project onto the same pixels from two different poses when the perspective effect is weak [15]. Much research [16], [19] has been conducted to resolve the dual solutions problem. Unlike the first three aspects, the last aspect is a remaining fundamental challenge.

The VFM system is embedded in the proposed system, such that the achievements mentioned in aspects (1)–(3) are inherited in the proposed one. Furthermore, as shown in Section IV, the proposed system is free from the problem introduced in aspect (4).

Although LiDARTag [13] also borrows the dictionary-based coding idea from the AprilTag system [5], most of the accumulations mentioned above are not preserved since LiDARTag follows the conventional approaches to process the 3D point cloud, which brings forward two inconveniences. Firstly, to utilize the single-linkage clustering algorithm, it is needed to guarantee that the representation of the marker in the point cloud meets a spatial requirement: the marker must have \( t \sqrt{2}/4 \) clearance around it [13] where \( t \) represents the marker’s size. If the marker is attached to a wall or a box, it is required to make sure \( \tau > t \sqrt{2}/4 \) where \( \tau \) is the thickness of the marker’s 3D object. Therefore, consider the case that 10 LiDARTags are placed in an indoor environment. Such a situation will result in a room crowded with tripods and large markers. This apparently wrecks the environment. The second inconvenience is that the algorithms to process the point cloud and point clustering are packaged in the implementation of LiDARTag which assumes that the marker is square and the pattern belongs to AprilTag 3 [5]. Suppose that it is desired to adopt a non-square marker system, such as CCTag [7], then it is required to reprogram the LiDARTag system.

III. MARKER DETECTION

A. Generation of the Intensity Image

The generation of an intensity image from a given unstructured point cloud can be summarized as transferring all the 3D points in the point cloud onto a 2D image plane by spherical
projection and rendering the corresponding pixels with intensity values.

Fig. 3 shows the coordinate systems and notations. \( p = [x_l, y_l, z_l]^T \) is an observed point in the 3D point cloud, with \( [x_l, y_l, z_l]^T \) denoting its Cartesian coordinates w.r.t. the LiDAR coordinate system \( \{L\} \) and \( i \) being the intensity. \( u \) is the projection of \( p \) onto the image plane, which has coordinates \( [u, v]^T \) w.r.t. the image coordinate system \( \{I\} \). Following [2], the Cartesian coordinates of \( p \) are first transformed to spherical coordinates \( [\theta, \phi, r]^T \):

\[
\begin{align*}
\theta &= \arctan\left( \frac{y_l}{x_l} \right) \\
\phi &= \arctan\left( \frac{z_l}{\sqrt{x_l^2 + y_l^2}} \right) \\
r &= \sqrt{x_l^2 + y_l^2 + z_l^2}
\end{align*}
\]

where \( \theta \) and \( \phi \) denote the azimuth and inclination, respectively. \( r \) is the range from \( p \) to the origin of \( \{L\} \).

Then, the image coordinates \( [u, v]^T \) of \( u \) are given by:

\[
u = \left\lfloor \frac{\theta}{\Theta_\alpha} \right\rfloor + u_\alpha, \quad v = \left\lfloor \frac{\phi}{\Theta_i} \right\rfloor + v_\Theta
\]

where \( \left\lfloor \cdot \right\rfloor \) represents rounding a value to the nearest integer. This rounding can cause information loss if the pixels are non-integer values at the beginning; however, this approximation is not fatal as can be seen in Section V. \( \Theta_\alpha \) and \( \Theta_i \) are the angular resolutions in \( u \) (azimuth) and \( v \) (inclination) directions, respectively, \( u_\alpha \) and \( v_\Theta \) are the offsets.

Assume it is desired that the point with zero-azimuth and zero-inclination to be projected to the center of the image, then the offsets will be: \( u_\alpha = I_w/2 \) and \( v_\Theta = I_h/2 \), where \( I_w \) and \( I_h \) being the image width and height which are determined by the maximum angular width \( P_w \) and height \( P_h \) of the point cloud \( I_w = \left\lceil \frac{P_w}{\Theta_\alpha} \right\rceil \), \( I_h = \left\lceil \frac{P_h}{\Theta_i} \right\rceil \). The pixel \( [u, v]^T \) is then rendered by a specific color corresponding to the intensity value \( i \). Refer to [20], [21] to see how the correspondence between color and intensity value is generated. For each pixel, the range information \( r \) is also saved for the sake of the following pose estimation. Thereafter, we step through the point cloud and repeat the above process. The pixels that are not mapped to any points will remain unobserved and are rendered by a unique predefined value. Note that if these pixels are visited later on in the marker detection process, they will not return any 3D points because they represent unobserved regions.

**B. Selections of the Angular Resolutions**

The selections of \( \Theta_\alpha \) and \( \Theta_i \) are crucial as they affect the quality of the intensity image directly. Yet, the resolution of this problem is quite simple. Suppose that the horizontal angular
resolution and vertical angular resolution given by the user manual of the employed LiDAR are $\Theta_h$ and $\Theta_v$. We should set $\Theta_a = \Theta_h$ and $\Theta_i = \Theta_v$.

Fig. 4. The intensity images generated under different angular resolution settings. The unobserved regions are rendered light green. (a), (b), and (c) are intensity images generated from the same point cloud given by one LiDAR scan of a Velodyne ULTRA Puck (mechanical LiDAR, $\Theta_h = 0.4^\circ$ and $\Theta_v = 0.33^\circ$). The LiDAR scan is extracted from the dataset provided by [13]. (d), (e), and (f) are intensity images generated from the same point cloud which is the integration of multiple LiDAR scans of a Livox Mid-40 (solid-state LiDAR, $\Theta_h = \Theta_i = 0.05^\circ$). (a), (b), and (c) are cropped to display. (a) $\Theta_a = 0.05^\circ$ and $\Theta_i = 0.05^\circ$. Raw image size = 7290 x 800. (b) $\Theta_a = 0.4^\circ$ and $\Theta_i = 0.33^\circ$. Raw image size = 900 x 121. (c) $\Theta_a = 1.6^\circ$ and $\Theta_i = 1.0^\circ$. Raw image size = 600 x 40. (d) $\Theta_a = \Theta_i = 0.025^\circ$. Raw image size = 1538 x 1178. (e) $\Theta_a = \Theta_i = 0.05^\circ$. Raw image size = 771 x 591. (f) $\Theta_a = \Theta_i = 0.5^\circ$. Raw image size = 81 x 62.

Fig. 5. Sampling patterns of the mechanical LiDAR and the solid-state LiDAR. Sampling points are represented by red scatters. (a) General sampling pattern of a mechanical LiDAR expressed in the azimuth/inclination coordinate system. This is a general schematic that does not correspond to any LiDAR model. (b) Sampling pattern of the Livox Mid-40 LiDAR after one-second integration. Note that the sampling patterns vary when it comes to different solid-state LiDAR models. But generally, the unscanned regions within the valid Field of View (FoV) appear as spots.

that without them the marker detection will fail. A detailed explanation of the necessity of the preprocessing is provided in [23].

C. 3D Feature Points Estimation

The processed image introduced in the previous section is then inputted into the embedded VFM system. Thereafter, the VFM system provides the detection information. In this section, we introduce how to project the 2D feature points given by the detection information back to the 3D space, such that they become 3D feature points expressed in the LiDAR coordinate system. As for the square markers, the feature points refer to the four vertices of the quad. While some of the VFM systems adopt the non-square design, the proposed method has no restriction on the marker shape.

As mentioned in Section III-A, the range information is stored for each pixel in the intensity image as well. Thus, a 2D pixel with range information $(u' = [u, v, r]^T)$ can be projected back to the 3D Cartesian coordinate system by solving the inverse of $(1)-(2)$ to find the corresponding $[x_i, y_i, z_i]^T$. Yet in the real world, it cannot be guaranteed that the feature points of the marker are exactly scanned by the LiDAR. Namely, the detected 2D features in Fig. 6(c) could correspond to the unobserved regions in the raw intensity image. As a matter of fact, this occurs frequently in our experiments, as well as in the dataset provided by [13]. Hence, when these detected but unscanned features are checked, there will be no range value $r$ returned and it is not feasible to solve the inverse of Eqs. (1)-(2).

To resolve this problem, we propose an algorithm as illustrated in Fig. 7. The algorithm is based on the fact that the markers are planar. Suppose that there is a 2D feature point $u_k = [u_k, v_k]^T$ that is detected but corresponds to an unobserved region in the raw intensity image. The azimuth $\theta_k$ of $u_k$ is determined by (2) through $\theta_k = \Theta_a(u_k - u_0)$. Define the unknown 3D point corresponding to $u_k$ as $p_k = [x_k, y_k, z_k]^T$ (the yellow point in Fig. 7). Hereafter, suppose that in the same column as $u_k$, there is a pair of observed pixels that are symmetric about $u_k$, and define their corresponding points as $p_u = [x_u, y_u, z_u]^T$ and $p_d = [x_d, y_d, z_d]^T$ (the black point in Fig. 7). As illustrated in (2), pixels in the same column approximately share the same azimuth. Thus, $p_u$, $p_k$, and $p_d$ are on the same plane, $\alpha$, which is specified by fixing the azimuth ($\theta = \theta_k$). After that, define the plane where the marker is located as $\beta$. Obviously, the
feature points w.r.t. $\alpha$ \{W\} = argmin $R_{\{W\}}$ and $R_p$, $p = \phi$ is the angle bisector of $\phi$. Consequently, the AprilTag system \[5\] will \cdots $p$, $\phi$ and $O_{\{L\}}$ are the inclinations of $p_d$, $p_k$, and $p_u$, respectively.

intersection of $\alpha$ and $\beta$ is a straight line $l$. Under the assumption that $p_u$ and $p_d$ are on the marker plane $\beta$, it can be shown that $p_u$, $p_k$, and $p_d$ are collinear and they all fall on $l$. To give a more explicit introduction to the algorithm, the side view of the plane $\alpha$ is presented in Fig. 8.

Undoubtedly, $O_{\{L\}} p_k$ is the angle bisector of $\angle p_u O_{\{L\}} p_d$ owing to $\phi_d - \phi_k = \phi_k - \phi_u$. Hence, in the light of the angle bisector properties, we have $p_k p_d / p_u p_k = O_{\{L\}} p_d / O_{\{L\}} p_u$. Note that $O_{\{L\}} p_d$ and $O_{\{L\}} p_u$ are the ranges of $p_d$ and $p_u$ which can be obtained if they are scanned. Thus, the unknown 3D coordinates of $p_k$ are estimated by $p_k = M_1 p_d + M_2 p_u$, where $M_1 = \text{diag}(\frac{1}{1+\gamma}, \frac{1}{1+\gamma}, \frac{1}{1+\gamma})$ and $M_2 = \text{diag}(\frac{1}{1+\gamma}, \frac{1}{1+\gamma}, \frac{1}{1+\gamma})$ with $\mu$ being the ratio $O_{\{L\}} p_d / O_{\{L\}} p_u$. So far we projected the 2D feature points in \{I\}, observed and unobserved by the LiDAR, back to \{L\}.

IV. LiDAR POSE ESTIMATION

The aim of LiDAR pose estimation is to seek the Euclidean transformation, $T = [R; t]$, from the world (inertia) coordinate system \{W\} to the LiDAR coordinate system \{L\}. $R$ is a $3 \times 3$ orthogonal matrix that represents the rotation. $t \in \mathbb{R}^3$ is the translation vector. Suppose that $f \in \mathbb{R}^3$ are the 3D coordinates of a feature point, the operation of $T$ on $f \in \mathbb{R}^3$ is $T \cdot f = R f + t$. LiDAR pose estimation can be resolved through optimally aligning two point sets while in real-world applications, such as SLAM and perception, the point correspondences between the two point sets are unknown, such that the correspondences are also needed to be optimally and iteratively searched [24]. However, as seen in the following, with the help of the fiducial marker system, the search for correspondences can be skipped in LiDAR pose estimation, which is a vital benefit brought by using the fiducial marker system.

Thus far two sets of 3D points are obtained. 1) $n$ feature points w.r.t \{L\}, denoted by $P_L = \{f_1, \cdots, f_n\}$, which are given by the 3D feature points estimation introduced in Section III-C; 2) $n$ feature points w.r.t \{W\}, denoted by $P_W = \{f_1', \cdots, f_n'\}$, which are predefined. Furthermore, the points in $P_L$ and $P_W$ are matched based on the ID number and vertex index given by the marker detection. Hence, the LiDAR pose estimation can be transformed into finding $[R; t]$ that optimally align $P_L$ and $P_W$. This is inherently a least square problem:

$$R^*, t^* = \arg\min_{R, t} \sum_{j=1}^n \| (R f'_j + t) - f_j \|^2$$

(3)

Considering that the point-correspondence between $P_L$ and $P_W$ is quite reliable thanks to the embedded VFM system, $R^*$ and $t^*$ can be calculated in closed form by the Singular Value Decomposition (SVD) method [2], [25]. Moreover, the points in $P_W$ are coplanar but not collinear, thus the solution of Eq. (3) is also unique and the proof is given in [25]. This is the reason why the proposed system is free from the multi-solution problem, which is unlike the VFM systems troubled by the rotation ambiguity problem [10], [15], [16]. This is a superiority of the proposed system over the planar VFM systems [5], [6].

Remark 1: Note that the LiDAR pose is specified by the definitions of \{W\} and \{L\}. It is a common method to define \{W\} by predefining the vertices [5]. However, the predefined vertices are optional in both the AprilTag system [5] and the proposed system. Without the predefined vertices, the definition of \{W\} is missing. Consequently, the AprilTag system [5] will only output the image coordinates of the vertices in the image plane and the proposed system only outputs the the 3D features
A Qualitative Evaluation

To qualitatively demonstrate the flexibility of the proposed IILFM system, Fig. 9 is presented. The usage of the IILFM system is as convenient as the VFM systems. In particular, the user can place the Letter size markers [5], [6] densely to compose a marker grid, as shown in Fig. 9(a), as well as attach some non-square markers [7] to the wall freely, as shown in Fig. 9(b). In summary, there is no spatial restriction on marker placement.

B Quantitative Evaluation

To further verify the pose estimation precision of the IILFM system, we compare the pose estimation result given by our system with the state-of-the-art LiDAR fiducial marker system, LiDARTag [13], which is also the only existing fiducial marker system for the LiDAR as far as we know, Table II is presented. Specifically, we use the rosbag (ccw_10 m.bag) provided by [13] as the benchmark on which we conduct the comparison. ccw_10 m.bag records the raw data of a 32-Beam Velodyne ULTRA Puck LiDAR scanning a 1.22 m×1.22 m AprilTag (tag16h6), from a distance of 10 meters while the plane of the LiDAR is perpendicular to the marker.

Three issues are illustrated in Fig. 11: 1) When more LiDAR scans are utilized, the pose estimation precision is slightly boosted. The reason behind it is that more LiDAR scans indicate a higher coverage percentage in the FoV [12], which implies better intensity image quality. 2) The IILFM system shows comparable precision as the VFM system. 3) Unlike the VFM system [3]–[5], the pose estimation precision does not degrade evidently as the distance increases.

Thereafter, the distance from the marker to the LiDAR is set as 2 meters and only the rotation of the LiDAR is adjusted (See Table I where the number of scans = 40). By comparing the errors in Fig. 11 and Table I, it is seen that when the plane of the LiDAR is angled towards the marker plane, the pose estimation performance of the IILFM system is as good as that when the LiDAR is perpendicular to the marker.

To validate the pose estimation accuracy of our system on the mechanical LiDAR as well as to compare the proposed system with the state-of-the-art LiDAR fiducial marker system, LiDARTag [13], which is also the only existing fiducial marker system for the LiDAR as far as we know, Table II is presented. Specifically, we use the rosbag (ccw_10 m.bag) provided by [13] as the benchmark on which we conduct the comparison.
Table II illustrates that the IILFM system is slightly inferior to LiDARTag in terms of accuracy. This is mainly because the LiDARTag system estimates the pose by finding the transmission that projects points inside the marker cluster into a predefined template (bounding box) at the origin of LiDAR [9], [13], while our system adopts the vertices to compute the pose directly just as the AprilTag [5] and ArUco [6] systems do. Our system utilizes fewer correspondences compared to [9], [13], and thus the ranging noise on a single point could have more effects on the pose estimation of our system. It should be noted that the intention of proposing the LiDAR fiducial marker system is to improve the real-world applications, such as SLAM [10], multi-sensor calibration [9], and AR [8]. As shown in Fig. 11, the proposed system outperforms the AprilTag system [5], which is already widely used in the above-mentioned applications. Moreover, the VFM systems [5], [6] use only four vertices of a marker (or a limited number of points if the marker is non-square). Note that it is definitely feasible to acquire more feature points from the inner coding area of the marker, however, this will increase the complexity of the VFM systems. Hence, following a simple but effective design concept, we also use only four vertices of a marker in our proposed system. If higher accuracy is needed, it is feasible to use the proposed marker detection information to find the point clustering of the marker in the point cloud and then input the point clustering into the pose estimation block of LiDARTag.

However, as mentioned previously, the superiority of the IILFM system is the flexibility and extensibility. For example, marker detection is infeasible for the LiDARTag system [13] in the scenarios shown in Fig. 1 and Fig. 10 since the marker placement does not satisfy the requirement of LiDARTag, and in addition, the current version of LiDARTag does not support any non-square marker, such as CCTag [7].

### C. Computational Time Analysis

A computational time analysis is conducted on a desktop with Intel Xeon W-1290P CPU. The LiDARTag [13] runs at around 100 Hz on it. The time consumption of the marker detection of our system, which includes intensity image generation, preprocessing, 2D features detection, and computation of 3D pose estimation, is shown in Table II.

| System | Vertex | x (m) | y (m) | z (m) | Error (m) |
|--------|--------|-------|-------|-------|-----------|
| Ground Truth | 1 | 9.739 | 0.758 | -0.161 | – |
| 2 | 9.940 | 0.772 | -0.732 | – |
| 3 | 10.272 | -0.414 | -0.032 | – |
| LiDARTag | 1 | 9.736 | 0.762 | -0.174 | 0.015 |
| 2 | 9.944 | 0.066 | -0.730 | 0.009 |
| 3 | 10.271 | -0.405 | -0.016 | 0.019 |
| IILFM | 1 | 9.728 | 0.766 | -0.184 | 0.013 |
| 2 | 9.963 | 0.052 | -0.705 | 0.033 |
| 3 | 10.307 | -0.432 | -0.016 | 0.044 |
| 4 | 10.045 | 0.253 | 0.356 | 0.041 |

**Fig. 11.** Pose estimation accuracy of the IILFM system and the AprilTag 3 system at different distances.

### Table I

**Table I: Pose Estimation Accuracy of the IILFM System with Different Euler Angles**

| Setup   | Term       | Ground Truth | IILFM   | Error  |
|---------|------------|--------------|---------|--------|
|         | x (m)     | 1.629        | 1.661   | -0.032 |
| pitch ≈ -15° | y (m) | -0.066       | -0.011  | -0.055 |
|         | z (m)     | 0.618        | 0.699   | -0.081 |
|         | roll (deg)| -2.673       | -4.936  | 2.263  |
|         | pitch (deg)| -15.060    | -21.145 | 6.085  |
|         | yaw (deg) | -0.171       | 0.546   | -0.717 |
| pitch ≈ 15° | x (m) | 1.684        | 1.622   | 0.062  |
|         | y (m)     | -0.063       | -0.088  | 0.026  |
|         | z (m)     | 0.590        | 0.660   | -0.070 |
|         | roll (deg)| 3.657        | 0.900   | 2.757  |
|         | pitch (deg)| 14.849     | 10.961  | 3.888  |
|         | yaw (deg) | 1.136        | -1.921  | 3.057  |
|         | x (m)     | 1.638        | 1.612   | 0.027  |
| yaw ≈ -15° | y (m) | -0.618       | -0.237  | -0.381 |
|         | z (m)     | 0.610        | 0.585   | 0.025  |
|         | roll (deg)| -0.009       | -0.792  | 0.783  |
|         | pitch (deg)| -0.662     | -2.088  | 1.426  |
|         | yaw (deg) | -15.387      | -17.937 | 2.550  |
| yaw ≈ 15° | x (m) | 1.627        | 1.652   | -0.005 |
|         | y (m)     | -0.034       | -0.146  | -0.098 |
|         | z (m)     | 0.609        | 0.553   | 0.037  |
|         | roll (deg)| 0.490        | -1.994  | 2.484  |
|         | pitch (deg)| -0.388     | -0.359  | -0.028 |
|         | yaw (deg) | 14.924       | 10.678  | 4.246  |
features, mainly depends on the size of the intensity image and the embedded VFM system. Suppose that the embedded VFM system is AprilTag 3 [5], the marker detection takes around 7 ms (approx 143 Hz) in the case shown in Fig. 9(c) where \( \Theta_a = 0.3^\circ \), \( \Theta_i = 1.33^\circ \), and intensity image size = 1201 \( \times \) 27. For the case shown in Fig. 9(a), the marker detection takes around 25 ms (approx 40 Hz), where intensity \( \Theta_a = \Theta_i = 0.05^\circ \) and image size = 771 \( \times \) 591. The time consumption of the following pose estimation process is around 1.8 \( \mu \)s as the closed-form solution can be obtained directly through SVD [25] (Refer to Section IV).

D. Limitations Analysis

There are some limitations on the application of the LiDAR fiducial marker systems. First, the distance from the object (marker) to the LiDAR must exceed the minimum detectable range. This limitation is caused by the hardware attributes and it affects all the LiDAR applications not only the LiDAR fiducial marker systems. Secondly, to utilize the solid-state LiDAR, it is required to wait for the growth of the scanned area inside the FoV to obtain a relatively dense point cloud. Again this is a limitation caused by the hardware attributes. In contrast, the mechanical LiDAR only requires one LiDAR scan to work, however, a larger and simpler marker is recommended if the angular resolution is large. Finally, the false positives are occasionally found in the experiments while the issue of wrong ID detection (not exactly the same as the false positive but similar) is also reported in LiDARTag [13]. For the proposed system, the false positive rate is mainly determined by the embedded VFM system, thus novel VFM systems with lower false positive rate, such as [5], [6], are preferred.

VI. CONCLUSIONS

In this work, a novel intensity image-based LiDAR fiducial marker is proposed. The main novelties of the system are the convenience and extensibility. The convenience refers to how users can print the marker on Letter size letter and place it freely, which is the same way to use the visual fiducial marker systems. The extensibility denotes that the popular visual fiducial marker systems, such as AprilTag [5], ArUco [6], CTag [7], and newly proposed ones in the future can be easily integrated into the proposed system. Hence, our system has no restrictions on the marker shape. Experimental results demonstrate that the proposed system has the similar accuracy as the visual fiducial marker system in terms of pose estimation.

Future work includes introducing deep-learning-based point cloud completion methods [26], as well as motion compensation strategies since it is assumed that the input point cloud is not overly sparse and already motion compensated.

ACKNOWLEDGMENT

The authors would like to thank Yicong Fu, Hassan Alkomy, and Brian Lynch for constructive discussions, Jiunn-Kai Huang for providing his experimental results.

REFERENCES

[1] L. Fan, X. Xiong, F. Wang, N. Wang, and Z. Zhang, “RangeDet: In defense of range view for Lidar-based 3D object detection,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2021, pp. 2918–2927.
[2] T. D. Barfoot, State Estimation for Robotics, Cambridge, U.K.: Cambridge Univ. Press, 2017.
[3] J. Wang and E. Olson, “AprilTag 2: Efficient and robust fiducial detection,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., 2016, pp. 4193–4198.
[4] E. Olson, “AprilTag: A robust and flexible visual fiducial system,” in Proc. IEEE Int. Conf. Robot. Automat., 2011, pp. 3400–3407.
[5] M. Kroggius, A. Haggenmiller, and E. Olson, “Flexible layouts for fiducial tags,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., 2019, pp. 1898–1903.
[6] F. J. Romero-Ramirez, R. Muñoz-Salinas, and R. Medina-Carnicer, “Sppeded up detection of squared fiducial markers,” Image Vis. Comput., vol. 76, pp. 38–47, 2018.
[7] L. Calvert, P. Gurdjos, C. Griwoz, and S. Gasparini, “Detection and accurate localization of circular fiducials under highly challenging conditions,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 562–570.
[8] D. Avola, L. Cinque, G. L. Foresti, C. Mercuri, and D. Pannone, “A practical framework for the development of augmented reality applications by using ArUco markers,” in Proc. Int. Conf. Pattern Recognit. Appl. Methods, 2016, vol. 2, pp. 645–654.
[9] J.-K. Huang and J. W. Grizzle, “Improvements to target-based 3D LiDAR to camera calibration,” IEEE Access, vol. 8, pp. 134,101–134,110, 2020.
[10] R. Muñoz-Salinas, M. J. Marin-Jimenez, and R. Medina-Carnicer, “SPM-SLAM: Simultaneous localization and mapping with squared planar markers,” Pattern Recognit., vol. 86, pp. 156–171, 2019.
[11] H. Ye, Y. Chen, and M. Liu, “Tightly coupled 3D LiDAR inertial odometry and mapping,” in Proc. IEEE Int. Conf. Robot. Automat., 2019, pp. 3144–3150.
[12] J. Lin and F. Zhang, “Loam livox: A fast, robust, high-precision lidar odometry and mapping package for LiDARs of small FoV,” in Proc. IEEE Int. Conf. Robot. Automat., 2020, pp. 3126–3131.
[13] J.-K. Huang, S. Wang, M. Ghaifari, and J. W. Grizzle, “LiDARTag: A real-time fiducial tag system for point clouds,” IEEE Robot. Automat. Lett., vol. 6, no. 3, pp. 4875–4882, Jul. 2021.
[14] G. P. Meyer, A. Laddha, E. Kee, C. Vallespi-Gonzalez, and C. K. Wellington, “LASERNet: An efficient probabilistic 3D object detector for autonomous driving,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2019, pp. 12677–12686.
[15] T. Collins and A. Bartoli, “Infinitesimal plane-based pose estimation,” Int. J. Comput. Vis., vol. 109, no. 3, pp. 252–286, 2014.
[16] Y. Liu, H. Schofield, and J. Shan, “Navigation of a self-driving vehicle using one fiducial marker,” in Proc. IEEE Int. Conf. Multisensor Fusion Integration Intell. Syst., 2021, pp. 1–6.
[17] C. Lu, S. Xia, M. Shao, and Y. Fu, “Arc-support line segments revisited: An efficient high-quality ellipse detection,” IEEE Trans. Image Process., vol. 29, pp. 768–781, 2020.
[18] C. Lu, S. Xia, W. Huang, M. Shao, and Y. Fu, “Circle detection by arc-support line segments,” in Proc. IEEE Int. Conf. Image Process., 2017, pp. 76–80.
[19] S.-F. Ch’ng, N. Sogi, P. Purkait, T.-J. Chin, and K. Fukui, “Resolving marker pose ambiguity by robust rotation averaging with clique constraints,” in Proc. IEEE Int. Conf. Robot. Automat., 2020, pp. 9680–9686.
[20] P. Kovesi, “Good colour maps: How to design them,” 2015, arXiv:1509.03700.
[21] R. B. Rusu and S. Cousins, “3D is here: Point cloud library (PCL),” in Proc. IEEE Int. Conf. Robot. Automat., 2011, pp. 1–4.
[22] R. A. Haddad and A. N. Akansu, “A class of fast Gaussian binomial filters for image restoration,” IEEE Trans. Image Process., vol. 39, no. 3, pp. 723–727, Mar. 1991.
[23] Y. Liu, H. Schofield, and J. Shan, “Intensity-image-based LiDAR fiducial marker system,” 2022, arXiv:2203.01816.
[24] J. Seraphin and G. Grisetti, “NICP: Dense normal based point cloud registration,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., 2015, pp. 742–749.
[25] K. S. Arun, T. S. Huang, and S. D. Blostein, “Least-squares fitting of two 3D point sets,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 9, no. 5, pp. 698–700, Sep. 1987.
[26] H. Xie, H. Yao, S. Zhou, J. Mao, S. Zhang, and W. Sun, “GRNet: Gridding residual network for dense point cloud completion,” in Proc. Eur. Conf. Comput. Vis., 2020, pp. 365–381.