An adaptive overlap-based ICP algorithm for multi-LIDAR calibration in low-overlap situations

Hanlin Li
School of Advanced Technology
Xi'an Jiaotong-Liverpool University
Suzhou 215123, China
Hanlin.Li21@student.xjtlu.edu.cn

Huajun Yuan
School of Advanced Technology
Xi'an Jiaotong-Liverpool University
Suzhou 215123, China
Huajun.Yuan21@student.xjtlu.edu.cn

Haocheng Zhao
Department of Artificial Intelligence
Institute of Deep Perception Technology, JITRI
Wuxi 214000, China
Haocheng.Zhao19@student.xjtlu.edu.cn

Bin Zou
Department of Research and Development
Suzhou Inteleizhen Intelligent Technology Co., Ltd
Suzhou 215123, China
dushuang@irobotop.com

*Limin Yu
Department of Communications and Networking
Xi'an Jiaotong-Liverpool University
Suzhou 215123, China
limin.yu@xjtlu.edu.cn

*Xinheng Wang
Department of Mechatronics and Robotics
Xi'an Jiaotong-Liverpool University
Suzhou 215123, China
Xinheng.Wang@xjtlu.edu.cn

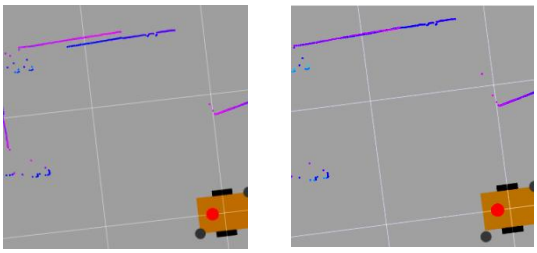
***Abstract*—LiDAR has been widely used as a core sensor in autonomous driving and mobile robotics. Due to the high price of spinning LiDARs, the industry always uses a combination of multiple calibrated solid-state LiDARs. However, there is usually only a small overlap FoV between LiDARs on account of their mounting positions. In such a low-overlap situation, conventional calibration methods always bring catastrophic results. To solve this problem, this paper proposed an adaptive overlap-based ICP algorithm, which can calibrate both 2D and 3D LiDARs. The algorithm adaptively extracts the overlapping point clouds by identifying the correct matching results and constructs an optimized ICP alignment. By controlling the radius range of the point cloud search, our method allows for quick extrinsic calibration of two LiDARs with low-overlap FoV. Experiments show that our algorithm achieves a more robust and accurate low-overlap registration than other methods, and also has speed advantages in general situations.**

Keywords—2D LiDAR, 3D LiDAR, overlap-based, ICP, multi-LiDAR calibration, point cloud registration

I. INTRODUCTION

The rise of new energy vehicles and Unmanned Aerial Vehicles (UAV) has turned Simultaneous Localization and Mapping (SLAM) into a hot research topic. Light Detection and Ranging (LiDAR), also known as Laser Radar, plays a key role in SLAM. It is capable of providing relative angle and sub-centimeter distance measurements for features in the environment [1]. Currently, LiDAR is divided into 360° mechanical spinning LiDAR and solid-state LiDAR. The former one detects better but is extremely high cost. The latter one is less expensive but has a smaller Field of View (FoV). To save cost, the most common solution in industry is to combine multiple solid-state LiDARs to achieve the same observation as a 360° LiDAR.

On the other hand, there are always errors during the manufacturing and mounting of LiDARs. The accumulation of these errors will cause the combined point cloud to fail to close [see Fig. 1(a)]. The original point clouds cannot reflect reality, which will reduce the accuracy of the localization and mapping part of SLAM. Therefore, we expect to obtain a more consistent point cloud by pre-calibrating LiDARs [see Fig. 1(b)].



(a)Input Point Cloud (b)Estimated Point Cloud

Fig. 1. The input and estimated point cloud

Most of the open-source LiDAR calibration methods [2-9] align the full point clouds to obtain the transformation matrix. These approaches only work when the point clouds have a large number of corresponding point pairs. However, the number of point clouds in a single scan of a solid-state LiDAR is only 5%-15% of that of a spinning LiDAR. This results in very few corresponding point pairs in solid-state LiDARs. In this case, those conventional calibration methods often fail to achieve the alignment.

In this paper, we came up with an adaptive overlap-based Iterative Closest Point (ICP) method and applied it to calibrate LiDARs with low-overlap FoVs. The point clouds within the overlapping FoV are used as the input of ICP. Our overlap filter adjusts the size of the overlapping region automatically by determining whether the last matching result was correct. After overlap filtering and downsampling filtering, only 30%-50% of the original point cloud remains. Then, our KD-tree optimized ICP method aligns the processed point clouds to get a new transformation. The above steps are repeated until global convergence. Compared with others, our method takes less time but achieves more accurate alignments in each iteration. Experimental results show that this method reduces the elapsed time by 8%-32% when calibrating solid-state LiDARs with low-overlap FoV.

The paper is organized as follows. Section II describes other related work. Section III details the adaptive overlap-based ICP algorithm. Afterwards, Section IV illustrates the experimental setup, and provides a comprehensive analysis

* Corresponding author.

978-1-6654-9144-0/22/$31.00 ©2022 IEEE

of the test results. The conclusion and further discussion are given in Section V.

II. Related Work

A. Common Calibration

Point cloud registration methods ICP [10] and NDT [11] are mainly used as the underlying algorithms for the calibration of multiple LiDARs. Normal Distributions Transform (NDT) was employed in [2] and provided an excellent code organization as well as a test environment. In [3], ICP was applied to match each single frame of point cloud data. [3] also provided calibration between LiDAR and other sensors e.g. camera. By finding the correspondence between the planar surfaces, calibration was also achieved in [4] without the use of ICP or NDT. Meanwhile, a new online calibration technology based on Lidar Odometry and Mapping [5-6] (LOAM) was proposed in [7]. The authors in [8] used this technique to provide initial values for ICP matching. In contrast, [9] used each LiDAR to construct the LOAM maps separately and then aligned maps with ICP.

NDT assumes that the point cloud is normally distributed within the grid, which is contrary to objective facts. Usually, ICP is faster and more accurate than NDT. M. Magnusson et.al in [12] compared ICP and NDT. They concluded that ICP has better performance in aligning structured point clouds. As a result, our method uses ICP as the underlying algorithm.

B. Low-overlap Calibration

The approaches in [2-9] strongly required a wide common FoV between each pair of sensors. The larger the overlap area is, the more point-to-point constraints can be used for calibration. If the overlapping FoV between two LiDARs falls below 30%, using these algorithms can lead to rapidly deteriorating results [13]. Therefore, a low-overlap LiDAR calibration method is needed to satisfy our solid-state LiDARs' situation.

To address the above problem, a low-overlap point cloud registration model was proposed in [13]. However, it is not suitable for mobile robots because it uses deep learning and has high hardware requirements. X. Liu and F. Zhang from Hong Kong University offered a targetless calibration method to solve the low overlap FoV problem by rotating the vehicle 360° [14]. Their team also added LiDAR-camera calibration based on plane features in 2022 [15]. But their method still requires LOAM to provide the initial pose and LOAM does not support 2D LiDAR. We expect to process the raw data for both 2D and 3D calibration. OpalsICP [16-17]matched a fixed range of point clouds directly. The point clouds were not processed and were prone to failure in case of poor initial poses. And KD-tree radius search in our algorithm outperforms voxel hulls in terms of speed and accuracy. [18] filtered a certain percentage of false matches according to the overlap ratio. But it would consume more time than processing the input directly. Unfortunately, opalsicp only had MatLab implementation for images and [18] had no open-source code. The proposed algorithm and these two cannot be compared in the ROS environment.

III. Methodology

A. Overview

The workflow of our proposed system is demonstrated in Figure .2. Normally, the point clouds from LiDARs need a coarse global registration to obtain an initial position before the fine local registration. We define H_0 as the result of coarse alignment and the initial value. This value does not need to be very accurate and can come from 4-Points Congruent Sets (4PCS) [19] or NDT or a Hand-Eye calibration. The difference between the initial pose and true value can be within approximate $0.4m$ and $0.15rad$.

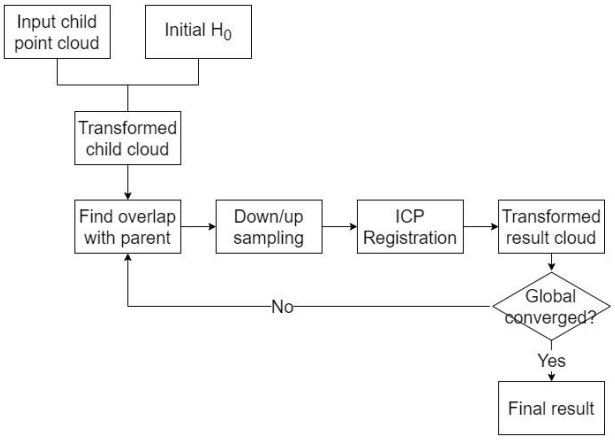


Fig. 2. Flow chart of the algorithm

The geometric centre of the parent LiDAR L_1 is set as the origin of the 3D coordinate system S. In S, the input displacement and Euler angle are defined as (x_0,y_0,z_0) and (α,β,γ). The rotation matrix R_0 is:

$$R_0 = R_z(\gamma) * R_y(\beta) * R_x(\alpha) \tag{1}$$

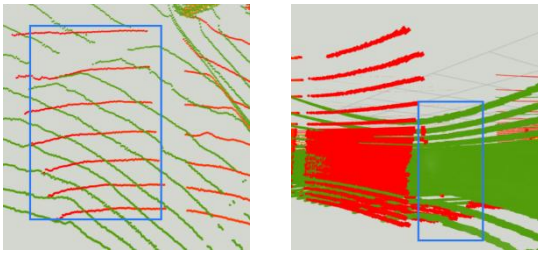
And the displacement matrix T_0 converted from the input:

$$T_0 = \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} \tag{2}$$

H_0 is a 4*4 homogeneous transformation matrix in our calculation and is used to describe the initial relationship between the parent LiDAR L_1 and the child L_2.

$$H_0 = \begin{bmatrix} R_0 & T_0 \\ 0 & 1 \end{bmatrix} \tag{3}$$

It will be very time-consuming to use LiDAR input directly for registration. Not only due to its huge amount of data, but the raw data contains a large amount of invalid information that interferes with the alignment leading to alignment failure, i.e. flat surface ripples [Fig. 3(a)] and derivative planes of edges [Fig. 3(b)].



(a) (b)

Fig. 3. Invalid point clouds are circled in blue.

Direct ICP cannot work as we expect. As a result, the data should be pre-processing (see Figure. 2). According to H_0, the transformed child point cloud P from LiDAR L_2 at time 0 is

$$P_{2,0}{}' = H_0 * \begin{bmatrix} P_{2,0} \\ 1 \end{bmatrix} \tag{4}$$

Based on Figure. 2, $O_{1,k}$ and $O_{2,k}$ are the overlapping parts of the parent LiDAR L_1 and the child L_2 respectively in time k:

$$O_{1,k} = \left\{ \forall p_1 \in P_{1,k} : \exists p_2{}' \in P_{2,k}{}', \sqrt{\sum_{m=1}^{n} \left(p_{1m} - p_{2m}{}' \right)^2} < \lambda \right\}$$
$$O_{2,k} = \left\{ \forall p_2{}' \in P_{2,k}{}' : \exists p_1 \in P_{1,k}, \sqrt{\sum_{m=1}^{n} \left(p_{2m}{}' - p_{1m} \right)^2} < \lambda \right\} \tag{5}$$

Where λ is an adaptive adjustment threshold based on convergence or not. After overlap extraction, we up-sample or down-sample the $O_{1,k}$ and $O_{2,k}$ according to their size (Figure. 2). The up-sampling method used here is linear interpolation and the down-sampling is voxel grid filter.

In general, the size of point clouds $O_{1,k}$ and $O_{2,k}$ would not be enormous (Especially in the situation with 2D LiDAR). Depending on LiDARs' extrinsic angle, overlap extraction can reduce the number of raw point clouds $P_{1,k}$ and $P_{2,k}$' by 50% - 90%. Voxel grid filter can reduce unfiltered $O_{1,k}$ and $O_{2,k}$ to 7% - 43%. Only a small portion of point clouds would be involved in registration, which greatly improves the speed of the ICP algorithm.

As shown in Fig. 2, a point-to-point ICP alignment on $O_{1,k}$ and $O_{2,k}$ is to generate a new transformed child point cloud. Then do another overlap extraction and ICP registration until global converged.

In 2D point cloud matching, due to the small number of point clouds, multiple iterations tend to deviate the result from the true value. So we use a fixed matrix H to assume a constant overlap area or control iteration number to rectify the aligning result.

B. Overlap Extraction

The threshold λ varies with whether the last ICP match was correct. When the result tends to the true value, the point clouds overlap more and a smaller λ is needed to reduce interference from other points except for the corresponding ones. When the result differs significantly from the true value, the point cloud is more dispersed and a larger λ is required to expand the range of selected points to make a larger alignment. Consequently, the size of $O_{1,k}$ and $O_{2,k}$ are used to balance λ:

- When the size of $O_{1,k}$, $O_{2,k}$ obtained in this iteration is larger than that in the previous iteration. We believe the point clouds are converging at time k. λ decreases by the decreasing factor m.
- When the size of $O_{1,k}$, $O_{2,k}$ obtained in this iteration is less than that in the previous iteration. The point clouds may be caught in a false registration at time k. So λ increases by the increasing factor n.

The ideal result of each iteration is that the point clouds tend to converge, so the ICP inputs need to be selected from large to small. Usually, n is 30% - 50% of m.

Set another threshold T, for a larger range of coarse alignments at the beginning of the match. This can also speed up the algorithm. Besides, KD-tree is constructed for $P_{1,k}$, $P_{2,k}$' separately. It is possible to lower the algorithmic complexity of this step from $O(n^2)$ to $O(\log^2 n)$.

The complete algorithm for extracting overlap is summarized in Algorithm. 1.

Algorithm 1: Overlap Extraction

Input: LiDAR scans $P_{1,k}$, $P_{2,k}$, threshold λ and T, transformation matrix H_k
Output: overlapped point cloud $O_{1,k}$, $O_{2,k}$

```
P'_{2,k} is matrix transformed from P_{2,k} by H_k;
Create KD-tree for P_{1,k};
for p' in P'_{2,k} do
    Do radiusSearch(p', λ, result);
    if result exists then
        Add p' into O_{2,k};
    end
end
if λ < T then
    create KD-tree for P'_{2,k};
    for p in P_{1,k} do
        Do radiusSearch(p, λ, result);
        if result exists then
            Add p into O_{1,k};
        end
    end
else
    O_{1,k} = P_{1,k};
end
return O_{1,k}, O_{2,k};
```

C. Point Cloud Registration

The input of the registration phase is $O_{1,k}$, $O_{2,k}$. ICP is used to compute the rigid body transformation that minimizes the average Euclidean distance between the point clouds. The various factors affecting the ICP algorithm are given in detail in [20]. Inspired by [20], the steps of our optimized ICP algorithm are as follows.

- Select suitable point clouds for alignment. We up/down-sample $O_{1,k}$, $O_{2,k}$ to get $O_{1,k}'$, $O_{2,k}'$, and use RANSAC [21] outlier rejection.
- Find the closest point in $O_{1,k}'$ corresponding to the point in $O_{2,k}'$. KD-tree is used to accelerate the search.
- Weighting the corresponding point pairs.
- According to current λ, discard pairs of points whose distance exceeds the threshold. And discard vertices that are at the boundary of the object.
- Point-to-point error function is used.

Since $O_{1,k}$, $O_{2,k}$ already have a high degree of overlap, convergence is achieved in only 3-5 iterations. Normally, we set maximum iterations of ICP < 10.

IV. Experiments and Results

The algorithm proposed in this paper has been implemented on both 2D and 3D LiDARs. Therefore we have built different platforms, including a customized multi-LiDAR vehicle with 2D LiDARs [Fig. 4(a)] and a virtual environment on a PC. The 2D LiDARs were tested in both

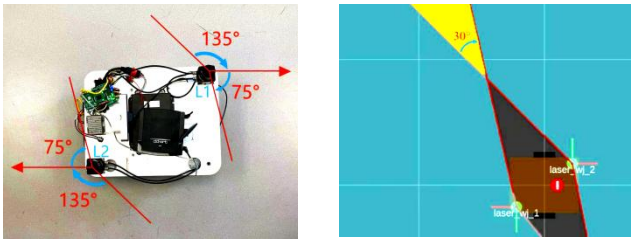
real and virtual environments. The 3D LiDARs experiments were conducted only in the virtual machine.

The virtual environment is described below. VMware builds Ubuntu 20.04 as our experimental OS. The CPU of the PC is intel-i7 7700HQ and the memory is dual-channel 2×8G 2400MHz. The environment contains ROS-noetic, PCL 1.10 and Eigen.

The algorithm in this paper is compared with other methods on all the mentioned platforms.

A. 2D-LiDAR Calibration Platform

Our customized multi-LiDAR platform is equipped with intel-i7 7600U, 16GB memory and two VanJee-716 2D solid-state LiDARs. Each LiDAR with a resolution of 0.33°, a scan rate of 15Hz and was set 210 degrees of FoV [see L_1, L_2 in Fig. 4(a)]. The arrow indicates the direction of LiDAR mounting. As shown in the picture [Fig. 4(b)], the blue region represents the area that was scanned only by one LiDAR, while the yellow area represents the scan overlap which is only 30°. Furthermore, LiDARs cannot detect the same obstacle within less than 90 cm.



(a) (b)

Fig. 4. Our customized multi-LiDAR platform

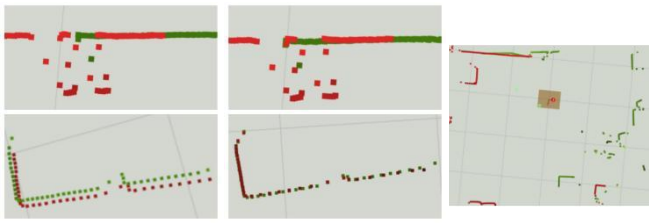
The vehicle did human-controlled Cartographer [22] mapping in a carton-built simulation environment. The map resolution is 0.005 (m). At the beginning of the mapping, we measured the distance between the centre of the vehicle and the nearest carton obstacle. It was defined as D_{start}. Also at the end of the mapping, this distance was defined as D_{end}. In Cartographer's probabilistic grid map, each grid stores the probability that the grid is hit by a LiDAR. So the D in the grid map were defined as the distance between the centre of the vehicle and the nearest $P_{\text{occupied}} > 0.5$ point. D_{start} and D_{end} collected in different situations are listed in Table I.

TABLE I. MAPPING RESULTS AFTER CALIBRATION

	D_{start} (m)	D_{end} (m)
Ground-Truth	0.37	0.48
Grid Map by Pure LiDAR	0.3538	0.4557
Grid Map by Calibrated LiDAR	0.3541	0.4619
Grid Map by LiDARs+odom+IMU	0.3561	0.4573
Grid Map by Calibrated LiDARs+odom+IMU	0.3557	0.4662

The Ground-Truth is considered to be the true value to compare with the distance measured in the grid maps. Usually, the closer the D is to the true value, the more accurate the map is. The result shows the D_{end} from calibrated LiDAR is closer to the true value. But our method could not effectively optimize D_{start}. This is because our algorithm ultimately publishes a transformation message of L_2, while D_{start} is measured at the beginning by L_1 fixed on the vehicle as a reference LiDAR. To further demonstrate the superiority of our approach, we also measured D_{straight} after 1m of straight travel and D_{rotation} after 90 degrees of rotation. The difference between the uncalibrated result and calibrated result was similar to that of D_{end}. Therefore, it can be proved that using the algorithm proposed in this paper would have better performance in Cartographer mapping. Besides, multi-sensor fusion gives more accurate mapping results than using only LiDAR.

None of the algorithms mentioned above was designed for the 2D LiDAR low-overlap situations. Accordingly, a comparison with other algorithms in these cases is meaningless. Figure. 5 demonstrates graphically the original point cloud, the calibrated point cloud and the failure result from other algorithms. It is easy to distinguish by the naked eye that the calibrated point cloud is smoother and has less ghosting than the uncalibrated one.



(a)Original (b)After Calibration (c)Failure from NDT-based

Fig. 5. 2D LiDAR calibration results

B. 3D-LiDAR Calibration Platform

The dataset in [5] is used to simulate the sensor calibration of 3D LiDARs. It contains a total of 75 frames of 6 3D LiDARs with 180° FoV and a topic publishing frequency of 2*Hz*. The LiDAR named *front_high* and the LIDAR named *right_rear* have an overlap of only 65 degrees and have only approximately 12% similar point clouds. These two LiDARs were used to simulate a low-overlap FoV calibration in a 3D environment. While LIDAR *front_high* and LIDAR *left_front* have over 70% similar point clouds, which represents the calibration under general conditions.

The algorithm proposed in this paper is compared with other popular algorithms. A total of 3 evaluation metrics were introduced into the experiment. During experiments, we found that the proposed method has a speed advantage in general 3D point cloud registration.

The LiDAR poses given in dataset [5] are considered to be the true values. The 3D translation error e_t and the 3D rotation error e_r [16] is defined to evaluate the performance of the proposed algorithm and other algorithms. If rotation error e_r and translation error e_t are close to 0, it means an accurate result.

$$e_t = \|\Delta T\| = \sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2}$$
$$e_r = \arccos\left(\frac{trace(\Delta R) - 1}{2}\right) \tag{6}$$

The LiDAR calibration results for different algorithms are shown in Table II. The initial positions given in the experiment are the same, which differs from the true value by approximately 0.4*m* and 0.14*rad*.

TABLE II. 3D CALIBRATION RESULT BY DIFFERENT ALGORITHM

Method	Duration (s)		e_t (m)		e_r (°)	
	f_h,l_f	f_h,r_r	f_h,l_f	f_h,r_r	f_h,l_f	f_h,r_r
NDT-based	11.40	11.93	0.063	0.092	0.810	0.769
ICP-based	8.43	-	0.047	-	0.512	-
Proposed	7.70	4.13	0.046	0.070	0.444	0.692

Other ICP algorithm and NDT algorithm were selected as control groups. Table II shows that our algorithm is faster and more accurate than NDT under different conditions. When calibrating the high-overlap LiDAR *f_h* and *l_f*, our method gave essentially the same calibration results as ICP, but with a small speedup. In the low-overlap situation, the ICP-based algorithm had a high probability of failure even when a good initial value was given. We observed that the failures were caused by the wrong matchings of the corresponding points. While our points selection strategy could effectively circumvent these mismatches and quickly achieve low-overlap calibration of LiDAR.

The execution speed of our algorithm highly depends on the setting of the threshold λ. In dense point clouds like *f_h* and *l_f*, lowering λ can significantly reduce the time of one iteration, but the convergence rate may be too slow. In low overlap *f_h* and *r_r*, multiple attempts are required to determine the λ.

The data in Table II are obtained with threshold λ of 0.5 and decreasing factor *m* of 0.1.

V. CONCLUSION

In this paper, a LiDAR calibration technique based on an adaptive overlap ICP algorithm is presented. This approach adds the KD-tree search and a new overlap-based point pairs selection strategy to the traditional point-to-point ICP. By identifying the correct match to control the input point cloud range, our algorithm can be better applied in low-overlap situation and is insensitive to the initial value. We have verified the calibration accuracy and mapping consistency of the proposed method in both simulated and real environments. The experiments in Section IV show that the proposed algorithm can calibrate a dual LiDAR system with rotation and translation errors less than 0.4*m* and 0.15*rad*.

A bad threshold λ may lead to non-convergence of ICP matching. In future work, we hope to accomplish the optimal fetching of λ and a joint calibration with other sensors.

ACKNOWLEDGMENT

This work was supported by Suzhou Inteleizhen Intelligent Technology Co., Ltd. And we are grateful to Tongpo Zhang for providing guidance in academic writing.

REFERENCES

[1] C. Gao and J. R. Spletzer, "On-line calibration of multiple LIDARs on a mobile vehicle platform," in 2010 IEEE International Conference on Robotics and Automation, 2010, pp. 279 − 284. doi: 10.1109/ROBOT.2010.5509880.

[2] A. Shan, (2019). multi_lidar_calibration [v2.8.12]. *https://github.com/AbangLZU/multi_lidar_calibration*

[3] Incloon, (2022). calibration_kit [v2.3.12]. *https://github.com/calibtoolkit/calibration_kit*

[4] J. Jiao et al., A Novel Dual-Lidar Calibration Algorithm Using Planar Surfaces. 2019. doi: 10.1109/IVS.2019.8814136.

[5] J. Zhang and S. Singh, "LOAM : Lidar Odometry and Mapping in real-time," Robotics: Science and Systems Conference (RSS), pp. 109–111, Jan. 2014.

[6] J. Lin and F. Zhang, "Loam livox: A fast, robust, high-precision LiDAR odometry and mapping package for LiDARs of small FoV," in 2020 IEEE International Conference on Robotics and Automation (ICRA), 2020, pp. 3126 − 3131. doi: 10.1109/ICRA40945.2020.9197440.

[7] J. Jiao, H. Ye, Y. Zhu, and M. Liu, "Robust Odometry and Mapping for Multi-LiDAR Systems With Online Extrinsic Calibration," IEEE Transactions on Robotics, vol. 38, no. 1, pp. 351 − 371, 2022, doi: 10.1109/TRO.2021.3078287.

[8] Z. Gong, C. Wen, C. Wang, and J. Li, "A Target-Free Automatic Self-Calibration Approach for Multibeam Laser Scanners," IEEE Trans Instrum Meas, vol. 67, pp. 1 − 3, Jan. 2017, doi: 10.1109/TIM.2017.2757148.

[9] A. Xu, (2022). sensor_calibration [v3.5.1]. *https://github.com/Xujianhong123Allen/sensor_calibration*

[10] P. Besl and H. D. McKay, "A method for registration of 3-D shapes. IEEE Trans Pattern Anal Mach Intell," Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 14, pp. 239−256, Mar. 1992, doi: 10.1109/34.121791.

[11] P. Biber and W. Strasser, "The normal distributions transform: a new approach to laser scan matching," in Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453), 2003, vol. 3, pp. 2743−2748 vol.3. doi: 10.1109/IROS.2003.1249285.

[12] M. Magnusson, A. Nuchter, C. Lorken, A. Lilienthal, and J. Hertzberg, Evaluation of 3D Registration Reliability and Speed - A Comparison of ICP and NDT. 2009. doi: 10.1109/ROBOT.2009.5152538.

[13] S. Huang, Z. Gojcic, M. Usvyatsov, A. Wieser, and K. Schindler, "PREDATOR: Registration of 3D Point Clouds with Low Overlap," Nov. 2020, [Online]. Available: http://arxiv.org/abs/2011.13005

[14] X. Liu and F. Zhang, "Extrinsic Calibration of Multiple LiDARs of Small FoV in Targetless Environments," IEEE Robot Autom Lett, vol. 6, no. 2, pp. 2036–2043, 2021, doi: 10.1109/LRA.2021.3061387.

[15] X. Liu, C. Yuan, and F. Zhang, "Targetless Extrinsic Calibration of Multiple Small FoV LiDARs and Cameras Using Adaptive Voxelization," IEEE Trans Instrum Meas, vol. 71, pp. 1–12, 2022, doi: 10.1109/TIM.2022.3176889.

[16] Y. Chen and G. Medioni, "Object modeling by registration of multiple range images," in Proceedings. 1991 IEEE International Conference on Robotics and Automation, 1991, pp. 2724–2729 vol.3. doi: 10.1109/ROBOT.1991.132043.

[17] P. Glira, N. Pfeifer, C. Briese, and C. Ressl, "A Correspondence Framework for ALS Strip Adjustments based on Variants of the ICP Algorithm," Photogrammetrie - Fernerkundung - Geoinformation, vol. 2015, Aug. 2015, doi: 10.1127/pfg/2015/0270.

[18] S. Nobili, R. Scona, M. Caravagna, and M. Fallon, "Overlap-based ICP tuning for robust localization of a humanoid robot," in 2017 IEEE International Conference on Robotics and Automation (ICRA), 2017, pp. 4721–4728. doi: 10.1109/ICRA.2017.7989547.

[19] D. Aiger, N. J. Mitra, and D. Cohen-Or, "4-Points congruent sets for robust pairwise surface registration," ACM Trans Graph, vol. 27, no. 3, Aug. 2008, doi: 10.1145/1360612.1360684.

[20] S. Rusinkiewicz and M. Levoy, "Efficient variants of the ICP algorithm," in Proceedings Third International Conference on 3-D Digital Imaging and Modeling, 2001, pp. 145 − 152. doi: 10.1109/IM.2001.924423.

[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, 1981, doi: 10.1145/358669.358692.

[22] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-time loop closure in 2D LIDAR SLAM," in 2016 IEEE International Conference on Robotics and Automation (ICRA), 2016, pp. 1271 − 1278. doi: 10.1109/ICRA.2016.7487258.