A Point Cloud Registration Method for Antarctic Power Cabins Based on an Improved SAC-IA Algorithm

Junyao Lu
School of Automation
Southeast University
Nanjing, China
jylu@seu.edu.cn

Shuo Shan
School of Automation
Southeast University
Nanjing, China
shuoshan@seu.edu.cn

Kanjian Zhang
School of Automation
Southeast University
Nanjing, China
kjzhang@seu.edu.cn

Haikun Wei*
School of Automation
Southeast University
Nanjing, China
hkwei@seu.edu.cn

2024 5th International Conference on Machine Learning and Computer Application (ICMLCA) | 979-8-3315-3033-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICMLCA63499.2024.10754370

***Abstract*—This paper presents a point cloud registration method for Antarctic power cabins based on an improved SAC-IA algorithm. To address the issues of measurement errors in structured light scanning and edge structure damage caused by traditional mean depth filtering, as well as inaccurate random point pair selection and long computation times in the SAC-IA algorithm, the following work was conducted: First, an improved depth filtering algorithm was designed to better filter out depth noise while preserving edge information. Second, ISS keypoint detection and cosine similarity were used to match the best feature pairs, combined with the RANSAC method to remove incorrect matches. Experimental results show that the improved algorithm significantly optimizes the number of key points and matching pairs, greatly saves computational resources, and enhances registration accuracy and robustness.**

Keywords-SAC-IA algorithm; point cloud registration; depth filtering; ISS keypoint detection; cosine similarity

I. INTRODUCTION

The Antarctic power cabin provides electricity and thermal energy to polar research equipment during unattended periods[1]. Inside the power cabin, a patrol robot equipped with a structured light 3D camera captures point cloud data to generate a high-precision model of the power cabin. This model aids in detecting and locating equipment faults, preventing equipment downtime. Compared to images, 3D models offer intuitive visualization tools[2], especially with high-precision spatial geometry data. Utilizing known equipment models, these can be identified within the point cloud. Through regular inspections by the robot, when structural anomalies occur in the equipment, diagnostic information is generated via transformation matrices, helping inspectors better understand the operating conditions and potential issues of the power cabin.

Due to the limited field of view of the camera, it is necessary to register point clouds captured from multiple viewpoints to create a complete point cloud map. The registration process is generally divided into two stages: initial registration and fine registration. Initial registration obtains the initial transformation matrix through the correspondence of the same feature points in the point cloud, with commonly used features such as FPFH. In the fine registration stage, the point cloud is accurately aligned based on the initial transformation matrix through the correspondence of neighboring points[3]. Quickly finding the correct correspondences in the initial registration stage is crucial for the success of the registration; incorrect point correspondences can lead to the failure of subsequent fine registration and result in map misalignment. In the coarse registration stage, reference [4] uses LPPF to quickly identify and match point pair features, achieving initial point cloud alignment. Reference [5] proposes an end-to-end point cloud registration network from coarse to fine, which effectively reduces errors and improves registration efficiency when handling point cloud data with noise and occlusions.

However, existing methods for point cloud modeling in Antarctic power cabins have some limitations. Structured light camera point clouds are more suitable for close-range, high-resolution, and rapid measurement scenarios[6]. Due to inherent measurement errors in structured light spots, the point cloud contains noise, which affects feature calculation. Additionally, because a single frame of the point cloud contains numerous points, calculating the features and correspondences for initial registration is time-consuming. Moreover, due to the presence of many planes within the power cabin, the points on the planes have similar features, making it easy to obtain incorrect correspondences.

In response to the aforementioned issues, this paper first addresses point cloud preprocessing[7]. The depth noise magnitude in the point cloud captured by the structured light camera increases with distance, necessitating filtering and denoising procedures. Subsequently, during the initial point cloud registration stage, the inspection robot continuously captures point clouds while in motion, with approximately 50% overlap between adjacent point clouds. It is essential to select key feature points to establish accurate correspondences. The contributions of this paper are as follows:

To address the issue of traditional mean depth filtering damaging edge structures, an improved depth filtering method was implemented, which better filters out depth noise while preserving edge information.

To address the problems of inaccurate random point pair selection and long computation times in the traditional SAC-IA algorithm, a filtering process was implemented to remove incorrect matches, significantly improving the accuracy of coarse registration and enhancing the algorithm's time efficiency.

II. METHODOLOGY

A. Improving Depth Filtering

The point cloud data captured in the power cabin shows that depth camera data errors are mainly concentrated on the z-axis. The farther from the camera, the greater the data fluctuation, causing originally flat point clouds to appear wavy. Even bilateral filtering[8] does not perform well on distant walls. Ordinary mean depth filtering damages edge information. To address this issue, an improved depth filtering algorithm is designed, which preserves edge information while performing adaptive radius mean filtering. The comparison of depth filtering effects is shown in Fig. 1.

(a) original image (b) Improved depth filtering (c) Ordinary depth filtering

Figure 1. Depth filtering effect

The specific process of improved depth filtering is as follows: for each point P with coordinates (P_x, P_y, P_z), the filtering coefficient $Zoomfactor$, and filtering radius $R = P_z * Zoomfactor$. Using KD-Tree range search, find all points within the radius R centered on point P and record as $P_{total} = \sum_{i=1}^{n} P_i$, where n is the total number of points found. The coordinates of the points P_i found within the search radius are (P_{ix}, P_{iy}, P_{iz}). Calculate the vector $V_i = (P_x - P_{ix}, P_y - P_{iy}, P_z - P_{iz})$ from point P to the other points within the radius. Sum the vectors from point P to all other points within the radius to obtain $V_{total} = \sum_{i=1}^{n} V_i$. Set a threshold value; if the value of V_{total} exceeds the threshold, point P is considered an edge point, and its original coordinates are retained. Otherwise, correct the coordinates of point P by calculating the sum of the Z-axis coordinates of all points within the radius $P_{totalZ} = \sum_{i=1}^{n} P_{iz}$. Update the Z-axis coordinate of point P to $P_z' = P_{totalZ}/n$, the X-axis coordinate of point P to $P_x' = P_x * (P_z' / P_z)$, and the Y-axis coordinate of point P to $P_y' = P_y * (P_z' / P_z)$.

B. Improved SAC-IA Algorithm

1) Problems Existing in Traditional SAC-IA Algorithm:

The traditional SAC-IA algorithm[9] has several limitations in initial point cloud alignment[10]. It relies on randomly selected key points for feature matching, leading to uneven distribution or lack of representativeness, affecting accuracy and robustness. Moreover, it directly matches features, which is prone to producing many incorrect matches in noisy or structurally complex point clouds, reducing matching accuracy. Additionally, SAC-IA's high computational complexity significantly increases processing time for large-scale point cloud data, impacting efficiency and practicality.

2) Improved Scheme:

To address the uneven or unrepresentative key point distribution in traditional SAC-IA, ISS[11] keypoint detector evaluates the local geometric properties of each point in the input point cloud, selecting points with significant geometric features as key points. The steps are: For each point p_i in the point cloud, define a radius r to determine its neighborhood $N(p_i)$. Calculate the covariance matrix C according to Eq. (1), where $|N(p_i)|$ represents the number of points in the neighborhood, p_j is the coordinate vector of a point in the neighborhood, and $\bar{p} = 1 / |N(p_i)| \Sigma_{P_j \in N(p_i)} p_j$ is the centroid of the points in the neighborhood. Perform eigenvalue decomposition on the covariance matrix C to obtain the eigenvalues $\lambda 1$、$\lambda 2$、$\lambda 3$ (assuming $\lambda 1 \geq \lambda 2 \geq \lambda 3$) and their corresponding eigenvectors. Calculate the feature value ratios $Saliency_{21} = \lambda_2 / \lambda_1$ and $Saliency_{32} = \lambda_3 / \lambda_2$ to evaluate the local geometric characteristics of the point. Set thresholds to control the saliency of the selected key points. Among the selected key points, use a non-maximum suppression algorithm to remove redundant key points, ensuring a uniform spatial distribution of key points.

$$C = \frac{1}{|N(p_i)|} \Sigma_{P_j \in N(p_i)} (p_j - \bar{p})(p_j - \bar{p})^T \tag{1}$$

Calculate the FPFH feature descriptor for each selected key point according to Eq. (2), where SPFH is the point pair feature histogram, and $d(p, p_i)$ is the distance between point p and its neighboring point p_i.

$$FPFH(p) = \sum_{i=1}^{k} \frac{1}{d(p, p_i)} S\,PFH(p, p_i) \tag{2}$$

To address the problem of numerous incorrect matches in the traditional SAC-IA algorithm when dealing with complex and repetitive structures, cosine similarity is used to measure the similarity between feature vectors to find the best matching pairs. The RANSAC[12] method is then employed to remove incorrect matches, improving registration robustness. The specific steps are as follows: For each key point in the source point cloud, calculate the similarity between its FPFH feature descriptor and the FPFH feature descriptors of key points in the target point cloud using Eq. (3), where $x \cdot y$ denotes the dot product of the vectors, and $\|x\|$ and $\|y\|$ denote the norms (lengths) of the vectors. Retain the pair of feature points with the highest similarity and add them to the matching pair set. From the initial set of matching pairs, randomly select a subset S. Use the matching pairs in the selected subset S to fit the transformation model parameters, namely the rotation matrix R and the translation vector t. For each matching pair (p_i, q_i), calculate

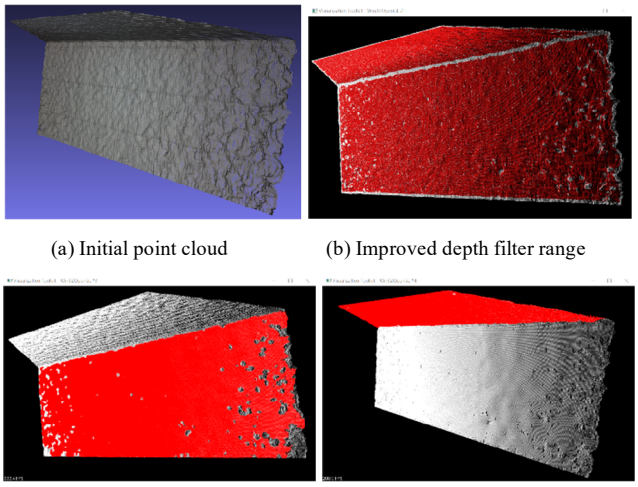
the distance between the source point p_i transformed by R and $t$and the target pointq_i, given by $distance(p_i, q_i) = \|Rp_i + t - q_i\|$. If this distance is less than the set threshold, the matching pair is considered an inlier. Repeat this calculation for all matching pairs to find the inlier set. Repeat the above process until the maximum number of iterations is reached. Choose the model with the most inliers as the final transformation model. The inlier set of the final transformation model is used as the filtered set of matching pairs.

$$CosineSimilarity(x, y) = \frac{x \cdot y}{\|\chi\|\|y\|} = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\Sigma_{i=1}^{n} x_i^2}\sqrt{\Sigma_{i=1}^{n} y_i^2}} \quad (3)$$

Finally, use the filtered set of matching pairs as the input for the SAC-IA algorithm to perform the initial registration.

III. Experimental Results And Analysis

A. Improved Depth Filtering Test



(a) Initial point cloud (b) Improved depth filter range

(c) Two fitted planes

Figure 2. The effect of the improved depth filtering

The experiment used data from the top wall of the power cabin to test the effects of the improved depth filtering, ordinary depth filtering, and bilateral filtering. The effect of the improved depth filtering is shown in Fig. 2. Plane segmentation was performed using the region growing segmentation algorithm (RGS) [13], and the segmented planes were fitted. The average error and variance of the point distances to the plane after filtering were calculated, as shown in Table I.

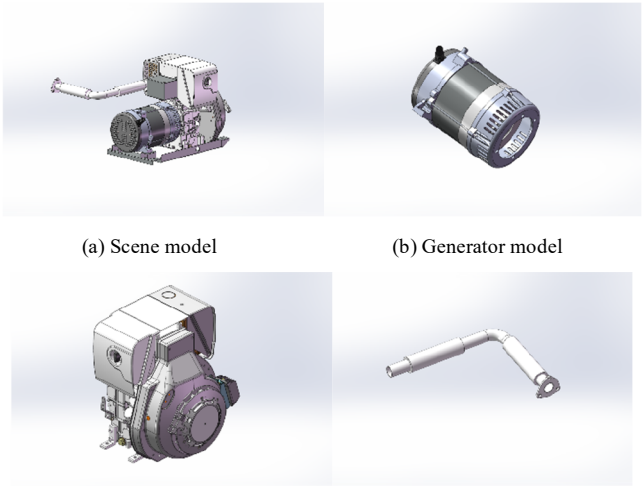
Data analysis shows that the improved depth filtering, compared to ordinary depth filtering, not only better preserves edge information but also slightly enhances the filtering effect on depth noise. When combined with bilateral filtering, the improved depth filtering maintains the filtering effect in areas with low depth noise and enhances the filtering effect in areas with high depth noise, compared to using bilateral filtering alone.

TABLE I. Filtering Result

Filtering Method	The Number of The Fitted Plane	Average Magnitude of Error	Error Variance
Ordinary depth filtering	1	3.62889	6.59092
	2	1.91052	2.75058
Improved depth filtering	1	3.59456	6.47503
	2	1.83556	2.66336
Bilateral filter	1	3.30896	6.03267
	2	1.70377	2.40803
Bilateral filtering + improved depth filtering	1	2.77452	3.74295
	2	1.90268	2.7412

B. Improved SAC-IA Algorithm Test

The test data were generated using the point cloud generation algorithm proposed in reference [14], including scene point clouds and model point clouds. The scene point cloud consists of three model point clouds, as shown in Fig. 3.



(a) Scene model (b) Generator model

(c) Diesel engine model (d) Exhaust pipe model

Figure 3. Test data

The relative positions of the scene point cloud and the model point cloud before registration are shown in Fig. 4.

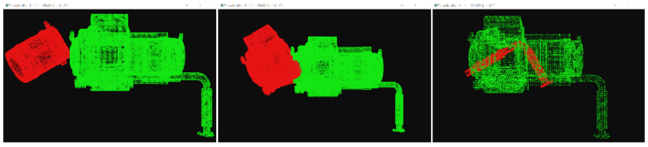


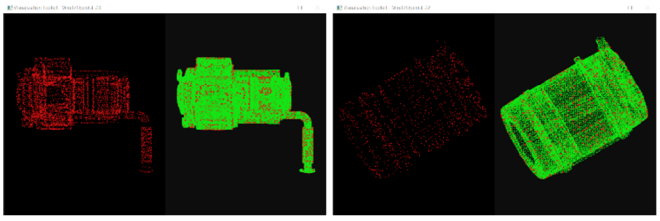
Figure 4. Relative position between point clouds

After ISS keypoint selection, the number of feature points was significantly reduced, retaining only the key points with significant geometric characteristics. Subsequently, the best feature pairs were matched using cosine similarity, and incorrect matches were filtered out using the RANSAC method, further reducing the number of feature points. The data in Table II clearly shows that the improved SAC-IA algorithm achieves significant optimization in the number of key points and matching pairs, greatly saving computational resources.

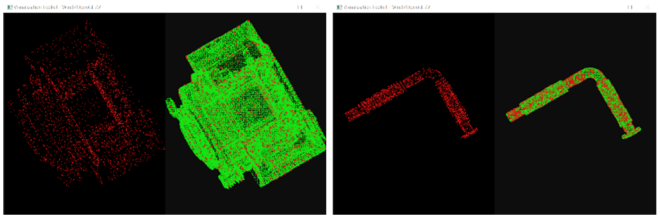
TABLE II. FILTERING RESULT

Type	Generator point cloud	Diesel engine point cloud	Exhaust pipe point cloud
Sampling grid size	2	2	1
Number of model point clouds	107740	266757	43722
Number of scene point clouds	403873	403873	546320
Model ISS keypoints	1130	2844	1669
Scene ISS keypoints	4573	4573	15797
Number of filtered matching pairs	120	340	73

The significant key points selected by the ISS keypoint detector are shown in Fig. 5.



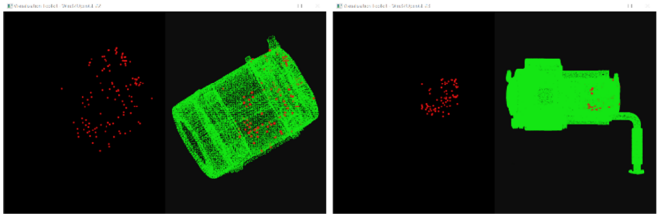
(a) Scene point cloud keypoints (b) Generator point cloud keypoint



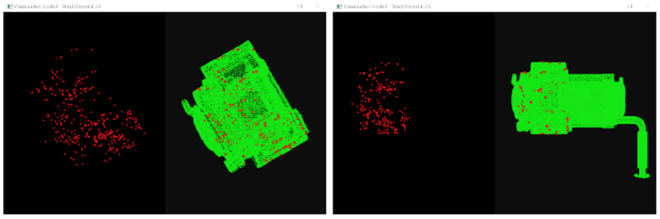
(c) Diesel engine point cloud keypoints (d) Exhaust pipe point cloud keypoints

Figure 5. ISS keypoints

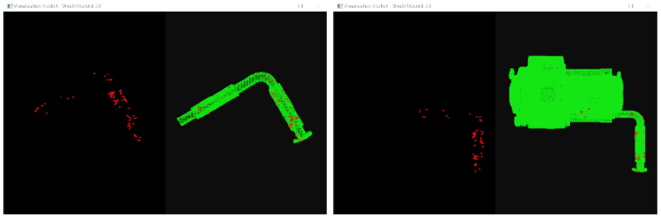
The best matching pairs found through cosine similarity comparison, and the inlier set obtained after removing incorrect matches using the RANSAC method, are shown in Fig. 6.



(a) Generator matching pairs



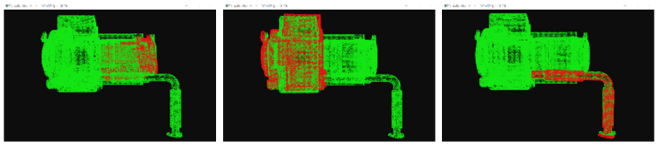
(b) Diesel engine matching pairs



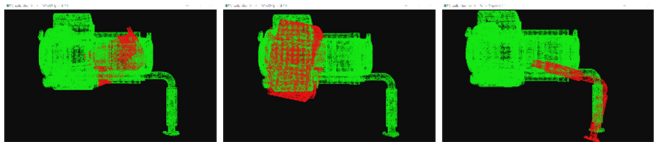
(c) Exhaust pipe matching pairs

Figure 6. Best matching pairs

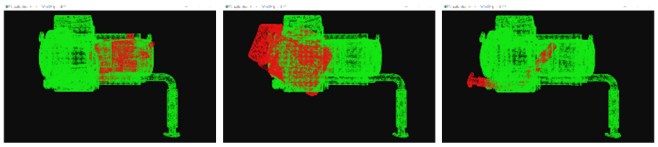
The registration results of the improved SAC-IA, ISS+SAC-IA, and ordinary SAC-IA methods are shown in Fig. 7.



(a) The improved SAC-IA results in this paper



(b) ISS+SAC-IA results



(c) Ordinary SAC-IA results

Figure 7. The registration results

The registration error in this paper is calculated by directly comparing the ground truth of the rigid transformation with the evaluated rigid transformation. The error is divided into rotational error e_R and translational error e_t, see Eq. (4) and Eq. (5), where (R_g, t_g) is the ground truth of the rigid transformation, and (R_m, t_m) is the estimated rigid transformation from the registration algorithm. The time taken by different registration algorithms and the resulting rotational and translational errors are shown in Table III.

$$e_R = arccos\frac{tr(R_g R_m^T - 1)}{2} \tag{4}$$

$$e_t = \|t_m - t_g\|_2 \tag{5}$$

The traditional SAC-IA algorithm takes a long time to compute and has a low registration success rate. By introducing ISS keypoint selection, the coarse registration time is significantly reduced, and the registration success rate is improved. The improved SAC-IA algorithm proposed in this paper combines ISS keypoint extraction with cosine similarity to find the best matching pairs and uses the RANSAC method to remove incorrect matches, further shortening the coarse

registration time and improving the matching accuracy and robustness. This improvement significantly enhances the efficiency and accuracy of the algorithm when processing large-scale point cloud data.

TABLE III. FILTERING RESULT

Registration Algorithm	Model Type	Time (s)	Error of Rotation	Error of Translation
This paper's improved SAC-IA algorithm	Generator	3.80835	0.267794	139.555
	Diesel engine	5.27396	0.139785	37.7432
	Exhaust pipe	3.44574	0.0849289	59.0043
ISS+SAC-IA	Generator	4.4766	0.433398	167.03
	Diesel engine	6.70199	0.235416	422.836
	Exhaust pipe	4.98509	0.315053	438.14
Ordinary SAC-IA	Generator	410.552	0.652892	216.632
	Diesel engine	1320.07	1.103	556.737
	Exhaust pipe	259.542	2.16351	778.101

IV. CONCLUSION

The improved depth filtering algorithm designed in this paper preserves edge information while maintaining or enhancing the filtering effect on depth noise. The proposed point cloud registration method for Antarctic power cabins, based on the improved SAC-IA algorithm, effectively addresses the issues of long computation times and low registration success rates in traditional SAC-IA algorithms by incorporating ISS keypoint selection, cosine similarity matching, and RANSAC filtering techniques. Experimental results show that the improved algorithm significantly reduces the number of key points and matching pairs, and significantly saves computation time and resources, greatly enhancing registration accuracy and robustness. This method demonstrates excellent efficiency and accuracy when processing large-scale point cloud data and holds significant application value.

ACKNOWLEDGMENT

This work was supported by the National Key Research and Development Program of China (Grant No. 2022YFC2807105), the Natural Science Foundation of Shenzhen (Grant No. JCYJ20210324121213036), and the Fundamental Research Funds for the Central Universities (2242024k30037, 2242024k30038).

REFERENCES

[1] X. Chen, K. Zhang, H. Wei, W. Zhu, "Design and Realization of the Temperature Control System for Unattended Antarctic Expedition Platform", Process Automation Instrumentation, vol. 34, no. 10, pp. 23–27, 2013.

[2] You H, Xu F, Du E. Robot-based real-time point cloud digital twin modeling in augmented reality[J]. Transforming Construction with Reality Capture Technologies, 2022.

[3] Besl P J, McKay N D. Method for registration of 3-D shapes[C]//Sensor fusion IV: control paradigms and data structures. Spie, 1992, 1611: 586-606.

[4] Yue X, Liu Z, Zhu J, et al. Coarse-fine point cloud registration based on local point-pair features and the iterative closest point algorithm[J]. Applied Intelligence, 2022, 52(11): 12569-12583.

[5] Hao R, Wei Z, He X, et al. Robust Point Cloud Registration Network for Complex Conditions[J]. Sensors, 2023, 23(24): 9837.

[6] Giancola S, Valenti M, Sala R. A survey on 3D cameras: Metrological comparison of time-of-flight, structured-light and active stereoscopy technologies[J]. 2018.

[7] Qing-guo T, Jin-tong L. Pre-processing of 3D scanning line point cloud data[C]//2010 International Conference on Computer Application and System Modeling (ICCASM 2010). IEEE, 2010, 10: V10-100-V10-104.

[8] Elad M. On the origin of the bilateral filter and ways to improve it[J]. IEEE Transactions on image processing, 2002, 11(10): 1141-1151.

[9] Rusu R B, Blodow N, Beetz M. Fast point feature histograms (FPFH) for 3D registration[C]//2009 IEEE international conference on robotics and automation. IEEE, 2009: 3212-3217.

[10] Wang L, Chen Y, Zong H, et al. A Point Cloud Registration Method for Substations Based on an Improved SAC-IA Algorithm[C]//2023 China Automation Congress (CAC). IEEE, 2023: 4622-4627.

[11] Zhong Y. Intrinsic shape signatures: A shape descriptor for 3D object recognition[C]//2009 IEEE 12th international conference on computer vision workshops, ICCV Workshops. IEEE, 2009: 689-696.

[12] Li J, Hu Q, Ai M. Point cloud registration based on one-point ransac and scale-annealing biweight estimation[J]. IEEE Transactions on Geoscience and Remote Sensing, 2021, 59(11): 9716-9729.

[13] Poux F, Mattes C, Selman Z, et al. Automatic region-growing system for the segmentation of large point clouds[J]. Automation in Construction, 2022, 138: 104250.

[14] Ding J, Zong H, Zhou J, et al. Study on the automatic modeling method of 3D information model for substations[C]//2022 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC). IEEE, 2022: 275-283.