Centralized RANSAC-Based Point Cloud Registration With Fast Convergence and High Accuracy

Kuo-Liang Chung, *Senior Member, IEEE*, and Wei-Tai Chang

***Abstract*—For point cloud registration, the purpose of this article is to propose a novel centralized random sample consensus (RANSAC) (C-RANSAC) registration with fast convergence and high accuracy. In our algorithm, the novel contributions are, first, the proposal of a scale histogram-based outlier removal to delete outliers from the initial line vector set L for constructing a reduced line vector set $\mathbf{L}_{\mathbf{red}}$; second, the handshake cooperation between the host RANSAC (H-RANSAC) only working on L and the local RANSAC (LCL-RANSAC) only working on $\mathbf{L}_{\mathbf{red}}$; third, in each handshake process, after receiving the global registration solution and the global iteration number x_H from H-RANSAC, LCL-RANSAC uses the received global solution as the initial solution of the modified TEASER++ (M-TEASER++) method to calculate its first local registration solution. If the first local registration solution satisfies the global iteration number inheritance condition, LCL-RANSAC directly sends the accumulated iteration number, $x_H + 1$, and the first local solution back to H-RANSAC; otherwise, LCL-RANSAC iteratively refines its local solution using the M-TEASER++ method, and then sends the resultant local solution and the required local iteration number x_{LCL} to H-RANSAC for updating the global solution, the global iteration number to $x_H := x_H + x_{\mathrm{LCL}}$, and the global confidence level. Due to $|L_{\mathrm{red}}| \ll |L|$ and employing the global iteration number inheritance condition test into our algorithm, we have conducted extensive experiments on testing point cloud pairs to show the registration accuracy and execution time merits of our algorithm relative to the state-of-the-art methods.**

***Index Terms*—Execution time, line vector set, outlier removal, point cloud registration (PCR), random sample consensus (RANSAC), registration accuracy.**

I. Introduction

3-D scanning techniques, such as the light detection and ranging scanning technique and the structured-light 3-D scanner, can provide accurate geometry information and have been widely used in remote sensing [1], [2], [3], [4], [5] and 3-D vision [6], [7], [8]. To generate the whole scene that multiview point clouds can cover, solving the point cloud registration (PCR) problem between the source and target point cloud sets is a challenging but important task. PCR aims to determine the matrix transformation, including scaling, rotation, and translation that aligns the points, which consider noise and variations in the point data acquisition process, such that the differences between corresponding points in different scans can be minimized. To solve the PCR problem, techniques, such as the iterative closest point (ICP) based approach [9] and the random sample consensus (RANSAC) based approach [10] are commonly employed. Feng et al. [11] offered a systematic and comprehensive review of state-of-the-art approaches for remote sensing registration, and these approaches include the intensity-based, feature-based, optical flow-based, deep learning-based, and hybrid techniques.

A. Related Works

ICP alternates between the closest point search in the target point cloud and the distance minimization between the aligned source points and the nearest target points iteratively, where the quaternion method [12] is used to realize the implementation. However, ICP often converges to a local minimum, and is sensitive to the initial registration solution, outliers, missing data, and partial overlaps, resulting in slow convergence [13]. In addition, ICP cannot deal with the scale change problem between the point cloud pairs. To provide a better initial registration solution for ICP, a principal component analysis approach [14] was proposed. To better determine the aligned source points and the corresponding target points in ICP, the angular-invariant feature approach [15] and the curvature feature similarity approach [16] were proposed. The Super 4-points congruent set (4PCS) method [17] introduced a smart indexing mechanism to reduce the execution time required in the 4PCS method [18]. In the fast global registration method [19], three rigid correspondences are first selected to construct a rough inlier set. After that, the graduated nonconvexity (GNC) function [20], [21] and the Geman-McClure penalty function are combined to solve the transformation matrix using the Gauss–Newton method.

To alleviate the influence of outliers in ICP, the sparse ICP approach [22] was proposed by utilizing the sparsity-inducing penalty function [23] to adaptively assign weights to outliers. To improve the convergence speed of the iterative process used in ICP, the Anderson acceleration approach [24] was proposed to predict the next iteration based on the behavior of the previous iterations. To tackle the two drawbacks: small convergence

Manuscript received 30 October 2023; revised 16 January 2024; accepted 9 February 2024. Date of publication 13 February 2024; date of current version 4 March 2024. This work was supported by the Ministry of Science and Technology, Taiwan under Grant MOST-111-2221-E-011-126- MY3. *(Corresponding author: Kuo-Liang Chung.)*

The authors are with the Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology, Taipei 106335, Taiwan (e-mail: klchung01@gmail.com).

Digital Object Identifier 10.1109/JSTARS.2024.3365516

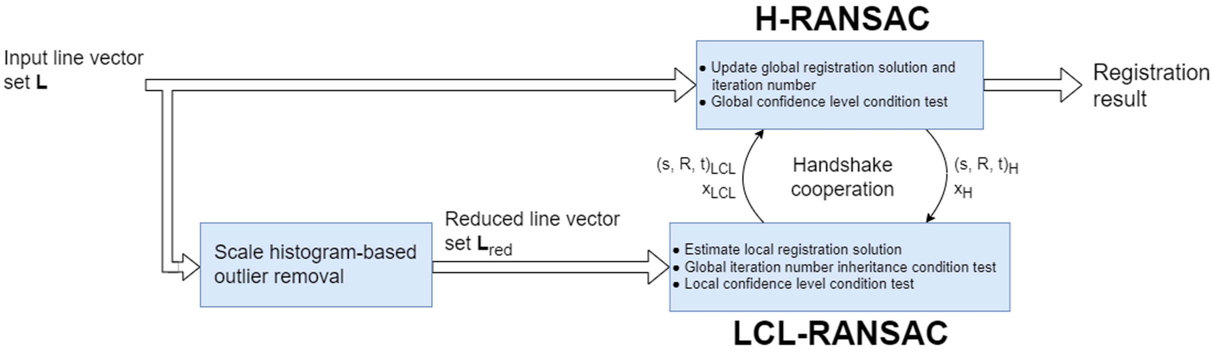
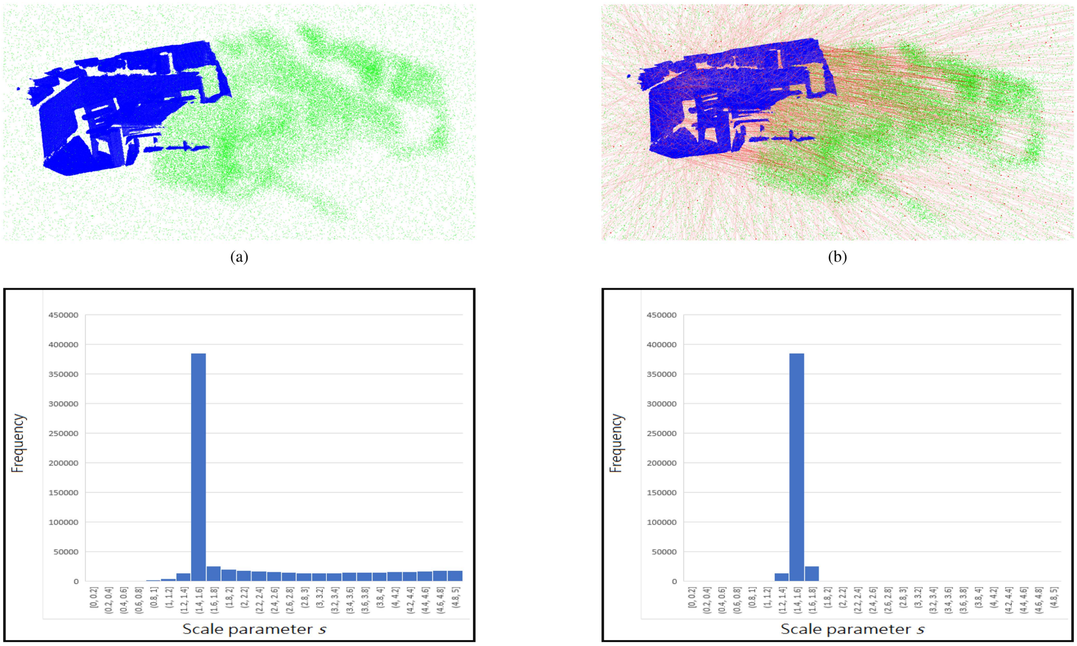

H-RANSAC
Input line vector set L
• Update global registration solution and iteration number
• Global confidence level condition test
Registration result
(s, R, t)LCL
xLCL
Handshake cooperation
(s, R, t)H
xH
Scale histogram-based outlier removal
Reduced line vector set Lred
• Estimate local registration solution
• Global iteration number inheritance condition test
• Local confidence level condition test
LCL-RANSAC

Fig. 1. Pipeline of our C-RANSAC registration algorithm.



(a)

(b)

(c)

(d)

Fig. 2. Scale histogram-based outlier removal for constructing the reduced line vector set $\mathbf{L}_{\text{red}}$. (a) Source point set $\mathbf{P}^s$ marked in blue and the target point set $\mathbf{P}^t$ marked in green. (b) Initial correspondence set $\mathbf{C}$ in which each correspondence is marked by a red line. (c) Scale histogram of the initial line vector set $\mathbf{L}$. (d) Scale histogram of $\mathbf{L}_{\text{red}}$.

basin and the sensitivity to outliers and partial overlaps, the robust symmetric ICP [25] was proposed by using a symmetric point-to-plane distance metric. In well-structured environments, the plane-based approach [26] was proposed to segment planes for registration. To improve the Sparse ICP method [22], the point-to-point registration problem is first transformed to a majorization–minimization problem [27]. Next, the Anderson acceleration approach was used to speed up its convergence. Then, a Welsch function-based error metric was proposed to boost the performance.

Unlike the ICP-related methods, RANSAC-based registration methods do not need an initial registration solution, but it does need an initial correspondence set to support the iterative hypothesis and verification (HAV) process via subset sampling trials. Due to achieving good registration robustness, the RANSAC approach for solving the PCR problem has received growing attention. However, the initial correspondence set is influenced by noise and outliers. In particular, when the outlier rate is high, it often leads to a huge number of subset sampling trials. The fast point feature histograms (FPFH) method [28] is used to extract feature descriptors of points. In FPFH, the initial correspondence set is constructed by comparing the similarity among these feature descriptors, and then, a RANSAC method, simply called FPFH+RANSAC, is applied to estimate the registration solution. Based on the deep learning framework combining ResNet [29] and UNet [30], the fully convolutional geometric feature (FCGF) [31] of each point is first extracted in a fast and memory-saving manner. Next, using these features, a reliable correspondence set is constructed, and then, a RANSAC method, simply called FCGF+RANSAC, is applied to estimate the registration solution. Experimental data demonstrated that the FCGF+RANSAC method is more accurate and faster than FPFH+RANSAC, 3DMatch [32], and PPFNet [33].

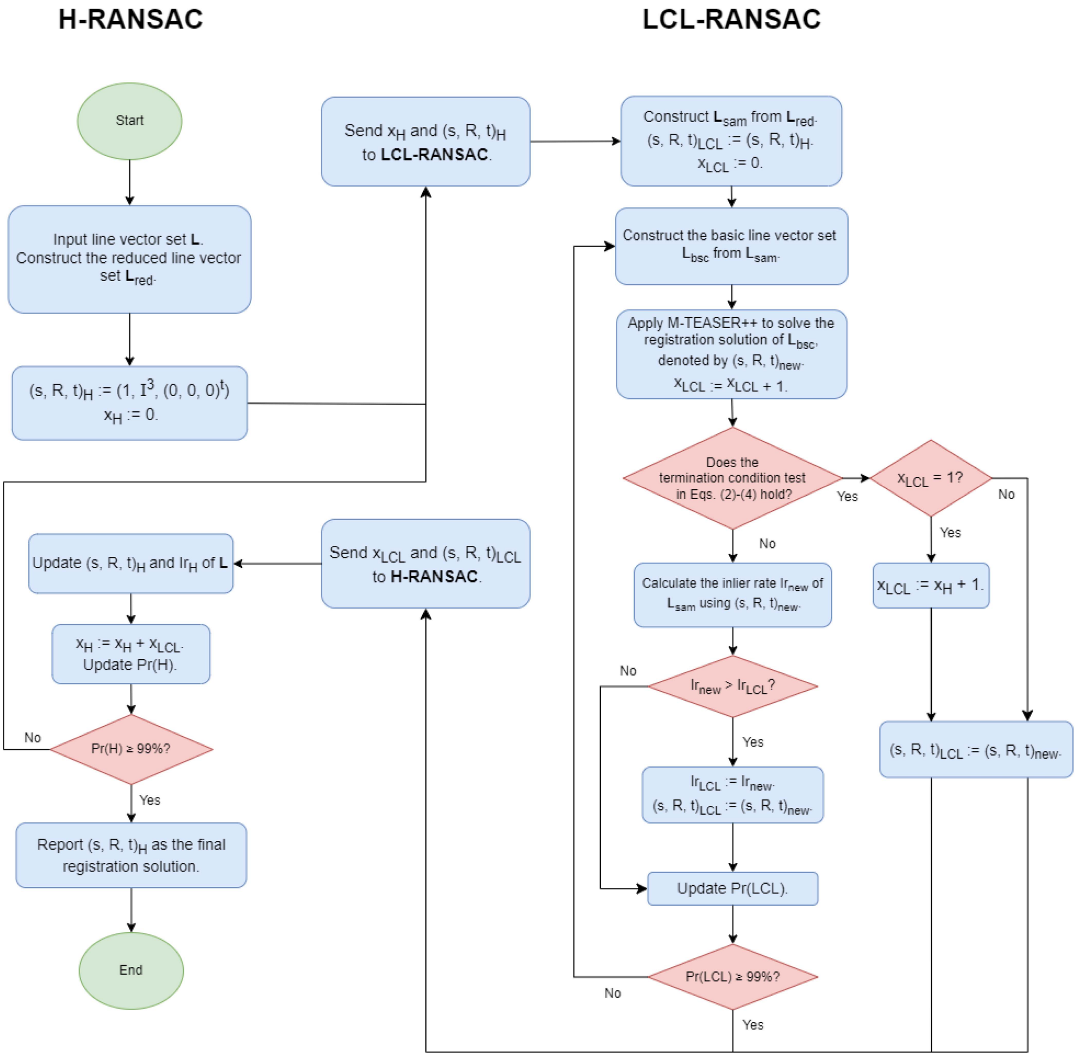

H-RANSAC
LCL-RANSAC
Start
Input line vector set $\mathbf{L}$. Construct the reduced line vector set $\mathbf{L}_{red}$.
$(s, R, t)_H := (1, I^3, (0, 0, 0)^t)$ $x_H := 0$.
Send x_H and $(s, R, t)_H$ to LCL-RANSAC.
Construct $\mathbf{L}_{sam}$ from $\mathbf{L}_{red}$. $(s, R, t)_{LCL} := (s, R, t)_H$. $x_{LCL} := 0$.
Construct the basic line vector set $\mathbf{L}_{bsc}$ from $\mathbf{L}_{sam}$.
Apply M-TEASER++ to solve the registration solution of $\mathbf{L}_{bsc}$, denoted by $(s, R, t)_{new}$. $x_{LCL} := x_{LCL} + 1$.
Does the termination condition test in Eqs. (2)-(4) hold?
Yes
No
$x_{LCL} = 1$?
No
Yes
$x_{LCL} := x_H + 1$.
Calculate the inlier rate Ir_{new} of $\mathbf{L}_{sam}$ using $(s, R, t)_{new}$.
$Ir_{new} > Ir_{LCL}$?
No
Yes
$Ir_{LCL} := Ir_{new}$. $(s, R, t)_{LCL} := (s, R, t)_{new}$.
$(s, R, t)_{LCL} := (s, R, t)_{new}$.
Update Pr(LCL).
Pr(LCL) ≥ 99%?
No
Yes
Send x_{LCL} and $(s, R, t)_{LCL}$ to H-RANSAC.
Update $(s, R, t)_H$ and Ir_H of $\mathbf{L}$
$x_H := x_H + x_{LCL}$. Update Pr(H).
Pr(H) ≥ 99%?
No
Yes
Report $(s, R, t)_H$ as the final registration solution.
End

Fig. 3. Flow chart of our C-RANSAC registration algorithm.

Barath and Matas [34] proposed a Graph-Cut RANSAC-based registration method using the graph-cut process [35]. Their registration method takes spatial coherences of points into consideration in order to separate the correct and wrong inliers of the consensus set, particularly correcting wrongly estimated inliers of the consensus set with an energy function locally minimized via the graph-cut process. Experimental data indicated that the Graph-Cut RANSAC-based registration method is simple but effective. In [36], it was shown that on average, combining with Graph-Cut RANSAC and one model estimator [37], [38] can achieve better registration results when compared with RANSAC, the Bayesian model estimation method [37], the universal framework for RANSAC [39], and the neural-guided RANSAC [40].

Based on the given line vector correspondence set $\mathbf{L}$, Yang et al. [41] proposed a truncated least squares estimation and semidefinite relaxation (TEASER) method. Later, Yang et al. [42] proposed an improved version of TEASER, called TEASER++. In their TEASER++ method, the PCR problem is first transformed into a cost minimization problem. Next, based on a graph model, the PCR problem is decomposed into the scale estimation problem, the rotation estimation problem, and the translation estimation problem. Experimental data demonstrated that the TEASER++ method outperforms RANSAC, the branch and bound approach [43], and the heuristic approach [44]. Considering only the rotation and translation estimations, TEASER++ also outperforms the Fast-GlobReg method [19]. To handle cases for high outlier rates, Shi et al. [45] proposed a method, called the reject outliers based on INvariants (ROBIN), to boost robustness performance for PCR. Instead of the maximal clique idea used in TEASER++, ROBIN utilizes the maximum k-core idea to remove outliers such that its accuracy is competitive with TEASER++, but it can speed up the execution time performance.

Improving the topological graph model-based method [46], Li et al. [47] first decomposed the PCR problem into the scale, rotation, and translation estimation subproblems. Based on the line vector correspondence set $\mathbf{L}$, they proposed a one-point RANSAC method to estimate the scale solution first. After that, the constructed maximal consensus set is used to remove outliers

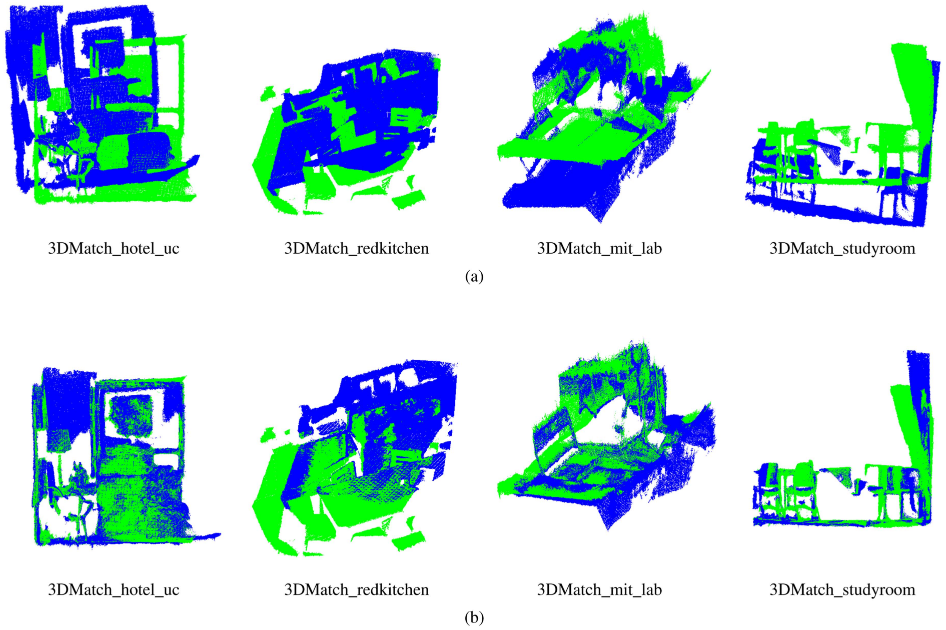


Fig. 4. Perceptual effects of our C-RANSAC registration algorithm. (a) Four testing point cloud pairs, where the source point set $\mathbf{P}^s$ marked in blue and the target point set $\mathbf{P}^t$ marked in green. (b) Four alignments.

from $\mathbf{L}$. To estimate the rotation solution, the scale-annealing biweight function, differentiation technique, and weighted least-square regression method are used together. Based on the estimated scale and rotation solutions, frequency-based potential inliers in the projected correspondence set are obtained. Finally, the translation solution is estimated. Experimental data demonstrated that the one-point RANSAC method outperforms FLORANSAC [48], K4PCS [49], TEASER++ [42], and FMP+BnB [50].

Differing from the graph-theoretic approach to reject outliers [45], Sun [51] proposed a random sampling with invariance and compatibility (RANSIC) method. In RANSIC, the theory of invariance and compatibility is used to extract inlier candidates from the initial correspondence set $\mathbf{C}$, and then the consensus set is determined by comparing the compatibility between each new candidate, which consists of three correspondences, and all the old candidates. The abovementioned iterative random sampling process is repeated until the termination condition is reached. Experimental results illustrated that the RANSIC method outperforms Fast-GlobReg [19], GNC-TLS [44], the singular value decomposition-based RANSAC method [52], the guaranteed outlier removal (GORE) method [53], and TEASER [41]. Recently, based on two layers of random sampling operation, a double-layered sampling with consensus maximization solver, called DANIEL [54], leading to low computational cost and high robustness effects under high outlier rates. Experimental data demonstrated the superiority of DANIEL relative to GNC-TLS [44] and GORE [53].

B. Contributions

In this article, a new centralized RANSAC (C-RANSAC) registration algorithm with fast convergence and high accuracy is proposed. The pipeline of our C-RANSAC registration algorithm is depicted in Fig. 1 where the host RANSAC (H-RANSAC) and the local RANSAC (LCL-RANSAC) cooperate in a handshake way. To clarify the novelties and contributions of our algorithm, the functionality of each module in Fig. 1 is expressed as follows.

1) In the first novelty of our algorithm, a scale histogram-based outlier removal method is proposed to delete outliers from the initial line vector set $\mathbf{L}$ for constructing a reduced line vector set $\mathbf{L}_{\text{red}}$ with $|\mathbf{L}_{\text{red}}| \ll |\mathbf{L}|$.
2) Differing from the conventional RANSAC approach, the second novelty is that in our C-RANSAC registration algorithm, there are two RANSACs, H-SANSAC, and LCL-RANSAC. H-SANSAC, which works on the set $\mathbf{L}$, cooperates with LCL-RANSAC, which works on the set $\mathbf{L}_{\text{red}}$, in a handshake way.
3) The third novelty of our algorithm is that after receiving the global registration solution $(s, \mathrm{R}, t)_H$ and the global iteration number x_H from H-SANSAC, LCL-RANSAC uses $(s, \mathrm{R}, t)_H$ as the initial solution of the modified TEASER++ (M-TEASER++) method, which will be described in the Appendix, to calculate its first local registration solution $(s, \mathrm{R}, t)_{\text{LCL}}$. Next, LCL-RANSAC checks whether the global iteration number inheritance condition "$(s, \mathrm{R}, t)_{\text{LCL}}$ is close to $(s, \mathrm{R}, t)_H$" holds or not. If the inheritance condition holds, it means that the first local registration solution $(s, \mathrm{R}, t)_{\text{LCL}}$ is seemingly obtained by executing the HAV process $x_H + 1$ iterations using H-SANSAC, so LCL-RANSAC sets the local iteration number to $x_{\text{LCL}} = x_H + 1$. Furthermore, LCL-RANSAC sends $(s, \mathrm{R}, t)_{\text{LCL}}$ and x_{LCL} back to H-SANSAC. Otherwise, if the global iteration number inheritance condition fails, LCL-RANSAC iteratively refines its local solution using the M-TEASER++ method, and finally, it sends the resultant local registration solution and the local iteration

TABLE I
REGISTRATION ACCURACY AND EXECUTION TIME COMPARISON

s^{err} (10^{-2})

Outlier rate Method	0.5	0.6	0.7	0.8	0.9	avg
TEASER++ [42]	0.02	**0.022**	**0.027**	0.049	0.259	0.075
One-point RANSAC [47]	0.03	0.05	0.09	0.19	0.39	0.15
RANSIC [51]	**0.019**	**0.022**	0.028	**0.038**	**0.066**	**0.035**
Ours	**0.017**	**0.02**	**0.015**	**0.021**	**0.044**	**0.023**

R^{err} (degree)

Outlier rate Method	0.5	0.6	0.7	0.8	0.9	avg
TEASER++ [42]	0.06	0.064	0.069	0.083	0.106	0.076
One-point RANSAC [47]	**0.021**	**0.023**	**0.029**	0.157	0.406	0.127
RANSIC [51]	**0.013**	**0.015**	**0.019**	**0.026**	**0.045**	**0.023**
Ours	0.039	0.044	0.05	**0.062**	**0.092**	**0.057**

t^{err} (cm)

Outlier rate Method	0.5	0.6	0.7	0.8	0.9	avg
TEASER++ [42]	**0.234**	**0.246**	**0.272**	**0.324**	**0.531**	**0.322**
One-point RANSAC [47]	0.67	0.85	1.24	4.71	20.47	5.588
RANSIC [51]	0.389	0.48	0.66	1.009	1.923	0.892
Ours	**0.161**	**0.178**	**0.203**	**0.249**	**0.399**	**0.238**

RMSE (10^{-3})

Outlier rate Method	0.5	0.6	0.7	0.8	0.9	avg
TEASER++ [42]	4.666	**4.909**	**5.383**	**6.122**	**8.738**	**5.964**
One-point RANSAC [47]	6.5	8.3	12.3	40.2	113.6	36.18
RANSIC [51]	**4.163**	4.925	6.273	8.696	15	7.812
Ours	**2.714**	**3.022**	**3.499**	**4.404**	**7.124**	**4.153**

Execution Time (second)

Outlier rate Method	0.5	0.6	0.7	0.8	0.9	avg
TEASER++ [42]	1.701	1.035	0.711	0.546	0.497	0.898
One-point RANSAC [47]	7.806	7.645	7.733	7.756	7.557	7.699
RANSIC [51]	**0.008**	**0.008**	**0.008**	**0.009**	**0.037**	**0.014**
Ours	**0.121**	**0.102**	**0.101**	**0.099**	**0.093**	**0.103**

The best result is marked in boldface black color, and the second-best result is marked in boldface orange color.

TABLE II
REGISTRATION ACCURACY AND EXECUTION TIME COMPARISON OF 3DMATCH DATASET

Average accuracy Method	R^{err} (°)	t^{err} (cm)	RMSE	Time (s)
Graph-Cut RANSAC [36]	3.174	9.313	0.070	**0.223**
TEASER++ [42]	**2.648**	8.685	**0.058**	0.904
One-point RANSAC [47]	2.833	**8.505**	0.061	4.646
RANSIC [51]	3.534	10.145	0.078	**0.025**
Ours	**1.915**	**6.505**	**0.048**	1.136

The best result is marked in boldface black color, and the second-best result is marked in boldface orange color.

number to H-SANSAC for updating the global solution, iteration number, and confidence level.

4) Based on the point cloud datasets: 3DMatch [55] and RGB-D [56], the registration accuracy and the execution time comparison for the considered methods are made. The four registration accuracy metrics used are scale error, rotation error, translation error, and root mean square error (RMSE). The execution time performance is measured in seconds. Comprehensive experimental data have demonstrated the registration accuracy and execution time merits of our C-RANSAC registration algorithm when compared with the state-of-the-art methods, such as Graph-Cut RANSAC [36], TEASER++ [42], one-point RANSAC [47], and RANSIC [51].

The rest of this article is organized as follows. In Section II, the proposed scale histogram-based outlier removal method is presented to construct a reduced line vector correspondence set. In Section III, our C-RANSAC algorithm is presented. In Section IV, thorough experimental results are demonstrated to justify the superiority of our algorithm. Finally, Section V concludes this article.

II. Scale Histogram-Based Outlier Removal Method

The proposed scale histogram-based outlier removal method shown in Fig. 1 is presented in detail for constructing the reduced line vector set $\mathbf{L}_{\text{red}}$ from the initial line vector set $\mathbf{L}$. The two sets $\mathbf{L}$ and $\mathbf{L}_{\text{red}}$ will be used as the working spaces by H-RANSAC and LCL-RANSAC, respectively.

A. Initial Line Vector Set

Given a source point cloud set $\mathbf{P}^s$, a target point cloud set $\mathbf{P}^t$, and the initial correspondence set $\mathbf{C} = \{(x_i, y_i) \in \mathbb{R}^3 \times \mathbb{R}^3 \mid x_i \in \mathbf{P}^s \text{ and } y_i \in \mathbf{P}^t \text{ for } 1 \leq i \leq |\mathbf{P}^s|\}$, the goal of PCR is to estimate a positive scale parameter $s \in \mathbb{R}$, an orthogonal rotation matrix $\mathrm{R} \in \mathrm{SO}(3)$, and a translation vector $t \in \mathbb{R}^3$ such that the aligned source point set and the original target point set satisfies

$$\underset{s,R,t}{\text{Min}} \sum_{i=1}^{|\mathbf{P}^s|} w_i ||(s\mathrm{R}x_i + t) - \mathrm{y_i}||^2 \tag{1}$$

where y_i is modeled as $y_i = s\mathrm{R}x_i + t + n_i$ in which n_i denotes a bounded measurement noise, which can be assumed to be a zero-mean Gaussian noise [42], [47].

Based on the initial correspondence set $\mathbf{C}$, it is easy to construct the initial line vector set $\mathbf{L} = \{(v^x_{i,j}, v^y_{i,j})$ for $1 \leq i < j \leq |\mathbf{C}|\}$ where $v^x_{i,j} = x_i - x_j$, $v^y_{i,j} = y_i - y_j$, and $|\mathbf{L}| = \frac{|\mathbf{C}|(|\mathbf{C}|-1)}{2}$.

B. Constructing the Reduced Line Vector Set $\mathbf{L}_{red}$

Based on the initial line vector set $\mathbf{L}$, we take one example to assist in explaining the proposed scale histogram-based outlier removal method for constructing the reduced line vector set $\mathbf{L}_{\text{red}}$. Fig. 2(a) depicts one source point set $\mathbf{P}^s$ marked in blue and the target point set $\mathbf{P}^t$ marked in green, where $\mathbf{P}^s$ is adopted from the "3DMatch" dataset, which is available on the website https://3dmatch.cs.princeton.edu/ [55], and $\mathbf{P}^t$ is randomly transformed from $\mathbf{P}^s$. To simulate outliers and the noise effect on correspondences between $\mathbf{P}^s$ and $\mathbf{P}^t$, Gaussian-bounded noise is artificially added to all the target points in $\mathbf{P}^t$, and a random translation is artificially applied to 90% of the target points in $\mathbf{P}^t$. The detailed construction of the artificial testing point clouds is referred to Section IV-A-1. In Fig. 2(b), the lines marked in red denote the initial correspondence set $\mathbf{C}$ with $|\mathbf{C}| = 1993$, which has 90% of outliers, and the number of initial line vectors in $\mathbf{L}$ equals 1 985 028 $\left(= \frac{1993 \times 1992}{2}\right)$.

Considering each line vector $(v^x_{i,j},\ v^y_{i,j}) \in \mathbf{L}$, let $s_{i,j}$ denote the positive scale of the Euclidean length of $v^y_{i,j}$ over that of $v^x_{i,j}$, i.e., $s_{i,j} = \frac{\|v^y_{i,j}\|}{\|v^x_{i,j}\|}$. Fig. 2(c) demonstrates the scale histogram of the scale set of all line vectors in Fig. 2(b), where in the scale histogram $\mathbf{H}(s)$, the x-axis indicates the scale parameter s and the y-axis indicates the frequency of s. In Fig. 2(c), there are 25 quantized scales on the x-axis, and each quantized scale as well as the corresponding frequency constitutes a bin with a width of 0.2.

To remove outliers from $\mathbf{L}$, we retain the line vectors in the bins with the highest peak and those in the left and right neighboring bins for constructing the reduced line vector set $\mathbf{L}_{\text{red}}$. Fig. 2(d) illustrates the reduced scale histogram in which it yields to $\mathbf{L}_{\text{red}}$ with $|\mathbf{L}_{\text{red}}| = 411305$. For this example, the outlier removal rate of the line vector set $\mathbf{L}$ equals 79.3% $\left(= \frac{|\mathbf{L}|-|\mathbf{L}_{\text{red}}|}{|\mathbf{L}|} = \frac{(1\,985\,028-411\,305)}{1\,985\,028}\right)$. As a result, the reduced line vector set $\mathbf{L}_{\text{red}}$ is used as the working space for LCL-RANSAC and the line vector set $\mathbf{L}$ is used as the working space for H-RANSAC.

III. Our C-RANSAC Registration Algorithm

The pipeline of our C-RANSAC registration algorithm has been sketched in Fig. 1 and the contributions of our algorithm is listed first. Next, two handshake processes are simulated to explain why if the initial global solution is not correct, how our algorithm can get better solution, approaching the final correct solution effectively. Then, the flow chart of our algorithm is offered. Finally, the analysis of fast convergence and high accuracy benefits of our algorithm is provided.

A. Our C-RANSAC Registration Algorithm and Flow Chart

The proposed C-RANSAC registration algorithm is listed as follows.

Algorithm 1: C-RANSAC Registration.

```
Input: The line vector set $\mathbf{L}$.
Output: The registration solution of $\mathbf{L}$.
Step 1: (Constructing the reduced line vector set $\mathbf{L}_{red}$)
Construct the reduced line vector set $\mathbf{L}_{red}$ using the scale
  histogram-based outlier removal method.
Step 2: (Initialization of H-RANSAC)
H-RANSAC sets the initial global registration solution to
  $(s, \mathrm{R}, t)_H = (1,\ \mathbf{I}^3,\ (0,\ 0,\ 0)^t)$ and sets the iteration
  number to $x_H = 0$. Then, H-RANSAC sends $(s, \mathrm{R}, t)_H$
  and $x_H$ to LCL-RANSAC. Go to Step 3.
Step 3: (Calculating the first registration solution of $\mathbf{L}_{sam}$
  by LCL-RANSAC)
LCL-RANSAC first constructs a new line vector subset
  $\mathbf{L}_{sam}$ which is randomly selected from $\mathbf{L}_{red}$. For
  notation's convenience, LCL-RANSAC first renames the
  received global solution $(s, \mathrm{R}, t)_H$ as $(s, \mathrm{R}, t)_{LCL}$. Next,
  LCL-RANSAC takes the received global solution, which
  has been renamed to $(s, \mathrm{R}, t)_{LCL}$, as the initial solution of
  the M-TEASER++ method (see Appendix) to calculate
  the first local registration solution $(s, \mathrm{R}, t)_{new}$ based on
  the newly generated basic line vector set $\mathbf{L}_{bsc}$ which is
  randomly sampled from $\mathbf{L}_{sam}$. In addition,
  LCL-RANSAC sets the tentative iteration number to
  $x_{LCL} = 1$. Go to Step 4.
Step 4: (Global iteration number inheritance condition test)
Let $(s_{LCL}, \mathrm{R}_{LCL}, t_{LCL}) = (s, \mathrm{R}, t)_{LCL}$ and
  $(s_{new}, \mathrm{R}_{new}, t_{new}) = (s, \mathrm{R}, t)_{new}$. LCL-RANSAC
  checks the following three inequalities:

      $|s_{new} - s_{LCL}| \leq 2\tau$                                             (2)

      $\arccos\left(\frac{\mathrm{tr}(\mathrm{R}_{\text{new}}(\mathrm{R}_{\text{LCL}})^{\mathrm{T}}) - 1}{2}\right) \leq 0.01$     (3)

      $\|t_{new} - t_{LCL}\| \leq \tau,$                                           (4)

  where the noise bound $\tau$ is set by users. If the three
  inequalities in (2)–(4) does not hold, LCL-RANSAC
  goes to Step 5; otherwise, LCL-RANSAC sets the local
  iteration number to

      $x_{LCL} := \begin{cases} x_H + 1 & \text{if } (s, \mathrm{R}, t)_{LCL} \text{ is renamed} \\ & \text{from } (s, \mathrm{R}, t)_H \\ x_{LCL}; & \text{otherwise} \end{cases}$     (5)

  In (5), if $(s, \mathrm{R}, t)_{LCL}$ is renamed from $(s, \mathrm{R}, t)_H$, it
  indicates that the first local registration solution obtained
  by LCL-RANSAC is close to the received global
  registration solution. Then, LCL-RANSAC directly
  returns $x_{LCL}$ and $(s, \mathrm{R}, t)_{new}$ to H-RANSAC, and goes
  to Step 6.
```

```
Step 5: (Local confidence level condition test)
LCL-RANSAC updates its local confidence level to
  $Pr(LCL) = 1 - (1 - Ir_{LCL})^{x_{LCL}}$ where $Ir_{LCL}$
  denotes the currently highest inlier rate of $\mathbf{L}_{sam}$. If
  $Pr(LCL) \geq 99\%$, LCL-RANSAC returns the currently
  best local registration solution $(s, \mathrm{R}, t)_{LCL}$ and the
  iteration number $x_{LCL}$ to H-RANSAC, and then goes to
  Step 6. If $Pr(LCL)$ is less than 99%, LCL-RANSAC
  performs the M-TEASER++ method on the newly
  generated basic line vector set $\mathbf{L}_{bsc}$ to refine the local
  registration solution of $\mathbf{L}_{sam}$, denoted by $(s, \mathrm{R}, t)_{new}$,
  and set the local iteration number to $x_{LCL}$ to $x_{LCL} :=$
  $x_{LCL} + 1$. Go to Step 4.
Step 6: (Updating the global solution and iteration number)
H-RANSAC utilizes the received local registration solution
  to update the global inlier rate and the global registration
  solution of $\mathbf{L}$. In addition, H-RANSAC updates the global
  iteration number to $x_H := x_H + x_{LCL}$. After that,
  H-RANSAC updates the global confidence level to
  $Pr(H) = 1 - (1 - Ir_H)^{x_H}$ where $Ir_H$ denotes the
  currently highest inlier rate of $\mathbf{L}$. Go to Step 7.
Step 7: (Global confidence level condition test)
If $Pr(H) < 99\%$, H-RANSAC transmits the currently
  best global registration solution $(s, \mathrm{R}, t)_H$ and the
  updated global iteration number $x_H$ to LCL-RANSAC
  and goes to Step 3. Otherwise, if $Pr(H) \geq 99\%$,
  H-RANSAC stops the algorithm and reports the final
  registration solution $(s, \mathrm{R}, t)_H$.
```

After listing our C-RANSAC algorithm, we simulate the first two handshakes to make our algorithm more understandable.

In the first handshake process, after completing Step 2, usually the initial global registration solution $(s, \mathrm{R}, t)_H$ is not correct and is far away from the final correct one. In Step 3, LCL-RANSAC utilizes the received initial global registration solution as the initial solution of the M-TEASER++ method to calculate the first local registration solution $(s, \mathrm{R}, t)_{\mathrm{new}}$ based on the newly generated basic line vector set $\mathbf{L}_{\mathrm{bsc}}$, which is randomly sampled from $\mathbf{L}_{\mathrm{sam}}$. Because the line vector subset $\mathbf{L}_{\mathrm{sam}}$ is randomly selected from the reduced line vector set $\mathbf{L}_{\mathrm{red}}$, which has removed outliers, the first local registration solution $(s, \mathrm{R}, t)_{\mathrm{new}}$ should be closer to the final correct registration solution when compared with the initial global registration solution $(s, \mathrm{R}, t)_H$. In Step 4, because usually, $(s, \mathrm{R}, t)_{\mathrm{new}}$ is much different from $(s, \mathrm{R}, t)_H$, the global iteration number inheritance condition test will, thus, fail, and then in Step 5, LCL-RANSAC iteratively refines the local registration solution of $\mathbf{L}_{\mathrm{sam}}$ until the local confidence level reaches the specified threshold. In Step 6, H-RANSAC utilizes the received local registration solution to update the global registration solution and inlier rate of $\mathbf{L}$. In addition, H-RANSAC utilizes the received local iteration number to update the accumulated global iteration number as $x_H + x_{\mathrm{LCL}}$, and then the global confidence level is further updated.

In the second handshake process, after receiving the updated global registration solution and global iteration number from H-RANSAC, LCL-RANSAC takes the received global registration solution as the initial solution of the M-TEASER++ method to calculate the first local registration solution $(s, \mathrm{R}, t)_{\mathrm{new}}$ of $\mathbf{L}_{\mathrm{sam}}$ again. Furthermore, in Step 4, LCL-RANSAC checks whether the global iteration number inheritance condition test holds or not. If the inheritance condition holds, LCL-RANSAC sends the first local registration solution and the iteration number, $x_H + 1$, back to H-RANSAC for updating the global solution, iteration number, inlier rate, and confidence level. Otherwise, if the inheritance condition fails, LCL-RANSAC iteratively refines the local registration of $\mathbf{L}_{\mathrm{sam}}$ by using the M-TEASER++ method.

For completeness, the detailed flow chart of our algorithm is illustrated in Fig. 3.

B. Analysis of Fast Convergence and High Accuracy Benefits of Our Algorithm

We begin the analysis of the fast convergence benefit of our algorithm, and then analyze the high accuracy benefit.

1) Fast Convergence Benefit: When our algorithm, i.e., Algorithm 1, terminates, the global confidence level of H-RANSAC satisfies

$$\begin{aligned}\mathrm{Pr}(H) &= 1 - (1 - \mathrm{Ir}_H)^{\sum_{i=1}^{m} x_{\mathrm{LCL}}^{(i)}} \\ &= 1 - (1 - \mathrm{Ir}_H)^M \\ &\geq 99\%\end{aligned} \tag{6}$$

where "m" indicates that LCL-RANSAC is called m times by the H-RANSAC. "$x_{\mathrm{LCL}}^{(i)}$," $1 \leq i \leq m$, denotes the local iteration number returning to H-RANSAC after completing the ith call to LCL-RANSAC.

Equation (6) indicates that based on the line vector set $\mathbf{L}$, H-RANSAC only updates the global confidence level, the global inlier rate, and the global registration solution m times instead of M times. Due to $m \ll M$, our algorithm has a fast convergence benefit relative to the traditional RANSAC method. We take the point cloud pair in Fig. 2(b) to clarify the fast convergence benefit of our algorithm.

From Fig. 2(b), it is known that $|\mathbf{L}| = 1\,985\,028$ and $|\mathbf{L}_{\mathrm{red}}| = 411\,305$. After randomly sampling 10% line vectors from $\mathbf{L}_{\mathrm{red}}$ to construct $\mathbf{L}_{\mathrm{sam}}$, we have $|\mathbf{L}_{\mathrm{sam}}| = 41\,130$. Furthermore, after randomly sampling 30% line vectors from $\mathbf{L}_{\mathrm{sam}}$ to construct the basic line vector $\mathbf{L}_{\mathrm{bsc}}$, we have $|\mathbf{L}_{\mathrm{bsc}}| = 12\,339$. After performing our C-RANSAC registration algorithm on Fig. 2(b), the number of handshake processes between H-RANSAC and LCL-RANSAC is five; on the other hand, we have $m = 5$. Among the five handshakes, our experience shows that the five local iteration numbers sent from LCL-RANSAC to H-RANSAC are $x_{\mathrm{LCL}}^{(1)} = 2$, $x_{\mathrm{LCL}}^{(2)} = 2$, $x_{\mathrm{LCL}}^{(3)} = 5$, $x_{\mathrm{LCL}}^{(4)} = 2$, and $x_{\mathrm{LCL}}^{(5)} = 12$; the global iteration inheritance condition holds in the third and fifth handshake processes. It can be checked that the five iteration numbers required in the M-TEASER++ method used by LCL-RANSAC are 2, 2, 1, 2, and 1, respectively. Due to $|\mathbf{L}_{\mathrm{red}}| \ll |\mathbf{L}|$

and employing the global iteration number inheritance condition test into our algorithm, our algorithm has a fast convergence benefit, i.e., an execution time reduction benefit.

2) High Registration Accuracy Benefit: In our C-RANSAC algorithm, at each time, once LCL-RANSAC receives the global registration solution $(s, \mathrm{R}, t)_H$ from H-RANSAC, LCL-RANSAC uses $(s, \mathrm{R}, t)_H$ as the initial solution of the M-TEASER++ method to calculate the first local solution based on $\mathbf{L}_{\text{red}}$. Because in $\mathbf{L}_{\text{red}}$ with $|L_{\text{red}}| \ll |L|$, the outliers have been removed to some degree from the initial line vector set $\mathbf{L}$, the local registration solution estimated by LCL-RANSAC has higher accuracy than that estimated by the classical RANSAC-based registration method on $\mathbf{L}$. Because each time, H-RANSAC cooperates with LCL-RANSAC in a handshake way, it implies that the global registration solution estimated by H-RANSAC has higher accuracy than that using the classical RANSAC method.

IV. Experimental Results

To evaluate the registration performances of our C-RANSAC registration algorithm, simply called "Ours," and the comparative methods, two experiment designs are provided.

In the first experiment design with considering the scale solution, we follow the same experiment design used in [42] and [51] to compare the accuracy performance and the execution time performance among our C-RANSAC algorithm, TEASER++ [42], one-point RANSAC [47], and RANSIC [51].

In the first experiment design, the four accuracy metrics, namely the scale error (s^{err}), the rotation error (R^{err}), the translation error (t^{err}), and the RMSE [11], are used to evaluate the accuracy performance of each considered method. Let the ground truth registration solution be denoted as $(s^{\text{gt}}, \mathrm{R}^{\text{gt}}, t^{\text{gt}})$ and the estimated registration solution of the considered method be denoted as $(s^{\text{est}}, \mathrm{R}^{\text{est}}, t^{\text{est}})$. The four accuracy metrics are defined as

$$s^{\text{err}} = |s^{\text{gt}} - s^{\text{est}}| \tag{7}$$

$$\mathrm{R}^{\text{err}} = \arccos\left(\frac{\mathrm{tr}(\mathrm{R}^{\text{gt}}(\mathrm{R}^{\text{est}})^{\mathrm{T}}) - 1}{2}\right) \tag{8}$$

$$t^{\text{err}} = \|t^{\text{gt}} - t^{\text{est}}\| \tag{9}$$

$$\text{RMSE} = \sqrt{\frac{1}{|\mathbf{P}^s|}\sum_{i=1}^{|\mathbf{P}^s|} \|x_i^{\text{gt}} - x_i^{\text{est}}\|^2} \tag{10}$$

where x_i^{gt} denotes the mapped point by performing the ground truth registration solution, $(s^{\text{gt}}, \mathrm{R}^{\text{gt}}, t^{\text{gt}})$, on $x_i \in \mathbf{P}^s$, x_i^{est} denotes the mapped point by performing the estimated registration solution, $(s^{\text{est}}, \mathrm{R}^{\text{est}}, t^{\text{est}})$, on $x_i \in \mathbf{P}^s$, and $\mathrm{tr}(\cdot)$ denotes the trace of the matrix. The construction of the testing point clouds and the performance comparison is presented in Section IV-A.

In the second experiment design without considering the scale solution, to compare the accuracy performance and the execution time performance among our C-RANSAC algorithm, Graph-Cut RANSAC [36], TEASER++ [42], one-point RANSAC [47], and RANSIC [51], the three accuracy metrics, namely R^{err}, t^{err}, and RMSE, are used to evaluate the accuracy performance of each considered method. The real point cloud dataset 3DMatch with 1623 point cloud pairs is used. The related performance comparison is presented in Section IV-B.

For fairness, all comparative methods and the proposed method have been implemented on a computer with an Intel Core i7-8700 CPU 3.2 GHz and 32 GB RAM. The operating system is the Microsoft Windows 10 64-bit operating system. The program development environment is the Windows Subsystem for Linux and MATLAB R2022b. The C++ source code of our C-RANSAC registration method can be accessed from the website: https://github.com/ivpml84079/C-RANSAC.

A. Registration Accuracy and Execution Time Comparison: Considering Scale Solution

1) Constructing the Artificial Testing Point Clouds: In the first experiment design, as the 22 testing source point clouds, eight point clouds are selected from the 3DMatch dataset and 14 point clouds are selected from the RGB-D dataset. Next, we apply the voxel grid downsampling method, namely "Voxelgrid" in the Point Cloud Library [57], to downsample every source point cloud such that the size of the sampled source point cloud $\mathbf{P}^s$ equals $|\mathbf{P}^s| = 2000$. Then, we create the corresponding target point cloud $\mathbf{P}^t$ by performing a random transformation on each source point cloud $\mathbf{P}^s$, and the transformed point cloud is viewed as the ground truth registration solution used in (7)–(10).

In detail, given one sampled source point cloud $\mathbf{P}^s$, using the random transformation procedure [42], the ground truth target point cloud $\mathbf{P}^t$ is generated such that the generated ground truth scale solution s^{gt}, rotation solution R^{gt}, and translation solution t^{gt} satisfy the three constraints: $1 \le s^{\text{gt}} \le 5$, $\|t^{\text{gt}}\| \le 1$, and $\mathrm{R}^{\text{gt}} \in \mathrm{SO}(3)$. After performing the random transformation on $\mathbf{P}^s$, the random bounded noise n_i is added to each target point in $\mathbf{P}^t$, where $n_i = ([-1,1] * \tau, [-1,1] * \tau, [-1,1] * \tau)\mathrm{m}$ in which the noise bound τ is set to 0.05 [58], [59] and "m" denotes the unit "meter." Following the same outlier rates considered in [42] and [51], based on each target point cloud $\mathbf{P}^t$, we first randomly sample target points from $\mathbf{P}^t$ based on the five different outlier rates, namely 0.5, 0.6, 0.7, 0.8, and 0.9. and then we translate the five sampled target points over the interval $(\pm[5, 10], \pm[5, 10], \pm[5, 10])\mathrm{m}$. The construction of the initial correspondence set $\mathbf{C}$ and the initial line vector set $\mathbf{L}$ are, thus, followed.

2) Registration Accuracy Comparison and Discussion: For every testing point cloud pair, each considered method is tested 100 times. In terms of s^{err}, R^{err}, t^{err}, and RMSE, the accuracy performances of the considered methods are demonstrated in the former four subtables of Table I, respectively. In the last column of each subtable, "avg" is the abbreviation of "average value."

In Table I, at one specific outlier rate, among the considered methods, the best accuracy result, i.e., the least error, is marked in black color and boldface, and the second-best accuracy result is marked in orange color and boldface. From Table I, we observe that in terms of the two accuracy metrics, s^{err} and t^{err}, our algorithm has the best accuracy performance. In terms of the rotation error $\mathrm{R}^{\mathrm{err}}$, the RANSIC method has the best accuracy performance, and our algorithm has the second-best accuracy performance. In terms of the metric RMSE, our algorithm has the best accuracy performance, and the TEASER++ method has the second-best accuracy performance.

In terms of the overall accuracy metric "RMSE," our algorithm has the best accuracy performance, and the TEASER++ method has the second-best accuracy performance. As discussed in Section III-B2, in the reduced line vector set $\mathbf{L}_{\mathrm{red}}$ used by LCL-RANSAC in our C-RANSAC algorithm, the outliers have been removed from the initial line vector set $\mathbf{L}$, the local registration solution estimated by LCL-RANSAC has higher accuracy. In addition, because each time, H-RANSAC cooperates with LCL-RANSAC in a handshake way, it leads to a global registration solution.

Although the RANSIC method exhibits the best rotation performance, in terms of RMSE, its overall registration accuracy performance is the third-best due to the deficiency in translation accuracy. The main reason is that RANSIC directly calculates the difference between the mean vector of the source points and that of target points in the inlier set as the translation solution.

3) Execution Time Comparison and Discussion: In terms of seconds, the execution time performance of each considered registration method is shown in the last subtable of Table I.

The last subtable of Table I indicates that the RANSIC method has the best execution time performance. Our algorithm has the second-best execution time performance. Because RANSIC utilizes the theory of invariance and compatibility to extract inlier candidates from the initial correspondence set $\mathbf{C}$, the cardinality of the consensus set determined in each iteration is small, leading to the best execution time performance. Our good execution time performance in Table I justifies the fast convergence benefit analysis in Section III-B1.

In summary, from the abovementioned registration accuracy and execution time comparison and discussion for the 22 testing point cloud pairs with considering scale solution, the execution time performance of our C-RANSAC registration algorithm is the second-best and the registration accuracy performance of our algorithm is the best relative to the three state-of-the-art registration methods.

B. Registration Accuracy and Execution Time Comparison: Without Considering Scale Solution

In this experiment design without considering the scale solution, the 3DMatch dataset with 1623 point cloud pairs is used as the testing dataset. Similar to the first experiment design, the noise bound τ is also set to 0.05 in the second experiment design. $\mathrm{R}^{\mathrm{err}}$, t^{err}, and RMSE, which have been defined in (8)–(10), are used to compare the registration accuracy performance among Graph-Cut RANSAC, TEASER++, one-point RANSAC, RANSIC, and our algorithm.

1) Registration Accuracy Comparison and Discussion: The accuracy performances of each considered method are demonstrated in Table II.

From Table II, we observe that in terms of the three accuracy metrics, $\mathrm{R}^{\mathrm{err}}$, t^{err}, and RMSE, our C-RANSAC algorithm has the best accuracy performance, justifying the high accuracy benefit analyzed in Section III-B2. In terms of $\mathrm{R}^{\mathrm{err}}$ and RMSE, the TEASER++ method has the second-best accuracy performance; In terms of t^{err}, the one-point RANSAC method has the second-best accuracy performance.

The perceptual effects of our C-RANSAC algorithm are illustrated in Fig. 4. Fig. 4(a) illustrates the four testing point cloud pairs, namely hotel_uc, redkitchen, mit_lab, and studyroom, which are adopted from the 3DMatch dataset, where the source point set $\mathbf{P}^s$ is marked in blue and the target point set $\mathbf{P}^t$ is marked in green. Fig. 4(b) demonstrates the alignment effects of the registration results using our algorithm. From Fig. 4(b), we observe that the four aligned source point clouds marked in blue have a good match with the four corresponding target point clouds, indicating good perceptual registration effects of our registration algorithm.

2) Execution Time Comparison and Discussion: In terms of seconds, the execution time performance of each considered registration method is shown Table II.

From Table II, we observe that the RANSIC method has the best execution time performance. The Graph-cut RANSAC method has the second-best execution time performance. Because there are a certain quantity of outliers in the 3DMatch dataset, it reduces the outlier removal effect of the scale histogram-based outlier removal method used in our algorithm, degrading the execution time performance of our algorithm.

In summary, from the abovementioned registration accuracy and execution time comparison and discussion for the 3DMatch dataset without considering scale solution, although the execution time performance of our C-RANSAC registration algorithm is fair, the registration accuracy performance of our algorithm is the best relative to the four state-of-the-art registration methods.

V. Conclusion

The proposed C-RANSAC algorithm has been presented for PCR. In our algorithm, the first novelty is the proposal of a scale histogram-based outlier removal to delete outliers from the initial line vector set $\mathbf{L}$ for constructing a reduced line vector set $\mathbf{L}_{\mathrm{red}}$. The second novelty of our algorithm is that the H-RANSAC and the LCL-RANSAC cooperate in a handshake way. H-RANSAC only works on the initial line vector set $\mathbf{L}$, but LCL-RANSAC only works on the reduced line vector set $\mathbf{L}_{\mathrm{red}}$, which is constructed by removing outliers from $\mathbf{L}$. The third novelty of our algorithm is that once LCL-RANSAC receives the global registration solution from the H-RANSAC, the first local registration solution is obtained by the M-TEASER++ method. If the global iteration number inheritance condition holds, it seems that the first local registration solution has been obtained by running the HAV process $x_H + 1$ times by H-RANSAC, and then LCL-RANSAC returns the first local registration solution and the iteration number $x_H + 1$ to H-RANSAC for updating the global solution, iteration number, and confidence level. Even if the global iteration number inheritance condition fails, LCL-RANSAC iteratively refines its local solution using the M-TEASER++ method, and then sends the resultant local solution and the required local iteration number x_{LCL} to H-RANSAC for updating the global solution, the global iteration number to $x_H := x_H + x_{\text{LCL}}$, and the global confidence level. We have conducted extensive experiments on testing point cloud pairs to show the registration accuracy and execution time merits of our algorithm when compared with Graph-Cut RANSAC, TEASER++, one-point RANSAC, and RANSIC.

The future work is to increase the specified confidence level but sacrifice the execution time to achieve a better tradeoff between the registration accuracy and the execution time requirement for our algorithm.

Appendix
M-TEASER++ Method

At each iteration, LCL-RANSAC performs the M-TEASER++ method on the newly constructed basic line vector set $\mathbf{L}_{\text{bsc}}$ to estimate the local registration of $\mathbf{L}_{\text{bsc}}$. The differences between TEASER++ and M-TEASER++ are highlighted as follows.

A. Estimating the Local Scale Solution

LCL-RANSAC randomly selects one line vector s from $\mathbf{L}_{\text{bsc}}$, and then calculates the new inlier rate of $\mathbf{L}_{\text{bsc}}$, denoted by Ir^s_{new}. If $\text{Ir}^s_{\text{new}} > \text{Ir}^s_{\text{LCL}}$, it performs the assignment operation: $\text{Ir}^s_{\text{LCL}} := \text{Ir}^s_{\text{new}}$. The local confidence level of $\mathbf{L}_{\text{bsc}}$ is updated. The above s-selection, local inlier rate update, and confidence level update are repeated until the specified confidence level 99% is reached, and then the maximal consensus set $\subseteq \mathbf{L}_{\text{bsc}}$ is figured out. Finally, a least square technique is performed on the maximal consensus set to obtain the local scale solution of $\mathbf{L}_{\text{bsc}}$.

Prior to solving the rotation parameter of $\mathbf{L}_{\text{bsc}}$, based on the obtained scale solution of $\mathbf{L}_{\text{bsc}}$, denoted by s, for every line vector $(v^x_{i,j}, v^y_{i,j})$ in $\mathbf{L}_{\text{bsc}}$, we scale $v^x_{i,j}$ by the scale solution s to obtain a scaled line vector. For convenience, the scaled line vector set is denoted by $\mathbf{L}^s_{\text{bsc}}$.

B. Estimating the Local Rotation Solution

Prior to applying the iterative GNC-TLS method [44] to calculate the rotation solution R of $\mathbf{L}^s_{\text{bsc}}$, the GNC surrogate function is used as a penalty function by setting a larger (smaller) weight for a line vector in $\mathbf{L}^s_{\text{bsc}}$ with a smaller (larger) residue. In TLS, when the weight is less than the specified threshold, which is decreasingly updated at each iteration, the weight is set to zero for truncating the unreliable line vector in $\mathbf{L}^s_{\text{bsc}}$.

Based on the reliable line vectors in $\mathbf{L}^s_{\text{bsc}}$, the SVD technique is applied to obtain the tentative rotation solution, and the rotation solution update and the weight update are repeated until the sum of the weighted residuals converges. For LCL-RANSAC, the initial rotation matrix is set as R_{LCL}, which is sent from the H-RANSAC. This modified initial rotation matrix setting strategy leads to faster convergence and better rotation accuracy.

Beforeproceeding to solve the translation parameter, based on the obtained rotation solution of $\mathbf{L}^s_{\text{bsc}}$, denoted by R, for every line vector $(v^x_{i,j}, v^y_{i,j})$ in $\mathbf{L}^s_{\text{bsc}}$, we rotate $v^x_{i,j}$ by the rotation solution R to obtain a rotated source vector. Collecting all these rotated source vectors and the corresponding target vectors, we reconstruct the correspondence set, namely $\mathbf{C}^{s\text{R}}_{\text{bsc}}$.

C. Estimating the Local Translation Solution

Based on $\mathbf{C}^{s\text{R}}_{\text{bsc}}$, the adaptive voting-based method used in [42] is applied to solve the translation parameter $t = (t_1, t_2, t_3)$. While solving each independent translation parameter $t_i, 1 \leq i \leq 3$, the max-stabbing method [60] is used to enumerate all possible consensus subsets, and the largest one is chosen as the maximal consensus set. Note that for LCL-RANSAC, the received global tentative translation solution t_{LCL}, which is sent by the H-RANSAC, is used to assist in realizing the max-stabbing method for determining the maximal consensus set. Furthermore, a least square method is applied to calculate the translation solution of $\mathbf{C}^{s\text{R}}_{\text{bsc}}$.

After completing the M-TEASER++ method, the local registration solution of $\mathbf{L}_{\text{bsc}}$, namely $(s, \text{R}, t)_{\text{new}}$, is returned to the LCL-RANSAC.

Acknowledgment

The authors would like to thank the three reviewers for their valuable comments and the proofreading help of C. Harrington to improve the manuscript. They also would like to appreciate the agreement to modify the available codes of TEASER++ [42].

References

[1] D. Hong et al., “More diverse means better: Multimodal deep learning meets remote-sensing imagery classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 5, pp. 4340–4354, May 2021.

[2] H. Hu, Y. Hou, Y. Ding, G. Pan, M. Chen, and X. Ge, “V2PNet: Voxel-to-point feature propagation and fusion that improves feature representation for point cloud registration,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 16, pp. 5077–5088, May 2023.

[3] F. Ghorbani, H. Ebadi, A. Sedaghat, and N. Pfeifer, “A novel 3-D local DAISY-style descriptor to reduce the effect of point displacement error in point cloud registration,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 15, pp. 2254–2273, Feb. 2022.

[4] W. Tao, S. Xu, W. Huang, S. Hu, and M. Pang, “A distinctive binary descriptor and 2-point RANSACWC for point cloud registration,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 16, pp. 7529–7542, Aug. 2023.

[5] J. Li, Q. Hu, and M. Ai, “Robust geometric model estimation based on scaled welsch Q-norm,” *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 8, pp. 5908–5921, Aug. 2020.

[6] J. Li, Q. Hu, and M. Ai, “RIFT: Multi-modal image matching based on radiation-variation insensitive feature transform,” *IEEE Trans. Image Process.*, vol. 29, pp. 3296–3310, Dec. 2020.

[7] R. Huang, Y. Xu, W. Yao, L. Hoegner, and U. Stilla, “Robust global registration of point clouds by closed-form solution in the frequency domain,” *ISPRS J. Photogrammetry Remote Sens.*, vol. 171, pp. 310–329, 2021.

[8] Z. Wang, S. Yan, L. Wu, X. Zhang, and B. Chen, “Robust point clouds registration with point-to-point LP distance constraints in large-scale metrology,” *ISPRS J. Photogrammetry Remote Sens.*, vol. 189, pp. 23–35, 2022.

[9] P. J. Besl and N. D. McKay, “A method for registration of 3-D shapes,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 239–256, Feb. 1992.

[10] 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, pp. 381–395, 1981.

[11] R. Feng, H. Shen, J. Bai, and X. Li, “Advances and opportunities in remote sensing image geometric registration: A systematic review of state-of-the-art approaches and future research directions,” *IEEE Geosci. Remote Sens. Mag.*, vol. 9, no. 4, pp. 120–142, Dec. 2021.

[12] B. K. Horn, “Closed-form solution of absolute orientation using unit quaternions,” *J. Opt. Soc. Amer.*, vol. 4, no. 4, pp. 629–642, 1987.

[13] H. Pottmann, Q. Huang, Y.-L. Yang, and S.-M. Hu, “Geometry and convergence analysis of algorithms for registration of 3D shapes,” *Int. J. Comput. Vis.*, vol. 67, pp. 277–296, 2006.

[14] S. J. He, S. T. Zhao, F. Bai, and J. Wei, “A method for spatial data registration based on PCA-ICP algorithm,” *Adv. Mater. Res.*, vol. 718, pp. 1033–1036, 2013.

[15] J. Jiang, J. Cheng, and X. Chen, “Registration for 3-D point cloud using angular-invariant feature,” *Neurocomputing*, vol. 72, no. 16–18, pp. 3839–3844, 2009.

[16] Z. Yao, Q. Zhao, X. Li, and Q. Bi, “Point cloud registration algorithm based on curvature feature similarity,” *Measurement*, vol. 177, 2021, Art. no. 109274.

[17] N. Mellado, D. Aiger, and N. J. Mitra, “Super 4PCS fast global point cloud registration via smart indexing,” *Comput. Graph. Forum*, vol. 33, no. 5, pp. 205–215, 2014.

[18] D. Aiger, N. J. Mitra, and D. Cohen-Or, “4-points congruent sets for robust pairwise surface registration,” *ACM Trans. Graph.*, vol. 27, 2008, Art. no. 85.

[19] Q.-Y. Zhou, J. Park, and V. Koltun, “Fast global registration,” in *Proc. Eur. Conf. Comput. Vis.*, pp. 766–782, 2016.

[20] M. J. Black and P. Anandan, “The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields,” *Comput. Vis. Image Understanding*, vol. 63, no. 1, pp. 75–104, 1996.

[21] A. Blake and A. Zisserman, *Visual Reconstruction*. Cambridge, MA, USA: MIT Press, 1987.

[22] S. Bouaziz, A. Tagliasacchi, and M. Pauly, “Sparse iterative closest point,” *Comput. Graph. Forum*, vol. 32, no. 5, pp. 113–123, 2013.

[23] F. Bach et al., “Optimization with sparsity-inducing penalties,” *Foundations Trends Mach. Learn.*, vol. 4, no. 1, pp. 1–106, 2012.

[24] A. L. Pavlov, G. W. Ovchinnikov, D. Y. Derbyshev, D. Tsetserukou, and I. V. Oseledets, “AA-ICP: Iterative closest point with anderson acceleration,” in *Proc. IEEE Int. Conf. Robot. Autom.*, 2018, pp. 3407–3412.

[25] J. Li, Q. Hu, Y. Zhang, and M. Ai, “Robust symmetric iterative closest point,” *ISPRS J. Photogrammetry Remote Sens.*, vol. 185, pp. 219–231, 2022.

[26] K. Favre, M. Pressigout, E. Marchand, and L. Morin, “Plane-based accurate registration of real-world point clouds,” in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, 2021, pp. 2018–2023.

[27] J. Zhang, Y. Yao, and B. Deng, “Fast and robust iterative closest point,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 7, pp. 3450–3466, Jul. 2022.

[28] R. B. Rusu, N. Blodow, and M. Beetz, “Fast point feature histograms (FPFH) for 3D registration,” in *Proc. IEEE Int. Conf. Robot. Autom.*, 2009, pp. 3212–3217.

[29] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[30] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional networks for biomedical image segmentation,” in *Proc. 18th Int. Conf. Med. Image Comput. Comput.-Assist. Interv.*, 2015, pp. 234–241.

[31] C. Choy, J. Park, and V. Koltun, “Fully convolutional geometric features,” in *Proc. Int. Conf. Comput. Vis.*, 2019, pp. 8958–8966.

[32] A. Zeng, S. Song, M. Nießner, M. Fisher, J. Xiao, and T. Funkhouser, “3DMatch: Learning the matching of local 3d geometry in range scans,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017.

[33] H. Deng, T. Birdal, and S. Ilic, “PPFNet: Global context aware local features for robust 3 d point matching,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 195–205.

[34] D. Barath and J. Matas, “Graph-cut RANSAC,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 6733–6741.

[35] V. Kolmogorov and R. Zabin, “What energy functions can be minimized via graph cuts?,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 2, pp. 147–159, Feb. 2004.

[36] D. Barath and J. Matas, “Graph-cut RANSAC: Local optimization on spatially coherent structures,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 9, pp. 4961–4974, Sep. 2022.

[37] P. H. S. Torr, “Bayesian model estimation and selection for epipolar geometry and generic manifold fitting,” *Int. J. Comput. Vis.*, vol. 50, pp. 35–61, 2002.

[38] D. Barath, J. Noskova, M. Ivashechkin, and J. Matas, “MagSAC, a fast, reliable and accurate robust estimator,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 1304–1312.

[39] R. Raguram, O. Chum, M. Pollefeys, J. Matas, and J.-M. Frahm, “USAC: A universal framework for random sample consensus,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 2022–2038, Aug. 2013.

[40] E. Brachmann and C. Rother, “Neural-guided RANSAC: Learning where to sample model hypotheses,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 4322–4331.

[41] H. Yang and L. Carlone, “A polynomial-time solution for robust registration with extreme outlier rates,” in *Proc. Robot. Sci. Syst.*, 2019.

[42] H. Yang, J. Shi, and L. Carlone, “TEASER: Fast and certifiable point cloud registration,” *IEEE Trans. Robot.*, vol. 37, no. 2, pp. 314–333, Apr. 2021.

[43] J. Yang, H. Li, D. Campbell, and Y. Jia, “Go-ICP: A globally optimal solution to 3D ICP point-set registration,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 11, pp. 2241–2254, Nov. 2016.

[44] H. Yang, P. Antonante, V. Tzoumas, and L. Carlone, “Graduated non-convexity for robust spatial perception: From non-minimal solvers to global outlier rejection,” *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 1127–1134, Apr. 2020.

[45] J. Shi, H. Yang, and L. Carlone, “ROBIN: A graph-theoretic approach to reject outliers in robust estimation using invariants,” in *Proc. IEEE Int. Conf. Robot. Autom.*, 2021, pp. 13820–13827.

[46] J. Li, P. Zhao, Q. Hu, and M. Ai, “Robust point cloud registration based on topological graph and cauchy weighted LQ-norm,” *ISPRS J. Photogrammetry Remote Sens.*, vol. 160, pp. 244–259, Feb. 2020.

[47] J. Li, Q. Hu, and M. Ai, “Point cloud registration based on one-point RANSAC and scale-annealing biweight estimation,” *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 11, pp. 9716–9729, Nov. 2021.

[48] K. Lebeda, J. Matas, and O. Chum, “Fixing the locally optimized RANSAC—Full experimental evaluation,” in *Proc. Brit. Mach. Vis. Conf.*, 2012, pp. 1–11.

[49] P. W. Theiler, J. D. Wegner, and K. Schindler, “Keypoint-based 4-points congruent sets—Automated marker-less registration of laser scans,” *ISPRS J. Photogrammetry Remote Sens.*, vol. 96, pp. 149–163, 2014.

[50] Z. Cai, T.-J. Chin, A. P. Bustos, and K. Schindler, “Practical optimal registration of terrestrial LiDAR scan pairs,” *ISPRS J. Photogrammetry Remote Sens.*, vol. 147, pp. 118–131, 2019.

[51] L. Sun, “RANSIC: Fast and highly robust estimation for rotation search and point cloud registration using invariant compatibility,” *IEEE Robot. Autom. Lett.*, vol. 7, no. 1, pp. 143–150, Jan. 2022.

[52] K. S. Arun, T. S. Huang, and S. D. Blostein, “Least-squares fitting of two 3-D point sets,” *IEEE Trans. Pattern Anal. Mach. Intell.*, no. 5, pp. 698–700, Sep. 1987.

[53] A. P. Bustos and T.-J. Chin, “Guaranteed outlier removal for point cloud registration with correspondences,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 12, pp. 2868–2882, Dec. 2018.

[54] E. Hu and L. Sun, “Daniel: A fast and robust consensus maximization method for point cloud registration with high outlier ratios,” *Inf. Sci.*, vol. 614, pp. 563–579, 2022.

[55] A. Zeng, S. Song, M. Nießner, M. Fisher, J. Xiao, and T. Funkhouser, “3DMatch: Learning local geometric descriptors from RGB-D reconstructions,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1802–1811.

[56] K. Lai, L. Bo, and D. Fox, “Unsupervised feature learning for 3D scene labeling,” in *Proc. IEEE Int. Conf. Robot. Autom.*, 2014, pp. 3050–3057.

[57] R. B. Rusu and S. Cousins, “3D is here: Point cloud library (PCL),” in *Proc. IEEE Int. Conf. Robot. Autom.*, 2011, pp. 1–4.

[58] M. Milanese, *Estimation and Prediction in the Presence of Unknown but Bounded Uncertainty: A. Survey*. Boston, MA, USA: Springer US, 1989, pp. 3–24.

[59] D. Bertsekas, "Control of uncertain systems with a set-membership description of the uncertainty," Ph.D. dissertation, Dept. Elect. Eng., Massachusetts Inst. Tech., Cambridge, MA, USA, 1971.

[60] M. De Berg, O. Cheong, M. Van Kreveld, and M. Overmars, *Computational Geometry: Algorithms and Applications*, 3rd ed. Berlin, Germany: Springer, 2008.

Kuo-Liang Chung (Senior Member, IEEE) received the B.S., M.S., and Ph.D. degrees from Department of Computer Science and Information Engineering at National Taiwan University, Taipei, Taiwan, in 1982, 1984, and 1990, respectively.

He has been a Chair Professor with the Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan, since 2009. His research interests include point cloud registration and image processing.

Dr. Chung was a recipient of the Distinguished Research Award (2004–2007, 2019–2022) from the Ministry of Science and Technology, Taiwan. He has been an Associate Editor for *Journal of Visual Communication and Image Representation* since 2011.

Wei-Tai Chang received the B.S. degree from the Department of Computer Science and Information Engineering at Yuan Ze University, Taoyuan, Taiwan, in 2022. He is currently working toward the M.S. degree in computer science and information engineering with the National Taiwan University of Science and Technology, Taipei, Taiwan.

His research interests include point cloud registration and image processing.