A4LidarTag: Depth-based fiducial marker for extrinsic calibration of solid-state Lidar and camera

Yusen Xie[1†], Lei Deng[2†], Ting Sun[1*], Yeyu Fu[1], Jian Li[2], Xinglong Cui[1], Hanxi Yin[2], Shuixin Deng[1], Junwei Xiao[2], Baohua Chen[2]

Abstract— Visual-based simultaneous localization and mapping (SLAM) systems perform weakly in object tracking and map reconstruction due to the unreliable depth measurement originating from image-only data. Light Detection and Ranging (LiDAR) can be coupled to overcome the drawback of uncertain depth estimation. The prerequisite for performing data fusion is to align visual-Lidar sensors to a specific coordinate system with extrinsic pose by calibrating. The conventional extrinsic calibration frameworks either rely on markers in artificial large-size calibration boards or uncontrollable natural scenes (Fig. 2), limiting stability and convenience. In this paper, we have designed a novel marker pattern, A4LidarTag, composed of circular holes. The difference in depth measurement is used to encode location information. Based on A4LidarTag, the automatic extrinsic calibration framework between solid-state Lidar (SSL) and the camera is developed. The proposed framework can be implemented in close range (within 1 meter) and on an A4-size calibration board. The average reprojection error resulting from Lidar point clouds projection is about 0.12 pixels. Experiments show excellent efficiency and versatility in both indoor and outdoor scenes. Source code is available on https://github.com/xieyuser/A4LidarTag.

Index Terms— Computer Vision for Automation; Calibration and Identification; depth based A4LidarTag; solid-state Lidar and camera

I. INTRODUCTION

The drawback inherent in the visual SLAM system [1]–[4] is a lack of information to obtain precise depth, which leads to inaccurate localization, failure of tracking, and imprecise map reconstruction. On the contrary, Lidar can provide accurate distance measurement in complex and dynamic real-world scenes [5]–[7]. Integrating Lidar into the visual SLAM system helps SLAM overcome the camera's drawbacks in special scenes. Thus more and more attention is paid to visual-Lidar SLAM systems in the research and the industry [8]–[10].

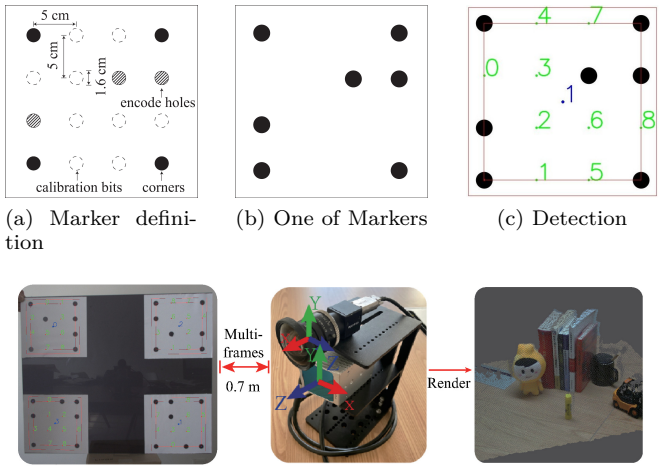
Manuscript received: December, 1, 2021; Revised March, 16, 2022; Accepted April, 20, 2022.

This paper was recommended for publication by Editor Cesar Cadena upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by the Program for the Top Young Talents of Beijing High-level Innovation and Entrepreneurship(G04070017)(†Yusen Xie and Lei Deng are co-first authors.) (*Corresponding author: Ting Sun.)

[1]Yusen Xie, Lei Deng, Ting Sun, Yeyu Fu, XingLong Cui and Shuixin Deng are with Joint International Research Laboratory of Advanced Photonics and Electronics, Beijing Information Science & Technology University, Beijing, China, {xieyusen, sunting}@bistu.edu.cn, dally211@163.com

[2]Jian Li, Hanxi Yin, Junwei Xiao and Baohua Chen are with Department of Automation, Tsinghua University, Beijing, China

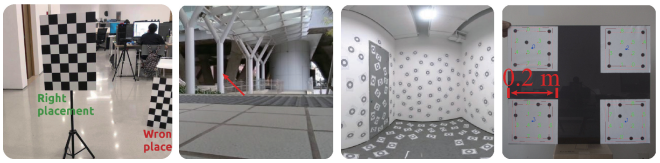
Digital Object Identifier (DOI): see top of this page.



(a) Marker definition (b) One of Markers (c) Detection

(d) Illustration of our calibration work

Fig. 1. (a)(b)The A4LidarTag we designed for automatic extrinsic calibration. Each marker has four corners, three encode holes and nine calibration bits. Calibration bits are the feature points for 3D-2D matching. (c) Text in blue is the marker ID, and the text in green is the sorted calibration bits that the marker has provided. Red lines are the border of the marker. (d) Visual-Lidar system captures multi-poses and multi-frames A4LidarTag-based calibration board, then rendering image from the camera to point clouds from Lidar with the extrinsic pose.



(a) Checkerboard (b) PES (c) Singleshot (d) Ours

Fig. 2. Different frameworks for automatic extrinsic calibration. (a) Target-based framework with large marker and remote distance [11]. (b) Target-less framework requires sufficient geometry structure [12]. (c) The room with the amount of markers is hard to build [13]. (d) Our A4LidarTag.

Camera and Lidar contain different modalities of data. Performing data fusion needs to align two sensors together in space. Once the accurate extrinsic pose between the camera and Lidar is obtained by calibrating, they can be transformed into each other. The sensors system with the abundant features of the image and the depth measurement of Lidar can be regarded as an enhanced RGB-D camera [8], [10], which can be used as a measurement system for visual-Lidar systems [8], [10]

or a ground truth supervised system for stereo vision [14].

With the continuous development of materials and fabrication processes, solid-state Lidars (SSL) based on Micro Electro Mechanical Systems (MEMS) technology (e.g., Livox series Lidar and Intel Realsense L515) have gained attention [6], [7], [11], [12], [15] for their excellent performance in terms of the low cost and the denser Lidar point clouds compared with the conventional multi-line spinning Lidars (e.g., Velodyne). Figure 3 shows three different types of Lidars. The well-designed mechanical structure of SSL ensures that point clouds scanned at different times do not repeat, and the field of view is limited instead of 360°. This paper focuses on the calibration of SSL and camera.

The general calibration framework aims to find the correspondence of 2D-3D features and optimize relative extrinsic poses. The feature extraction methods can be divided into two categories, the targetless methods based on the external environment [7], [12] and the target-based methods based on fiducial markers [11], [16]–[19]. Targetless methods require specific scenes equipped with sufficient geometric features (Fig. 2b). The size of markers in target-based methods is usually large, thus resulting in remote working distance and inconvenience (Fig. 2a).

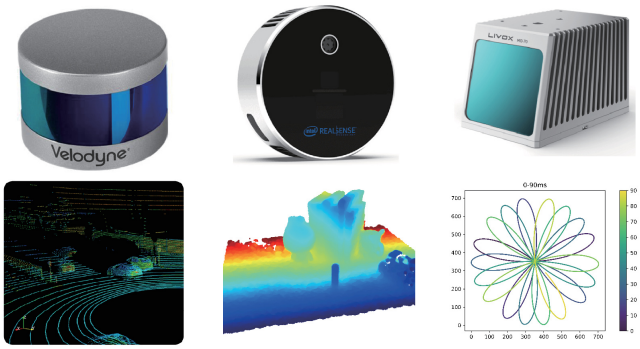
The depth-based A4LidarTag (Fig. 1a, 1b) proposed in this paper can solve extrinsic poses with high accuracy and efficiency at a close working distance (Fig. 1d). In this paper, the procedures of the extrinsic calibration are shown as follows: (a) Integrating camera and Lidar into the visual-Lidar system. (b) Capturing the images and point clouds of A4LidarTag. (c) Projecting point clouds to a 2D plane in the local Lidar coordinate system, followed by the linear interpolation of the depth image. (d) Detecting A4LidarTag and matching 2D-2D feature points in interpolated depth images and images. Un-projecting 2D feature points in depth image to get 3D feature points. (e) PnP [20] and Ransac [21] were implemented to solve extrinsic poses.

The main contributions of this paper are:

1) A depth-based marker pattern, A4LidarTag, is proposed to improve the accuracy and efficiency of extrinsic calibration in close working distance.
2) An automatic extrinsic calibration pipeline based on A4LidarTag for the visual-Lidar system is developed to improve the simplicity and effectiveness of calibration.
3) Experiments in indoor and outdoor scenes show that our algorithm can obtain accurate extrinsic poses robustly.

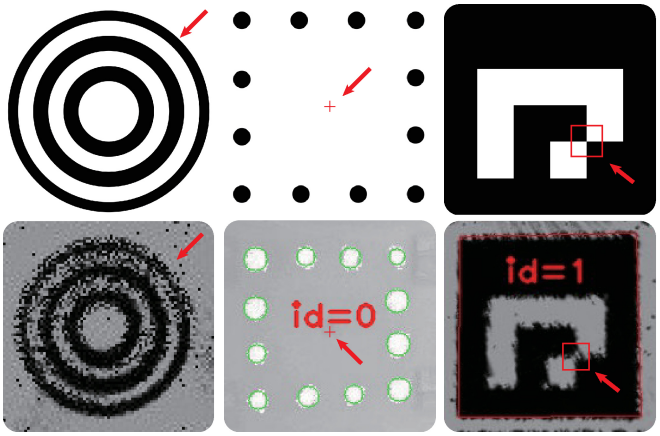
II. Related Work

The earliest known work on 3D-Lidar and camera calibration comes from the CMU robotics institute [16], using a calibration board and manually labeling multiple point clouds image pairs. Huang [19] et al. also develops



(a) Velodyne VLP-16 (b) RealSense L515 (c) Livox MID-70

Fig. 3. (a) Velodyne. Conventional mechanical multi-spinning Lidar with 360° of FOV. (b) Intel Realsense L515 (FOV: 70° × 55° (±3°)) and its scanned denser point clouds plotted by Open3D [22]. (c) Livox Mid-70 (FOV: 70.4° × 70.4°). It scans in irregular way. The scanning trajectory of 3D points are projected as image, where the color encodes the sampling time in 90 (ms).



(a) CCTag (b) Pi-Tag (c) Aruco

Fig. 4. The first image row is the variety marker in the image, and the second one is the result detected in the Lidar reflectivity map. These markers perform weakly in Lidar because of sparse point clouds and low-resolution reflectivity map. CCTag [23] does not work with low-resolution Lidar reflectivity maps. The single shot of Pi-Tag [24] based on projection invariance provides a few reference points (only the center of the marker). Arucomarker [25] can only be successfully detected when it is large enough (A3 print).

a method for point clouds edge extraction in the Matlab Lidar toolbox [18] in 2020. They extracts the calibration board in point clouds by optimizing the relative pose H between the calibration board and a priori artificial reference calibration board as much as possible.

SSL cannot utilize the calibration method of multi-spinning line Lidar because of its limited field of view and non-repetitive scanning way. Cui [11] uses a large checkerboard and propose a multi-frame optimization point clouds algorithm. The whole system need a laser printed black and white checkerboard with strict vertical placement, and the calibration average reprojection error (RMS) is 2.11 pixels in different Livox devices. Camvox [7] uses the targetless method to extract edges in the

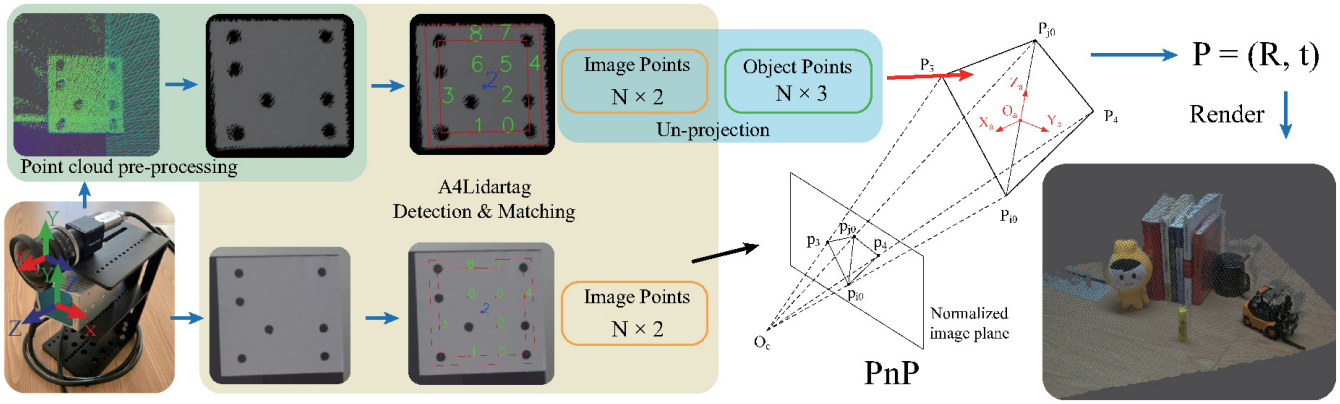


Fig. 5. Overview of our pipeline of calibration based on A4LidarTag. Coupled sensors system collects both images and points cloud, projecting point clouds as a two-dimensional depth image. Followed by A4LidarTag detection to estimate two-dimensional uv, three-dimensional will be unprojected with two-dimensional depth image in SSL. PnP [20] and Ransac [21] are used to estimate $P = (R, t)$. Finally, point cloud is rendered with a color image to verify calibration.

outdoor scenes for images, then solving the correspondence by ICP. In this work, feature matching is regarded as an optimization problem, which often lead to poor performance due to significant differences in features between different modalities or the issue of local optima. Similarly, Yuan [12] extracts the edge features in point clouds and image respectively and optimized the feature correspondence to solve the best extrinsic pose. The feature correspondence error is about 1 pixel in all scenes included in the experiment. Wang et al. [15] perform a 3D corners estimation of the calibration target based on the reflectivity of SSL and searched for the correspondence in the image and point clouds to complete the calibration. The targetless calibration pipeline based on environmental features is not always available because overmany geometry features are required.

Conventional markers designed for the visual system often perform weakly in Lidar, as is shown in Figure 2 and Figure 4. Lidar is of low resolution and lacks abundant features as image. Lidartag [26] with Apriltag enlarged and printed on a tilted placed calibration board, using multi-line scanning Lidar to extract the information contained in different reflectivity encoding modules. Lidartag can work under the absence of light conditions for Lidar marker detection and recognition. However, this can only be realized when a stricter placement of 45° is required. Cui [11] uses the classical checkerboard as the marker for corners estimation, such markers designed for Lidar were mostly based on reflectance. They often require further calibration distance and can not be spread in the entire field of view of Lidar. In this work, the method of estimating the parameters from the feature points of an incomplete marker placement in FOV is undoubtedly imprecise. Other frameworks based on reflectance are shown in Figure 4. To the best of our knowledge, we have not encountered specific marker systems that can be applied to both camera and Lidar in a close range.

III. Method

Our approaches are organized as follows. Pre-processing of the collected point clouds is firstly implemented in III-A. The design and detection of A4LidarTag proposed above are introduced in III-B. Both the RGB images captured by the camera and the depth images generated by the pre-processing of point clouds are input into the A4LidarTag detection algorithm in Section III-B to extract feature points. Automatic target-based extrinsic calibration pipeline is in III-C. At last, in Section IV, image rendering of point clouds is used to verify our calibration results in both indoor and outdoor scenes. An overview is summarized in Figure 5.

A. Pre-processing of Lidar point clouds

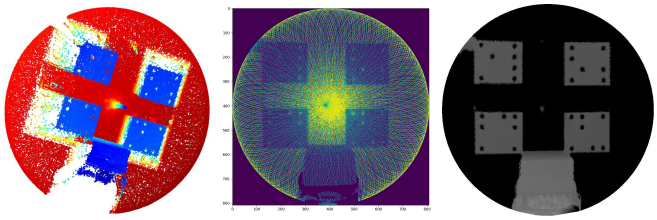
SSL can collect denser point clouds in a long enough integration time, which can maximize the details of the object's surface. Giving the horizontal or vertical field of view $FOV_{u/v}$ and the minimal horizontal or vertical angle resolution $R_{u/v}$, physical intrinsic of SSL K_{ssl} calculated by Eq.1,2,3 can be used to project 3D point clouds $X = [x, y, z, 1]^T$ to the 2D plane $Y = [u, v, 1]^T$, ($\mathbb{R}^3 \mapsto \mathbb{R}^2$ in Eq.4). $f_{x/y}$ is the focal length of SSL and $c_{x/y}$ is the size of projected 2D image. Z in Eq.4 is the depth value of a point in the 2D plane.

$$c_{u/v} = \frac{FOV_{u/v}}{R_{u/v}} \tag{1}$$

$$f_{u/v} = \frac{c_{u/v}}{\tan(FOV_{u/v}/2)} \tag{2}$$

$$K_{ssl} = \begin{pmatrix} f_u & 0 & c_u & 0 \\ 0 & f_v & c_v & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{3}$$

$$Z \begin{bmatrix} u & v & 1 \end{bmatrix}^T = K_{ssl} \begin{bmatrix} x & y & z & 1 \end{bmatrix}^T \tag{4}$$



(a) PC captured (b) Projection (c) Depth image

Fig. 6. The pipeline of pattern recognition from point clouds. (a) Original 3D point clouds SSL captured. (b) The two-dimensional image projected from an original point clouds with K_{ssl} Eq.3. (c) Depth image interpolated from (b).

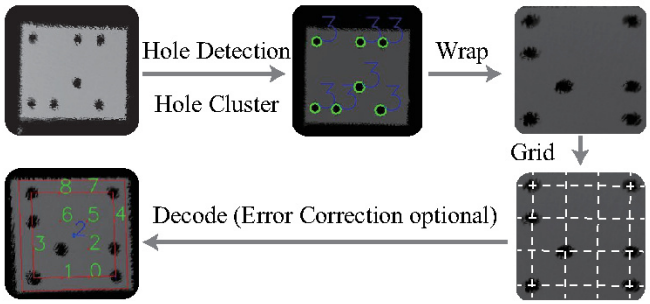


Fig. 7. Pipeline of A4LidarTag detection.

Linear interpolation operator [27] was applied to unstructured $\mathbb{R}^2$ point clouds to generate mesh grid image $\mathfrak{D}$ or $\mathfrak{I}$ of depth dimension D or reflectivity dimension I. They can be inputted for Lidar feature detection. Preprocessing of point clouds is detailed in Figure 6.

B. A4LidarTag

To avoid the limitations mentioned in Section II, a close-range depth-based marker system named A4LidarTag has been designed. The close calibration distance allows the calibration feature points to be spread over the entire field of view of SSL, through which the difference in depth values can be used to encode patterns. A4LidarTag generated in Section III-B.1 is identifiable individually with a unique layout and feature points. Moreover, an error tolerance mechanism is implemented to improve the robustness. Because of these features, A4LidarTag has the potential to become a multi-modalities localization and navigation marker system in the future, similar to visual navigation benchmarks ArucoMarker [25]. Figure 1a is an example of a designed A4LidarTag.

1) Automatic dictionary generation and marker generation: A4LidarTag comprises 16 geometric holes with the 4×4 square size. To solve the warping problem under different sensor poses, four corners of the marker for arranging patterns must be set initially, so the actual encoding holes are only 12 bits. Apparently, the actual marker capacity cannot be so large that some bits must be sacrificed to improve the redundancy of information, which can be used for error correction.

The algorithm of dictionary generation is described in Algorithm 1. The algorithm starts from an empty dictionary D. In the beginning, the algorithm generates a 12 bits 0&1 sequence M_{12} with a random fixed number C_0 of 1 (e.g., 001010000100). The number of 1 indicates the number of holes on the marker. Fewer holes will reduce the error generated by detecting the holes but declining the number of available markers. In the calibration experiment, we choose the number of encoding bits to 3. The newly generated marker is defined as M_{16}, which consists of M_{12} and 4 pre-defined corners in index 0,3,12,15. (e.g., 1001101000011001)

The distance between different markers is defined by Hamming distance [28]. If the distance between a randomly generated marker (including four rotations) and all markers in the current dictionary is less than the threshold d, then the marker will not be accepted. The dictionary generation program will stop when the dictionary contains the desired number of markers.

Taking the length of the dictionary into consideration, the available encoding space to satisfy the threshold d will shrink as the length increases. It won't be easy to generate markers that satisfy the condition eventually. The distance threshold d is reduced after a certain number of *maxiters* iterations to solve this problem.

After generating a patterned dictionary, the array M_{16} will be reshaped to 4×4 matrix, in which the black hole represents 1, and no hole represents 0. One of the A4LidarTag generated is demoed in Figure 1b.

Algorithm 1 Automatic Dictionary Generation

```
D ← φ, init empty dictionary
d ← d^0, init minimal hamming distance
c ← c^0, init encode bits
maxiters ← maxiters^0, init max iters
curiter ← 0, init current iter

while D < desired size S do
    generate 12 bits M_12 (contains c of 1)
    define M_16 = M_12 + 4 corners
    if element in D do HammingDistance > d then
        D ← M_16
        curiter = 0
        break
    else
        curiter = curiter + 1
    end if

    if curiter > maxiter then
        d ← d − 1
        curiter = 0
    end if
end while
```

2) Hole detection and cluster: The overview of detection is detailed in Figure 7. A4LidarTag is made up of

circular holes, so the robust and accurate holes detection algorithm is the basis of subsequent detection. Hole thresh operators have been implemented, e.g., radius, gray, circularity, convexity, inertia radio.

$$F(x,y) = \iint_I (x-\bar{x})(y-\bar{y})P(x,y))\mathrm{d}x\mathrm{d}y \tag{5}$$

$$E_I = F(x,x)\sin^2\theta - F(x,y)\sin 2\theta + F(y,y)\cos^2\theta \tag{6}$$

Radius and gray values are basic image parameters of holes. Circularity denotes the ratio of the hole size to its round. Convexity denotes the ratio of the hole size to its convex hull size. The term E_I defined in Eq.6 can be used to describe the rotational inertia characteristics. F is defined in Eq.5, $\bar{x}$ and $\bar{y}$ are the center of image P, x and y are the pixels coordinate of image P. θ is the image orientation in the definition of polar coordinates. Since E_I is a quadratic form, we can conclude from the eigenvalues and eigenvectors that the ratio of maximum eigenvalue and minimal eigenvalue is close to 1 when the hole is circular from [29].

Each of the hole operators above searches the image for potential holes. Only if the region satisfies the threshold of all operators will the region be considered as a hole.

After hole detection, the potential marker regions are segmented using the DBSCAN [30] density clustering method, and the density radius ε is set as α times the average hole radius. The spacing between markers cannot be too close because of the characteristics of density clustering.

3) Warp with convex hull and decode: Due to the uncertainty of the camera or Lidar pose, the board where the marker is located is not strictly parallel to the sensor. It is usually necessary to use the perspective transformation to warp the original image to the frontal view of the sensor to facilitate detection. When generating the A4LidarTag, we reserve the four corners at the edge of the holes as the initial value of the perspective transformation. To extract the four corners in the holes, the convex hull [31] of the hole needs to be extracted at first.

The result of the convex hull is usually not the consequence of the real corners. Using the law that holes in the same line are also in the same line after multi-view transformation, the angle computed by this hole and its left and right nearest neighbors was used. And the one whose angle is close to 180° must be the encode hole, and rather it is a corner.

After the perspective transform, the front view is re-detected to ensure accuracy. The value of the center pixel in each hole is compared with gray thresh to decode the 0&1 sequences M_{16}. Finally, the algorithm compares the decoding result with the ID in the dictionary D. And confirming that there is no error to complete the detection. Otherwise, error correction is required.

4) Error correction: In the case where each hole is detected individually, it is easy to fail to recognize the predefined pattern due to false detection. Therefore it is better for the marker to have some ability to correct errors.

When generating the marker dictionary, the Hamming distance [28] among separate markers is maximized so that the marker can still be recognized under error holes detection.

The minimal distance d of the separate marker in generated dictionary D in the experiment is 4, according to the quantitative law (Eq. 7) between d and the max bit number t which can be corrected, A4LidarTag with encoding bits of 3 can only correct 1-bit error. If there is more than one marker with the same distance d simultaneously, the detection will be considered a failure and return null. The concrete results is shown in the experimental part.

$$d = 2t + 1 \tag{7}$$

C. Calibration between camera and Lidar

1) A4LidarTag detection and 3D-2D matching: Interpolation depth images from SSL and images from the camera are inputted to detect A4LidarTag. The calibration points are sorted to seek the 3D-2D matching, which is used as the corresponding feature points in our algorithm.

2) Un-project Lidar corresponding points: Having known 2D image points $Y = [u, v, 1]^T$ detected from image and depth image from Lidar, depth value $\mathfrak{D}$ and intrinsic K_{RGB}, three-dimensional points $X = [x, y, z, 1]^T$ can be calculated by the following formula Eq.8.

$$\begin{bmatrix} x & y & z \end{bmatrix}^T = \mathfrak{D} \cdot K_{RGB}^{-1} \begin{bmatrix} u & v & 1 \end{bmatrix}^T \tag{8}$$

3) Calibration with PnP and Ransac: Knowing two-dimensional image points and Lidar three-dimensional points, camera pose $P = (R, t)$ can be found by the efficient EPnP [20] algorithm. To reduce the disruption of outliers, Ransac [21] is implemented to restrict the outer points. All inner points calculate the pose by changing the threshold ξ of outer points in Ransac [21].

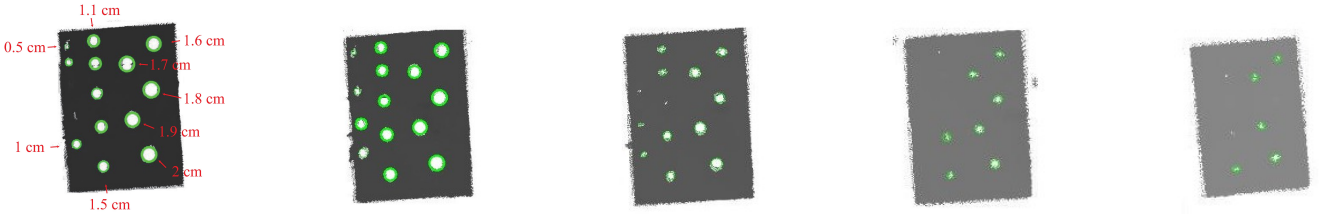
IV. Experiments

Livox series SSL are widely used in outdoor scenes, with the detection distance reaching more than 300 meters. The sensor system for the outdoor experiment is shown in Figure 5, with the Hikvision RGB camera [1] at the top and the Livox MID-70 [2] at the bottom, the relative pose of the two sensors kept constant. Intel Realsense L515 [3] is a revolutionary solid-state optical depth camera with proprietary MEMS oscilloscope scanning technology, low power consumption, and low cost

[1] https://www.hikvision.com/en/products/IP-Products/Network-Cameras/

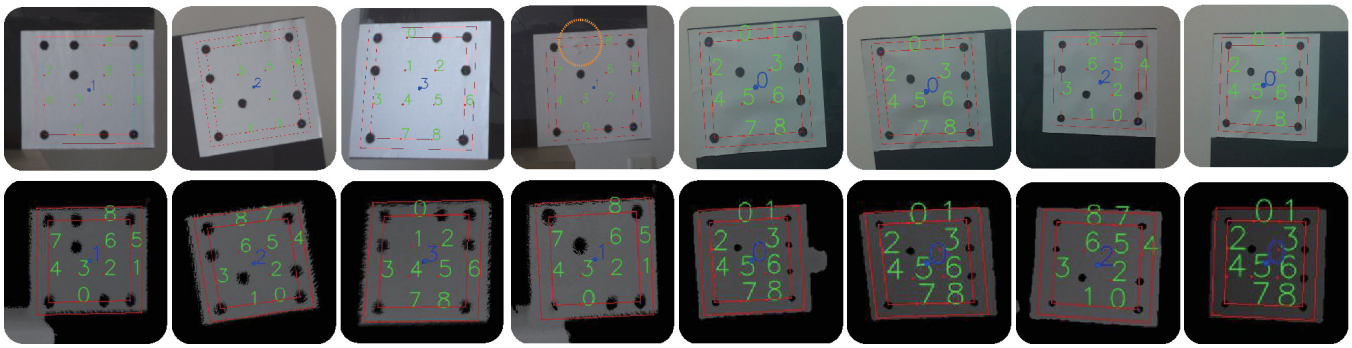
[2] https://www.livoxtech.com/mid-70

[3] https://www.intelrealsense.com/lidar-camera-l515/



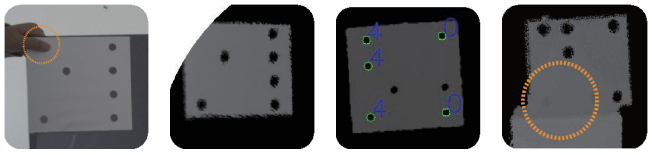
(a) 30 cm (b) 70 cm (c) 100 cm (d) 130 cm (e) 160 cm

Fig. 8. Visualization results of different holes size at different detection distance is detailed here. There are 15 holes burrowed with different sizes from 0.5-2 (cm) in 0.1 (cm) radius length increments on A4 paper. As (a) shows, these holes are arranged from the top left to the bottom right.



(a) (b) (c) (d) (e) (f) (g) (h)

Fig. 9. Illustration of the A4LidarTag detection. Rows 1 are images, and rows 2 are the corresponding depth images of SSLs. All examples contain different sensor poses. (a)(b)(c) show the successful detection in Livox MID-70. (d) Illustration of error correction. Orange circles are the mistakes we deliberately made. The marker with ID 1 still performs well when the encode holes are obscured. (e)(f)(g)(h) show the performance of our detection algorithm in L515.



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

Fig. 10. Illustration of some detection failures. (a)Warp failures because of corner missing. (b) Marker near the edge of the SSL FOV. (c) Cluster failure due to the missing detection of some holes. (d) Unclear depth differences with objects in the environment.

for a wide range of applications in indoor scenes. And its maximum detection distance can reach 9 meters. L515 is used to validate the performance of our algorithm in indoor scenes. Reprojection error (RMS) is regarded as the benchmark of calibration results. In our experiments, we assume that the intrinsic of the camera is already computed by [33].

A. Different hole sizes over different working distances

The reasonableness of the design of the calibration board was firstly verified. Holes were burrowed on A4 paper from 0.5-2 (cm) in 0.1 (cm) radius length increments. The distance of the calibration board from the sensor increased from 0.3 meters to 1.6 meters gradually. Compared with images, we were more concerned about the performance of the hole on the depth image. Figure 8 shows the detection results of different hole sizes at different distances. As envisioned by experience, the detection difficulty increases as the distance between the calibration board and the sensor increasing. Under the circumstance that the sensors are close to the marker, more markers cannot be obtained when the field of view is limited, and the markers are required to have a strong error tolerance.

Theoretically, the algorithm can work in longer distances. But in order to improve the convenience of calibration, it is unnecessary to increase the working distance, the performance is optimal at closer distances. The radius of holes we set in the experiment is 0.08 (m), and the distance between the calibration board and the sensors is 0.7 (m) in Livox sensor system. Point clouds of L515 are well structured in this way, the hole radius and calibration distance can be adjusted appropriately.

B. Marker recognition and error correction

In order to highlight the difference between the hole and the foreground in the image, the acrylic board is used as a marker background. The acrylic board, similar to the properties of glass, can have a particular color while allowing the Lidar to pass through, which is an excellent material to use as the base of the calibration board. The detailed detection results in image and Lidar are shown in Figure 9. Multiple sensor data sets are captured simultaneously during the calibration process to ensure

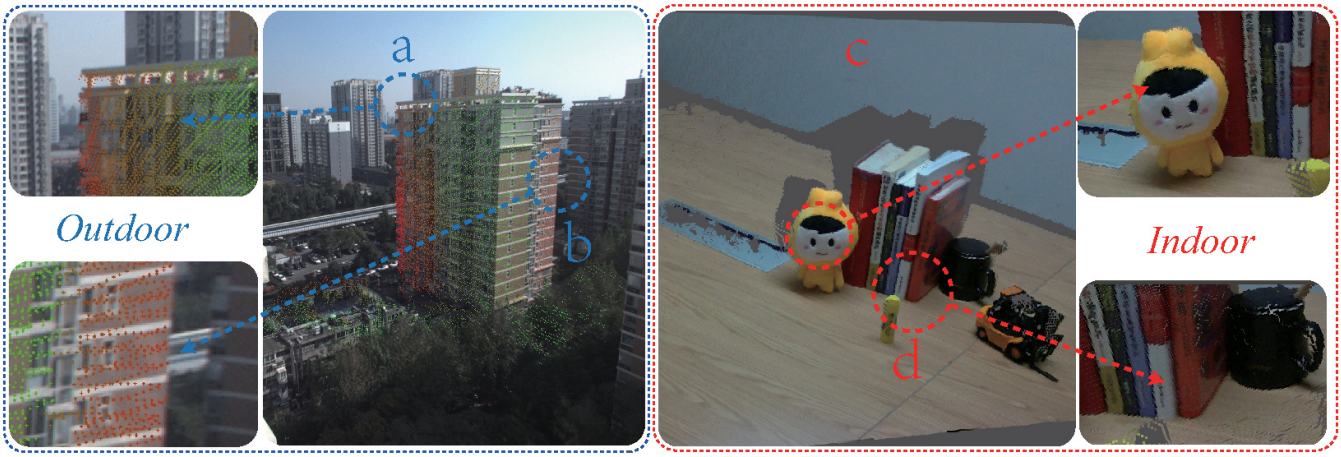


Fig. 11. Scene Left shows the result of point clouds reprojection in Livox MID-70. The color indicates the difference in depth. We can see that point clouds are clearly separated at the junction of the buildings in (a, b), which proves that the calibration result is reliable. Scene Right is the result of rendering the image into a point clouds in PyVista [32] captured by L515, the book spine and the doll is very clearly visible in (c, d).

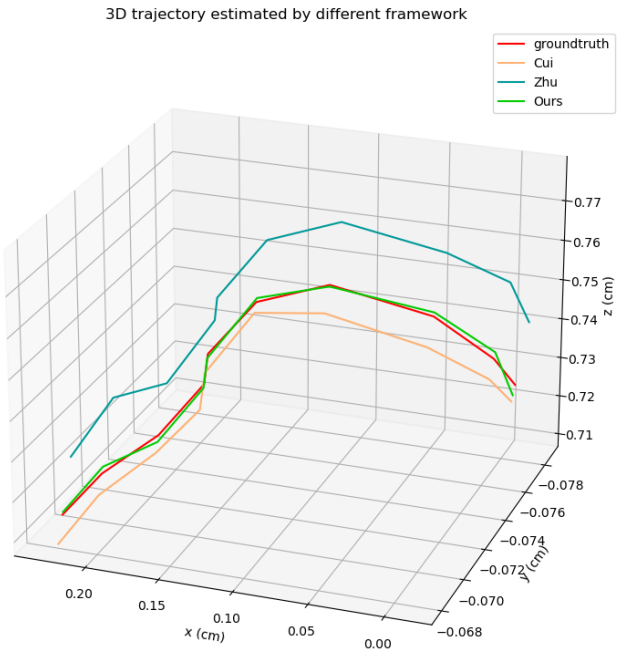


Fig. 12. The sensor was moved around the calibration target and computed poses of a certain point in the marker in ten different positions. And the trajectory was fit by combining these discrete points in order. Groundtruth in red is the trajectory computed by built-in extrinsic.

that the calibration board has as many different poses as possible. Experiments have proven that our method is highly accurate, robust and perfectly performed in dynamic handheld calibration scenes. Because of the imperfect design of A4LidarTag, it is inevitable that some cases of detection failure in Figure 10. Some improvements can be implemented in the future.

TABLE I

The RMS calculated by Ransac when setting different thresholds ξ shows that the number of interior points decreases, and the RMS (unit: pixels) decreases when the threshold is lowered.

ξ	16	8	4	2	1	0.5	0.2
inliers-Livox	216	216	202	130	46	17	5
RMS-Livox(pixels)	1.8654	1.8654	1.6897	1.1989	0.6054	0.2894	0.1229
inliers-L515	90	90	88	52	46	28	9
RMS-L515(pixels)	1.0179	1.0179	1.0053	0.9017	0.4718	0.2806	0.1391

TABLE II

Comparison of the quantity statistics of our method and others in SSL extrinsic calibration. AVG-RMS denotes the average RMS (unit: pixels) of all corresponding points, Ransac was implemented to decline noise.

Methods	Zhu [7]	Cui [11]	Ours-livox	Ours-L515
AVG-RMS (NO Ransac)	5.88	2.11	1.86	1.02
AVG-RMS (Ransac)	-	-	0.12	0.04

TABLE III

Comparison of the quantity statistics of our method and others in SSL extrinsic calibration. Error in the three-axis denotes the average distance (unit: cm) of all trajectory points.

Methods	Zhu [7]	Cui [11]	Ours
Error-x(cm)	0.1365	0.0501	0.0107
Error-y(cm)	0.0974	0.0342	0.0071
Error-z(cm)	1.1866	0.4246	0.0935

C. Calibration and reprojection

Firstly, we prepared a calibration board, as is shown in Figure 2d, which contained four identifiable markers around the board. Then the hand-held calibration board was placed in several positions with different poses in front of the sensors system. And several groups of images and corresponding point clouds in each position were captured. The sensors collected images and point clouds data in 6 different poses in the experiment.

In the experiment of Livox MID-70, when the threshold ξ in Ransac [21] was set to 4, 202 pairs of inner points were selected among 6 groups of data, and the average RMS was 1.687. When the threshold ξ was set to 0.2, 5 pairs of inner points were selected, and the average RMS reached 0.123. Table I shows RMS decreased as ξ decreased in different devices.

Table II shows the comparison of the statistical quantities of our method in Livox and others. When the Ransac threshold ξ was set to 4, the average RMS of our approach was the lowest, while most of the corresponding point RMS were between 1 pixel to 5 pixels. The average RMS of L515 reached 0.04 when ξ was 0.1, and nearly 53.4% of points were lower than 1 pixel in RMS. Compared with Zhu [7], our algorithm does not require excessive geometric features in the environment and can still work in the vast outdoors (e.g., playgrounds, skies, etc.). In addition, our algorithm can work robustly at closer distances by taking multiple sets of calibration boards in comparison with Cui [11], which can be computed effectively in the FOV of the camera and Lidar by changing different sensor poses.

The following Figure 11 shows that the image of the actual scene is reprojected into point clouds, and the calibration result is better in detail. We prove that calibration results at close distances are equally applicable at long distances. More results are shown in the video on Github.

D. 3D trajectory comparison with built-in extrinsic

We compared three different calibration frameworks with built-in extrinsic in 3D trajectory estimation with Intel Realsense L515. Built-in extrinsic was regarded as ground truth to validate the results estimated by other calibration frameworks. A4LidarTag was used to estimate sensor pose in different positions. The Euclidean distance between x,y, and z-axis were computed as shown in Table III. Compared with Zhu [7] and Cui [11], the error in the 3D pose estimation of our calibration framework was the lowest. Figure 12 is the 3D trajectory plotted by experiment results.

V. Conclusion

In this paper, a close-range target-based extrinsic calibration framework for the visual-Lidar SLAM system has been proposed. Depth-based A4LidarTag, composed of simple geometric holes, has been designed. And the same recognition algorithm is developed in both image and the depth image of Lidar. Unlike other extrinsic calibration frameworks, our algorithm is easy to be implemented but efficient. Besides, only an A4-size calibration board is required rather than complicated requirements. After experiments in both indoor and outdoor scenes, A4LidarTag performs well in both images and depth images, significantly reducing the calibration board's size and working distance while improving the calibration accuracy.

References

[1] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "Orb-slam: A versatile and accurate monocular slam system," IEEE Transactions on Robotics, vol. 31, no. 5, pp. 1147–1163, 2015. I

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

[3] J. Engel, T. Schöps, and D. Cremers, "Lsd-slam: Large-scale direct monocular slam," in ECCV, 2014. I

[4] K. Tateno, F. Tombari, I. Laina, and N. Navab, "Cnn-slam: Real-time dense monocular slam with learned depth prediction," in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 6565–6574. I

[5] J. Zhang and S. Singh, "Loam: Lidar odometry and mapping in real-time," in Robotics: Science and Systems, 2014. I

[6] J. Lin and F. Zhang, "Loam livox: A fast, robust, high-precision lidar odometry and mapping package for lidars of small fov," 2020 IEEE International Conference on Robotics and Automation (ICRA), pp. 3126–3131, 2020. I, I

[7] Y. Zhu, C. Zheng, C. Yuan, X. Huang, and X. Hong, "Camvox: A low-cost and accurate lidar-assisted visual slam system," 2021 IEEE International Conference on Robotics and Automation (ICRA), pp. 5049–5055, 2021. I, I, II, II, III, IV-C, IV-D

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

[9] J. Lin, C. Zheng, W. Xu, and F. Zhang, "R^2live: A robust, real-time, lidar-inertial-visual tightly-coupled state estimator and mapping," IEEE Robotics and Automation Letters, vol. 6, pp. 7469–7476, 2021. I

[10] J. Graeter, A. Wilczynski, and M. Lauer, "Limo: Lidar-monocular visual odometry," in 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2018, pp. 7872–7879. I, I

[11] J. Cui, J. Niu, Z. Ouyang, Y. Q. He, and D. Liu, "Acsc: Automatic calibration for non-repetitive scanning solid-state lidar and camera systems," ArXiv, vol. abs/2011.08516, 2020. 2, I, II, II, II, III, IV-C, IV-D

[12] C. Yuan, X. Liu, X. Hong, and F. Zhang, "Pixel-level extrinsic self calibration of high resolution lidar and camera in targetless environments," IEEE Robotics and Automation Letters, vol. 6, pp. 7517–7524, 2021. 2, I, II

[13] C. Fang, S. Ding, Z. Dong, H. Li, S. Zhu, and P. Tan, "Single-shot is enough: Panoramic infrastructure based calibration of multiple cameras and 3d lidars," ArXiv, vol. abs/2103.12941, 2021. 2

[14] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in Conference on Computer Vision and Pattern Recognition (CVPR), 2012. I

[15] W. Wang, K. Sakurada, and N. Kawaguchi, "Reflectance intensity assisted automatic and accurate extrinsic calibration of 3d lidar and panoramic camera using a printed chessboard," ArXiv, vol. abs/1708.05514, 2017. I, II

[16] R. Gomez-Ojeda, J. Briales, E. Fernández-Moral, and J. González, "Extrinsic calibration of a 2d laser-rangefinder and a camera based on scene corners," 2015 IEEE International Conference on Robotics and Automation (ICRA), pp. 3611–3616, 2015. I, II

[17] A. Dhall, K. Chelani, V. Radhakrishnan, and K. M. Krishna, "Lidar-camera calibration using 3d-3d point correspondences," ArXiv, vol. abs/1705.09785, 2017. I

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

[19] J. Huang and J. W. Grizzle, "Improvements to target-based 3d lidar to camera calibration," CoRR, vol. abs/1910.03126, 2019. [Online]. Available: http://arxiv.org/abs/1910.03126 I, II

[20] V. Lepetit, F. Moreno-Noguer, and P. Fua, "Epnp: An accurate o(n) solution to the pnp problem," Int. J. Comput. Vision, vol. 81, no. 2, p. 155–166, Feb. 2009. [Online]. Available: https://doi.org/10.1007/s11263-008-0152-6 I, 5, III-C.3

[21] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," Commun. ACM, vol. 24, no. 6, p. 381–395, Jun. 1981. [Online]. Available: https://doi.org/10.1145/358669.358692 I, 5, III-C.3, IV-C

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

[23] L. Calvet, P. Gurdjos, C. Griwodz, and S. Gasparini, "Detection and accurate localization of circular fiducials under highly challenging conditions," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 562–570. 4

[24] F. Bergamasco, A. Albarelli, and A. Torsello, "Pi-tag: A fast image-space marker design based on projective invariants," Machine Vision and Applications, vol. 24, 08 2013. 4

[25] S. Garrido-Jurado, R. Muñoz-Salinas, F. Madrid-Cuevas, and M. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," Pattern Recognition, vol. 47, no. 6, pp. 2280–2292, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031320314000235 4, III-B

[26] J.-K. Huang, M. G. Jadidi, R. Hartley, L. Gan, R. M. Eustice, and J. W. Grizzle, "Lidartag: A real-time fiducial tag using point clouds," ArXiv, vol. abs/1908.10349, 2019. II

[27] P. Virtanen and Gommers, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," Nature Methods, vol. 17, pp. 261–272, 2020. III-A

[28] M. Norouzi, D. J. Fleet, and R. Salakhutdinov, "Hamming distance metric learning," in Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1, ser. NIPS'12. Red Hook, NY, USA: Curran Associates Inc., 2012, p. 1061–1069. III-B.1, III-B.4

[29] G. Nagy, "Robot vision (berthold klaus paul horn)," SIAM Review, vol. 30, no. 1, pp. 150–152, 1988. [Online]. Available: https://doi.org/10.1137/1030032 III-B.2

[30] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg et al., "Scikit-learn: Machine learning in python," Journal of machine learning research, vol. 12, no. Oct, pp. 2825–2830, 2011. III-B.2

[31] S. Gillies et al., "Shapely: manipulation and analysis of geometric objects," toblerity.org, 2007–. [Online]. Available: https://github.com/Toblerity/Shapely III-B.3

[32] C. B. Sullivan and A. Kaszynski, "PyVista: 3d plotting and mesh analysis through a streamlined interface for the visualization toolkit (VTK)," Journal of Open Source Software, vol. 4, no. 37, p. 1450, may 2019. [Online]. Available: https://doi.org/10.21105/joss.01450 11

[33] Z. Zhang, "Flexible camera calibration by viewing a plane from unknown orientations," in Proceedings of the Seventh IEEE International Conference on Computer Vision, vol. 1, 1999, pp. 666–673 vol.1. IV