Fuzzy Cluster-based Group-wise Point Set Registration with Quality Assessment

Qianfang Liao, Da Sun, Shiyu Zhang, Amy Loutfi, and Henrik Andreasson

***Abstract*—This article studies group-wise point set registration and makes the following contributions: "FuzzyGReg", which is a new fuzzy cluster-based method to register multiple point sets jointly, and "FuzzyQA", which is the associated quality assessment to check registration accuracy automatically. Given a group of point sets, FuzzyGReg creates a model of fuzzy clusters and equally treats all the point sets as the elements of the fuzzy clusters. Then, the group-wise registration is turned into a fuzzy clustering problem. To resolve this problem, FuzzyGReg applies a fuzzy clustering algorithm to identify the parameters of the fuzzy clusters while jointly transforming all the point sets to achieve an alignment. Next, based on the identified fuzzy clusters, FuzzyQA calculates the spatial properties of the transformed point sets and then checks the alignment accuracy by comparing the similarity degrees of the spatial properties of the point sets. When a local misalignment is detected, a local re-alignment is performed to improve accuracy. The proposed method is cost-efficient and convenient to be implemented. In addition, it provides reliable quality assessments in the absence of ground truth and user intervention. In the experiments, different point sets are used to test the proposed method and make comparisons with state-of-the-art registration techniques. The experimental results demonstrate the effectiveness of our method. The code is available at https://gitsvn-nt.oru.se/qianfang.liao/FuzzyGRegWithQA**

***Index Terms*—group-wise registration, registration quality assessment, joint alignment, fuzzy clusters, 3D point sets.**

I. Introduction

POINT set registration finds spatial transformations to align sets of points (e.g., point clouds and range scans). It plays a critical role in various applications, such as computer vision, robotics, medical image analysis, etc. This article focuses on 3D rigid point set registration, where a transformation is composed of a rotation and a translation in 3D space.

Numerous registration methods have been developed, most of which are used for pair-wise registration aiming to align two point sets. In a pair-wise registration, one point set is fixed as the reference, and the other point set is moved/transformed to align with the reference set, as shown in Fig. 1(a). Generally, the transformation is derived by optimizing a certain metric/objective function that scores the alignment degree, such as point distance-based metrics [1]–[6] and different model-based metrics [7]–[16]. When minimizing/maximizing a metric, a smaller/greater value of the metric usually implies a better registration quality. However, without ground truth and user intervention, most registration methods cannot tell whether the point sets are correctly aligned solely based on the values of their metrics. Recently, several approaches [6], [15], [17] have been proposed that utilize different spatial properties of points to assess pair-wise registration qualities in the absence of ground truth and user intervention. These approaches further advance registration studies toward full automation.

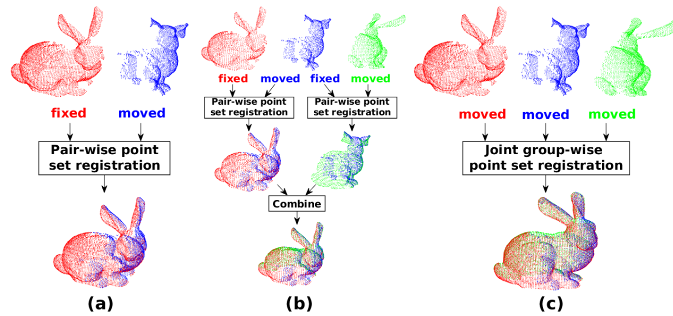


Fig. 1. Examples of point set registration using the Stanford bunny data[2]. **(a)**. A pair-wise registration, where the red point set is fixed as the reference and the blue point set is moved for the alignment. **(b)**. A sequential registration, which performs two pair-wise registrations to align three point sets. **(c)**. A joint registration, which moves all the three point sets on an equal footing.

Compared to pair-wise registration, group-wise registration is a more challenging problem in which multiple point sets need to be aligned (e.g., 3D reconstruction). Early methods to handle this problem are sequentially aligning multiple point set pairs and combining the results [18]–[20], as the example shown in Fig. 1(b). However, these methods generally lead to cumulative errors. To cope with this issue, several strategies have been developed, such as using different networks to organize the multiple pair-wise registrations, adding various post-processing phases to globally tune the local pair-wise registration results, and estimating the transformations of all the point sets jointly [21]–[36]. Note that the joint estimations of transformations, like Fig. 1(c), usually work better in giving unbiased results when compared to the methods based on multiple pair-wise registrations that take one point set as the reference at each time [30]. Despite significant contributions, existing group-wise registration methods often have inconvenient requirements, such as predefined closed loops of point sets, true correspondences of point sets, lengthy training phases, and expensive operations (exponentials, etc.). In addition, few of them consider registration quality assessments in the absence of ground truth and user intervention. The recently proposed pair-wise registration quality assessments [6], [15], [17] can be applied to sequentially check the alignment accu-

This work was supported by VINNOVA under the project TAMMP (2019-05878) and by FORMAS under the project RoboClean (2019-02264). (*Corresponding author: Da Sun.*)

The authors are with the Centre for Applied Autonomous Sensor Systems (AASS), Örebro University, Sweden. E-mail: {Qianfang.Liao, Da.Sun, Shiyu.Zhang, Amy.Loutfi, Henrik.Andreasson}@oru.se

[2]http://graphics.stanford.edu/data/3Dscanrep/

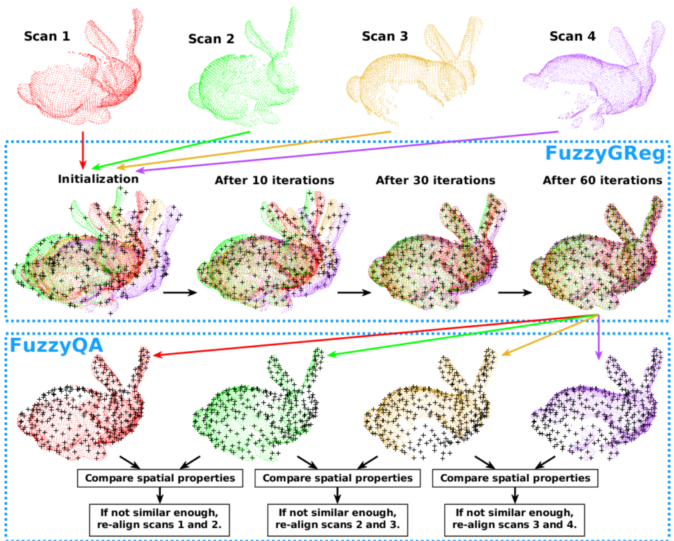


Fig. 2. An example of using the proposed method to align four bunny scans. The first row shows the initial poses of the four scans. The second row shows the registration process of FuzzyGReg, where each black "+" denotes a fuzzy cluster center. For the initialization, FuzzyGReg randomly selects 300 points from the four scans as the initial fuzzy cluster centers. Then, it iteratively updates the fuzzy cluster centers and the scan poses to achieve an alignment. Based on the results of FuzzyGReg, the third row shows the working principle of FuzzyQA: For every two consecutive scans, their spatial properties with respect to the 300 fuzzy cluster centers are calculated and compared. If the spatial properties are similar enough, the two consecutive scans are regarded as aligned; Otherwise, the two scans are re-aligned using FuzzyGReg.

racy of overlapping point set pairs in a group-wise registration. Nevertheless, these pair-wise assessments are independent of the group-wise registration methods and are based on some conditions usually not included in the group-wise registrations, thus requiring extra costs. Therefore, it is desirable to have an effective group-wise registration method with a convenient and cost-efficient registration quality assessment.

In this study, we propose a fuzzy cluster-based group-wise registration method named "FuzzyGReg" and the associated fuzzy cluster-based registration quality assessment named "FuzzyQA". Given a group of point sets, FuzzyGReg creates a model of fuzzy clusters and equally treats all the point sets as the elements of the fuzzy clusters. Then, the group-wise registration is turned into a fuzzy clustering problem. To resolve this problem, FuzzyGReg applies a fuzzy clustering algorithm to minimize a fuzzy cluster-based metric via an iterative calculation. This iterative calculation alternates between estimating the parameters of the fuzzy clusters based on the current transformed point sets and jointly transforming all the point sets toward a group-wise alignment based on the current fuzzy clusters until the termination condition is met. Next, FuzzyQA calculates the spatial property of each transformed point set with respect to the estimated fuzzy clusters and then checks the alignment accuracy based on the similarity degrees of the spatial properties, where no ground truth or user intervention is required. When a local misalignment is detected, a local re-alignment is applied to improve accuracy. Fig. 2 shows an example of the process. The proposed method does not have inconvenient requirements and takes low computational costs since the number of parameters to be calculated is small, and no expensive operations are involved. In addition, it has a reliable registration quality assessment technique that can be conveniently implemented based on the group-wise registration results. In the experiments, different point sets are used to test and compare our method with state-of-the-art registration techniques to demonstrate its effectiveness.

II. Related Work

In this section, we review some existing methods that are related to our work. Most existing methods of point set registration are developed for pair-wise registration, where the Iterative Closest Point (ICP) [1] is the best-known method. ICP aligns two point sets by iteratively minimizing the nearest-point distances, and it has several variants, such as the trimmed ICP [2], the sparse ICP [3], and the fast and robust ICP [4]. There are other point distance-based methods [5], [6] that minimize the distances between true corresponding points to realize alignments. Usually, the true correspondences of point sets are built based on feature descriptors [37]. Apart from the point distance-based methods, some studies represent point sets by certain models and then optimize the model-based metrics to attain alignments, such as the deep neural network-based methods match point sets based on the encoded geometric properties of points [7]–[11]; the probabilistic methods align point sets by maximizing some likelihood functions or minimizing the divergences between probability distributions of points [12]–[14]; and the fuzzy cluster-based methods achieve registrations via minimizing the distance losses of points around fuzzy cluster centers [15], [16].

For group-wise registration, an intuitive way is to sequentially align point set pairs and merge the results together [18]–[20]. However, this way suffers from cumulative errors. A number of strategies have been developed to handle this issue. The first strategy is proposed in [21], where the point sets are organized in a star network to perform the multiple pair-wise registrations. Then, any two point sets are linked by at most two edges to reduce the cumulative errors. Many other strategies follow a two-stage pipeline: The first stage is to collect local pair-wise registration results, and the second stage is to apply a global refinement. In [22], [23], optimizations are applied to remove inconsistent local matches and achieve globally consistent results. In [29], a Bayesian framework is proposed to align multiple point sets, where pair-wise correspondences are regarded as missing data and inferred through a maximum posteriori process. In [24], [25], with the principle that the total motion of the point sets along a cycle is zero, the pair-wise transformations are tuned based on cycles of the point set graph to balance the global error. Similarly, a method of Lie algebraic averaging of local motions is given in [26], which extends ICP to simultaneously register multiple point sets. This method is revised by [27] where the trimmed ICP [2] replaces ICP for higher robustness against non-overlapping outliers. Also, a weighted motion averaging method is presented in [28], which considers that different local motions have different contributions to the global alignment. A drawback of these methods is that they need to predefine closed loops of point sets.

In addition to the above two-stage strategies, some studies present joint group-wise registrations [30]–[33], which treat all the point sets on an equal footing to give unbiased results in the transformation estimations. In [30], the Joint Registration of Multiple Point Clouds (JRMPC) is proposed that considers all the point sets to be drawn from a Gaussian Mixture Model (GMM); Then, it applies an Expectation-Maximization (EM) algorithm to estimate the parameters of GMM while jointly transforming all the point sets to reach an alignment. The method in [31] revises JRMPC by replacing the GMM with a Student's-t mixture model. The EM Perspective for Multi-view Registration (EMPMR) is developed in [32] considering that each point is generated from one unique GMM, where its nearest neighbors in other point sets are treated as Gaussian centroids with equal covariance and membership probabilities; then, a framework of maximum likelihood estimation is built and optimized by an EM algorithm. Generally, the probabilistic methods need to calculate a relatively large number of parameters to obtain the probability distributions and require expensive operations like exponentials in their optimizations. In [33], the Joint Pair-wise Registration (JPR) is described that minimizes the pair-wise point-to-plane distances in a joint fashion to calculate the transformations. However, this method needs to manually select a relatively large number of parameters. Besides, it needs to know the normals of points and is sensitive to the noise contained in the normals.

Some true correspondence-based and deep learning-based methods can be found, such as [34] where an optimization-on-a-manifold method is developed to estimate the transformations of all the point sets based on their true correspondences; and [35], [36] where a deep neural network-based system and an end-to-end learnable algorithm are respectively proposed to align multiple point sets. Based on true correspondences or learned features of point sets, the registration methods usually can better handle low overlaps of point sets. However, true correspondences are not always easy to obtain. For example, the correspondences built by local feature descriptors [37] may contain a large portion of incorrect matches and may be relatively uninformative (repeated structures, many flat surfaces, etc.) [38]. The deep neural network-based methods often include lengthy training phases and are restricted in generalizing to untrained data. Although the system in [35] does not expect generalization since its training process is equivalent to solving the registration problem, its training takes a long time to converge without a good initialization.

In practice, incorrect alignments may occur due to different uncertainties (local optima, noise, etc.), and most existing registration methods cannot perceive the incorrectness without ground truth or user intervention. Recently, a few studies have devoted efforts to resolving this problem. The method called CorAl in [17] checks the pair-wise registration accuracy based on the joint and separate entropy of point sets, where a classifier needs to be pre-trained to detect misalignment. In [6], a pair-wise registration method called TEASER++ is developed that can certify the optimum of its registration metric. This method relies on knowing the true correspondences of point sets, which may not be obtainable as described above. In [15], a fuzzy cluster-based pair-wise registration quality assessment is proposed, which compares the average fuzzy cluster-based distance losses of point sets to determine a coarse alignment that poses a good initialization for refinement. This quality assessment is limited by requiring prior knowledge of the overlapping ratio between point sets. For group-wise registration, an uncertainty quantification method is developed in [33] that calculates the covariance matrices of outputs to estimate registration errors. However, this method assumes that the correspondence sets used in its estimation are subsets of ground truth correspondences, making it limited in practical applications. Therefore, it is desired to have a more effective quality assessment for group-wise registration.

The following nomenclature lists important symbols used in the remainder of this article.

NOMENCLATURE

$\mathcal{P}=\{\mathbb{P}_i\}_{i=1}^{N_P}$	A point set group having N_P point sets.
$\mathbb{P}_i=\{\boldsymbol{p}_{ij}\}_{j=1}^{N_{P_i}}$	The ith point set having N_{P_i} points $\boldsymbol{p}_{ij}$.
$\boldsymbol{\Theta}=\{\boldsymbol{\theta}_i\}_{i=1}^{N_P}$	The N_P transformations of $\mathcal{P}$.
$\boldsymbol{\theta}_i=(\boldsymbol{R}_i,\boldsymbol{t}_i)$	The transformation of $\mathbb{P}_i$ that consists of a rotation $\boldsymbol{R}_i$ and a translation $\boldsymbol{t}_i$.
$T(\boldsymbol{p}_{ij},\boldsymbol{\theta}_i)$ or $T\boldsymbol{p}_{ij}$	The transformed point $\boldsymbol{p}_{ij}$ using $\boldsymbol{\theta}_i$.
$T(\mathbb{P}_i,\boldsymbol{\theta}_i)$ or $T\mathbb{P}_i$	The transformed point set $\mathbb{P}_i$ using $\boldsymbol{\theta}_i$.
$T(\mathcal{P},\boldsymbol{\Theta})$	The transformed point sets $\{T\mathbb{P}_i\}_{i=1}^{N_P}$.
$\mathbb{C}=\{\boldsymbol{c}_k\}_{k=1}^{N_C}$	The N_C fuzzy cluster centers.
$\mu_{\boldsymbol{c}_k}(\boldsymbol{p}_{ij})$	The fuzzy membership of point $\boldsymbol{p}_{ij}$ in the fuzzy cluster centered at $\boldsymbol{c}_k$.
$\mathbb{C}^{(vt)}=\{\boldsymbol{c}_k^{(vt)}\}_{k=1}^{N_C}$	The N_C virtual fuzzy cluster centers calculated based on a point set $T\mathbb{P}_i$.
$\hat{\boldsymbol{\theta}}_i=(\hat{\boldsymbol{R}}_i,\hat{\boldsymbol{t}}_i)$	The transformation aligns $\mathbb{C}^{(vt)}$ to $\mathbb{C}$.
N_{Itr}	The predefined number of iterations.
$\mathbb{C}_i$	A subset of $\mathbb{C}$. Each of the fuzzy clusters centered at $\mathbb{C}_i$ contains the points of $T\mathbb{P}_i$ more than the average number.
$\mathbb{C}_{i(i+1)}$	The intersection set of $\mathbb{C}_i$ and $\mathbb{C}_{i+1}$.
μ_{th}	The fuzzy membership threshold defining the main ranges of fuzzy clusters.
$T\hat{\mathbb{P}}_i=\{T\hat{\boldsymbol{p}}_{ij}\}_{j=1}^{N_{\hat{P}_i}}$	A subset of $T\mathbb{P}_i$, in which the fuzzy memberships of each point in at least one of the fuzzy clusters centered at $\mathbb{C}_{i(i+1)}$ exceed μ_{th}.
$\boldsymbol{F}_{ik}$	The fuzzy covariance matrix calculated based on $T\hat{\mathbb{P}}_i$ with respect to $\boldsymbol{c}_k$.
$d_{i(i+1)}(k)$	The covariance matrix distance between $\boldsymbol{F}_{ik}$ and $\boldsymbol{F}_{(i+1)k}$.
$\overline{d}_{i(i+1)}$	The average value of $d_{i(i+1)}(k)$ of the elements in $\mathbb{C}_{i(i+1)}$.
$\overline{d}_{ub}$	The upper bound of $\overline{d}_{i(i+1)}$ for two aligned point sets.

III. PRELIMINARY KNOWLEDGE

Given N_P point sets, $\mathcal{P}=\{\mathbb{P}_i\}_{i=1}^{N_P}$, where $\mathbb{P}_i=\{\boldsymbol{p}_{ij}\in\mathbb{R}^3\}_{j=1}^{N_{P_i}}$ denotes the ith point set containing N_{P_i} points ($\boldsymbol{p}_{ij}$ is the 3D point coordinate), the group-wise registration aims to find a transformation for each $\mathbb{P}_i$, denoted by $\boldsymbol{\theta}_i$, such that the N_P transformed point sets, denoted as $T(\mathcal{P},\boldsymbol{\Theta})=$

$\{T(\mathbb{P}_i, \boldsymbol{\theta}_i)\}_{i=1}^{N_P}$, are correctly aligned together. The parameters are explained in the following. The transformation $\boldsymbol{\theta}_i = (\boldsymbol{R}_i, \boldsymbol{t}_i)$ is composed of a rotation matrix $\boldsymbol{R}_i \in SO(3)$ and a translation vector $\boldsymbol{t}_i \in \mathbb{R}^3$; $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_i\}_{i=1}^{N_P}$ denotes the set of the N_P transformations; $T(\mathbb{P}_i, \boldsymbol{\theta}_i) = \{T(\boldsymbol{p}_{ij}, \boldsymbol{\theta}_i)\}_{j=1}^{N_{P_i}}$ denotes the transformed point set $\mathbb{P}_i$ using $\boldsymbol{\theta}_i$; and the transformed point $T(\boldsymbol{p}_{ij}, \boldsymbol{\theta}_i)$ is derived by $T(\boldsymbol{p}_{ij}, \boldsymbol{\theta}_i) = \boldsymbol{R}_i \cdot \boldsymbol{p}_{ij} + \boldsymbol{t}_i$.

The proposed group-wise registration method is based on fuzzy clusters. Fuzzy clusters are effective and robust models that have been used in various applications [39]–[43], including pair-wise point set registrations [15], [16]. In this study, the proposed method is developed based on the well-known fuzzy c-means (FCM) clustering algorithm [44] to distill natural groupings of points and create fuzzy clusters. Given a point set $\mathbb{P} = \{\boldsymbol{p}_j \in \mathbb{R}^3\}_{j=1}^{N}$, we suppose that it is described by N_C fuzzy clusters. Usually, N_C is much smaller than N. The fuzzy cluster centers (also called centroids or prototypes) are denoted as $\mathbb{C} = \{\boldsymbol{c}_k \in \mathbb{R}^3\}_{k=1}^{N_C}$. FCM minimizes the following cost function to locate $\mathbb{C}$ and create fuzzy clusters:

$$\min_{\mathbb{C}} \left\{ J(\mathbb{P}, \mathbb{C}) = \sum_{j=1}^{N} \sum_{k=1}^{N_C} \mu_{\boldsymbol{c}_k}(\boldsymbol{p}_j)^m \cdot \|\boldsymbol{p}_j - \boldsymbol{c}_k\|^2 \right\} \quad (1)$$
$$s.t.\ 0 \le \mu_{\boldsymbol{c}_k}(\boldsymbol{p}_j) \le 1 \text{ and } \sum\nolimits_{k=1}^{N_C} \mu_{\boldsymbol{c}_k}(\boldsymbol{p}_j) = 1$$

where $m > 1$ governs the fuzziness degree and $m = 2$ is widely used; $\mu_{\boldsymbol{c}_k}(\boldsymbol{p}_j)$ denotes the fuzzy membership grade of point $\boldsymbol{p}_j$ in the fuzzy cluster centered at $\boldsymbol{c}_k$.

FCM is an unsupervised learning algorithm. Generally, it randomly selects N_C points from the point set $\mathbb{P}$ as the initial fuzzy cluster centers and then minimizes the cost function $J(\mathbb{P}, \mathbb{C})$ in (1) via an iterative calculation. In each iteration, first, the fuzzy memberships of the points are calculated based on the current fuzzy cluster centers as follows:

$$\mu_{\boldsymbol{c}_k}(\boldsymbol{p}_j) = \begin{cases} 1, & \text{if } \|\boldsymbol{p}_j - \boldsymbol{c}_k\| = 0 \\ 0, & \text{if } \exists_{r \ne k}\ \|\boldsymbol{p}_j - \boldsymbol{c}_r\| = 0 \\ \frac{1}{\sum_{r=1}^{N_C} \left(\frac{\|\boldsymbol{p}_j - \boldsymbol{c}_k\|}{\|\boldsymbol{p}_j - \boldsymbol{c}_r\|} \right)^{\frac{2}{m-1}}}, & \text{else} \end{cases} \quad (2)$$

Note that the fuzzy memberships calculated by (2) satisfy the constraints in (1). Then, based on the fuzzy memberships, the fuzzy cluster centers are updated by the following:

$$\boldsymbol{c}_k = \frac{\sum_{j=1}^{N} \mu_{\boldsymbol{c}_k}(\boldsymbol{p}_j)^m \cdot \boldsymbol{p}_j}{\sum_{j=1}^{N} \mu_{\boldsymbol{c}_k}(\boldsymbol{p}_j)^m},\ k = 1, \cdots, N_C \quad (3)$$

By repeating (2) and (3), the fuzzy cluster centers gradually converge to the optimal positions that minimize the cost function $J(\mathbb{P}, \mathbb{C})$ in (1). The iterative calculation of FCM stops when the termination condition is satisfied, such as reaching the predefined iteration number N_{Itr}.

Similar to many existing group-wise registration methods, the proposed method uses local optimizations to align point sets, which means that the initial pose differences between neighboring point sets in $\mathcal{P}$ should not exceed a certain range. In addition, the proposed method needs the overlapping regions of neighboring point sets to reach a certain level. Thus, we apply the following assumption in this article:

Assumption 1: Every two consecutive point sets in $\mathcal{P}$ have sufficient overlaps, and their initial pose differences are within the local convergence basin of the proposed method.

Generally, when a depth sensor smoothly moves in an environment to collect point sets (range scans), two successive poses of the sensor to generate point sets are usually not far away from each other. Also, when the depth sensor is installed on a mobile robot, the robot usually has odometry systems to record the motions that can be used to provide good initial poses of the point sets. Therefore, Assumption 1 works in many applications. In the experiments, we will test the sensitivities of the proposed method with respect to overlapping ratios and initial poses of point sets.

IV. Methodology

In this section, we elaborate FuzzyGReg and FuzzyQA. Also, a discussion about the proposed method is provided.

A. FuzzyGReg

For a group $\mathcal{P} = \{\mathbb{P}_i\}_{i=1}^{N_P}$, FuzzyGReg creates a model composed of N_C fuzzy clusters to describe the N_P point sets in $\mathcal{P}$, where all the points are equally considered as the elements of the fuzzy clusters. The target of FuzzyGReg is to identify the N_C fuzzy cluster centers, $\mathbb{C}$, and the N_P transformations of the point sets, $\boldsymbol{\Theta}$, such that the transformed point sets, $T(\mathcal{P}, \boldsymbol{\Theta})$, are accurately aligned together to form a consistent group that optimally fits the N_C fuzzy clusters centered at $\mathbb{C}$, like the example shown in Fig. 2.

To reach this target, we need to design an appropriate metric and the associated optimization algorithm for FuzzyGReg such that it can obtain the desired $\mathbb{C}$ and $\boldsymbol{\Theta}$ through optimizing the metric. From the cost function of FCM in (1), when a bunch of points well fit a model of fuzzy clusters, the points and the fuzzy cluster centers achieve the minimum value of this cost function. Inspired by (1), we let the optimization problem of FuzzyGReg be the following:

$$\min_{\mathbb{C}, \boldsymbol{\Theta}} \Bigg\{ J(T(\mathcal{P}, \boldsymbol{\Theta}), \mathbb{C}) = \sum_{i=1}^{N_P} J(T(\mathbb{P}_i, \boldsymbol{\theta}_i), \mathbb{C}) =$$
$$\sum_{i=1}^{N_P} \sum_{j=1}^{N_{P_i}} \sum_{k=1}^{N_C} \mu_{\boldsymbol{c}_k}(T(\boldsymbol{p}_{ij}, \boldsymbol{\theta}_i))^m \cdot \|T(\boldsymbol{p}_{ij}, \boldsymbol{\theta}_i) - \boldsymbol{c}_k\|^2 \Bigg\}$$
$$s.t.\ 0 \le \mu_{\boldsymbol{c}_k}(T(\boldsymbol{p}_{ij}, \boldsymbol{\theta}_i)) \le 1,\ \sum\nolimits_{k=1}^{N_C} \mu_{\boldsymbol{c}_k}(T(\boldsymbol{p}_{ij}, \boldsymbol{\theta}_i)) = 1,$$
$$\boldsymbol{R}_i^T \cdot \boldsymbol{R}_i = \boldsymbol{I}_3,\ \text{and } \det(\boldsymbol{R}_i) = 1 \quad (4)$$

where $J(T(\mathcal{P}, \boldsymbol{\Theta}), \mathbb{C})$ is the metric (total cost) of the optimization problem, and $J(T(\mathbb{P}_i, \boldsymbol{\theta}_i), \mathbb{C})$ is the per-point-set cost. Since each point set is equally treated as the elements of the fuzzy clusters, the total cost is the sum of all the per-point-set costs. $\mu_{\boldsymbol{c}_k}(T(\boldsymbol{p}_{ij}, \boldsymbol{\theta}_i))$ is the fuzzy membership of the point $T(\boldsymbol{p}_{ij}, \boldsymbol{\theta}_i)$ in the fuzzy cluster centered at $\boldsymbol{c}_k$ and can be calculated based on $T(\boldsymbol{p}_{ij}, \boldsymbol{\theta}_i)$ and $\mathbb{C}$ using (2).

To optimize the metric $J(T(\mathcal{P}, \boldsymbol{\Theta}), \mathbb{C})$ in (4), we design an iterative optimization algorithm, which is revised from FCM and adds a step to jointly calculate the transformations of all the point sets in each iteration. The details of this optimization algorithm are the following.

1. Initialization. To perform the optimization, we select the following parameters: The number of fuzzy clusters (N_C), the number of iterations (N_{Itr}), the initial fuzzy cluster centers $\mathbb{C}$, and the initial transformations $\boldsymbol{\Theta}$ of the point sets. The selections of N_C and N_{Itr} are presented in Section IV. C. For $\mathbb{C}$, we randomly choose N_C points from the N_P point sets as the initial values, like the example shown in Fig. 2. For $\boldsymbol{\Theta}$, when N_P is not a large number such that the optimization can be directly applied to align $\mathcal{P}$ based on Assumption 1, we let the initial transformations $\boldsymbol{\Theta}$ be zero motions, where all the rotations are $\boldsymbol{R}_i = \boldsymbol{I}_3$ and all the translations are $\boldsymbol{t}_i = [0\ 0\ 0]^T$. When N_P is a large number, the method to determine the initial $\boldsymbol{\Theta}$ is discussed in Section IV. C.

2. Iterative calculation. At the beginning of each iteration, we have the current fuzzy cluster centers $\mathbb{C}$ and the current transformations $\boldsymbol{\Theta}$ of the point sets. Based on these parameters, each iteration performs two tasks. The first task is to estimate new transformations, $\boldsymbol{\Theta}^{(new)} = \{\boldsymbol{\theta}_i^{(new)}\}_{i=1}^{N_P}$, for the point sets to better fit $\mathbb{C}$. The second task is to estimate new fuzzy cluster centers, $\mathbb{C}^{(new)} = \{\boldsymbol{c}_k^{(new)}\}_{k=1}^{N_C}$, based on the new transformed point sets, $T(\mathcal{P}, \boldsymbol{\Theta}^{(new)})$.

For the first task, note that the new transformation of each point set can be estimated independently. For the current ith point set $T(\mathbb{P}_i, \boldsymbol{\theta}_i)$, based on the current fuzzy cluster centers $\mathbb{C}$, we use (2) to obtain the fuzzy memberships $\mu_{\boldsymbol{c}_k}(T(\boldsymbol{p}_{ij}, \boldsymbol{\theta}_i))$ of its points; Then, according to (3), we can calculate N_C virtual fuzzy cluster centers with respect to $T(\mathbb{P}_i, \boldsymbol{\theta}_i)$, denoted by $\mathbb{C}^{(vt)} = \{\boldsymbol{c}_k^{(vt)}\}_{k=1}^{N_C}$, as follows:

$$\boldsymbol{c}_k^{(vt)} = \frac{\sum_{j=1}^{N_{P_i}} \mu_{\boldsymbol{c}_k}(T(\boldsymbol{p}_{ij}, \boldsymbol{\theta}_i))^m \cdot T(\boldsymbol{p}_{ij}, \boldsymbol{\theta}_i)}{\sum_{j=1}^{N_{P_i}} \mu_{\boldsymbol{c}_k}(T(\boldsymbol{p}_{ij}, \boldsymbol{\theta}_i))^m} \tag{5}$$

If $T(\mathbb{P}_i, \boldsymbol{\theta}_i)$ perfectly fits $\mathbb{C}$, we will have $\mathbb{C}^{(vt)} = \mathbb{C}$. Nevertheless, this condition is usually not satisfied. Thus, we next find a transformation, denoted by $\hat{\boldsymbol{\theta}}_i = (\hat{\boldsymbol{R}}_i, \hat{\boldsymbol{t}}_i)$, for $\mathbb{C}^{(vt)}$ to align with $\mathbb{C}$. Consequently, $\hat{\boldsymbol{\theta}}_i$ can move $T(\mathbb{P}_i, \boldsymbol{\theta}_i)$ to better fit $\mathbb{C}$. This transformation can be derived by resolving the following weighted least squares problem:

$$\begin{aligned} &\min_{\hat{\boldsymbol{\theta}}_i = (\hat{\boldsymbol{R}}_i, \hat{\boldsymbol{t}}_i)} \sum_{k=1}^{N_C} w_{ik} \cdot \|\hat{\boldsymbol{R}}_i \cdot \boldsymbol{c}_k^{(vt)} + \hat{\boldsymbol{t}}_i - \boldsymbol{c}_k\|^2 \\ &s.t.\ (\hat{\boldsymbol{R}}_i)^T \cdot \hat{\boldsymbol{R}}_i = \boldsymbol{I}_3 \text{ and } \det(\hat{\boldsymbol{R}}_i) = 1 \end{aligned} \tag{6}$$

where the weight is $w_{ik} = \sum_{j=1}^{N_{P_i}} \mu_{\boldsymbol{c}_k}(T(\boldsymbol{p}_{ij}, \boldsymbol{\theta}_i))^m$. According to the study in [45], the weighted least squares problem in (6) has a closed-form solution as follows:

$$\begin{cases} \hat{\boldsymbol{R}}_i = \boldsymbol{U}_i \cdot \boldsymbol{S}_i \cdot (\boldsymbol{V}_i)^T \\ \hat{\boldsymbol{t}}_i = \sum_{k=1}^{N_C} w_{ik} \cdot (\boldsymbol{c}_k - \hat{\boldsymbol{R}}_i \cdot \boldsymbol{c}_k^{(vt)}) / \sum_{k=1}^{N_C} w_{ik} \end{cases} \tag{7}$$

where $\boldsymbol{S}_i = \mathrm{diag}(1, 1, \det(\boldsymbol{U}_i)\det(\boldsymbol{V}_i)) \in \mathbb{R}^{3\times 3}$; $\boldsymbol{U}_i$ and $\boldsymbol{V}_i$ are the left and right matrices derived from the singular value decomposition of the term $\boldsymbol{C}\boldsymbol{\Lambda}_i(\boldsymbol{C}^{(vt)})^T$, in which $\boldsymbol{C} = [\boldsymbol{c}_1\ \boldsymbol{c}_2\ \cdots\ \boldsymbol{c}_{N_C}] \in \mathbb{R}^{3\times N_C}$, $\boldsymbol{C}^{(vt)} = [\boldsymbol{c}_1^{(vt)}\ \boldsymbol{c}_2^{(vt)}\ \cdots\ \boldsymbol{c}_{N_C}^{(vt)}] \in \mathbb{R}^{3\times N_C}$, and $\boldsymbol{\Lambda}_i = \mathrm{diag}(w_{i1}, w_{i2}, \cdots, w_{iN_C}) \in \mathbb{R}^{N_C \times N_C}$. Then, the new transformation of $\mathbb{P}_i$, $\boldsymbol{\theta}_i^{(new)} = (\boldsymbol{R}_i^{(new)}, \boldsymbol{t}_i^{(new)})$, can be derived by combining $\hat{\boldsymbol{\theta}}_i = (\hat{\boldsymbol{R}}_i, \hat{\boldsymbol{t}}_i)$ with the current transformation $\boldsymbol{\theta}_i = (\boldsymbol{R}_i, \boldsymbol{t}_i)$ as follows:

$$\begin{cases} \boldsymbol{R}_i^{(new)} = \hat{\boldsymbol{R}}_i \cdot \boldsymbol{R}_i \\ \boldsymbol{t}_i^{(new)} = \hat{\boldsymbol{R}}_i \cdot \boldsymbol{t}_i + \hat{\boldsymbol{t}}_i \end{cases} \tag{8}$$

By repeating (5)-(8) for each point set $\mathbb{P}_i$, $\boldsymbol{\Theta}^{(new)}$ is obtained. The first task is complete.

For the second task, based on $T(\mathcal{P}, \boldsymbol{\Theta}^{(new)})$ and the fuzzy memberships $\mu_{\boldsymbol{c}_k}(T(\boldsymbol{p}_{ij}, \boldsymbol{\theta}_i))$ calculated in the first task, $\mathbb{C}^{(new)}$ can be estimated by the following revised from (3):

$$\boldsymbol{c}_k^{(new)} = \frac{\sum_{i=1}^{N_P} \sum_{j=1}^{N_{P_i}} \mu_{\boldsymbol{c}_k}(T(\boldsymbol{p}_{ij}, \boldsymbol{\theta}_i))^m \cdot T(\boldsymbol{p}_{ij}, \boldsymbol{\theta}_i^{(new)})}{\sum_{i=1}^{N_P} \sum_{j=1}^{N_{P_i}} \mu_{\boldsymbol{c}_k}(T(\boldsymbol{p}_{ij}, \boldsymbol{\theta}_i))^m} \tag{9}$$

The second task is complete.

After completing the above two tasks, the current parameters are updated by $\mathbb{C} \leftarrow \mathbb{C}^{(new)}$ and $\boldsymbol{\Theta} \leftarrow \boldsymbol{\Theta}^{(new)}$. Then, the next iteration starts if the termination condition is not met. As the example shown in Fig. 2, the iterative calculation of FuzzyGReg gradually changes the fuzzy cluster centers and the poses of the point sets toward a group-wise alignment.

3. Termination. When the number of performed iterations reaches N_{Itr}, the iterative calculation stops, and the current $\mathbb{C}$ and $\boldsymbol{\Theta}$ are the solutions to the optimization problem (4).

The above optimization process of FuzzyGReg is summarized in Algorithm 1. Based on the results of FuzzyGReg, next, we introduce FuzzyQA, which can automatically detect misalignment and perform re-alignment to improve accuracy.

Algorithm 1: FuzzyGReg

```
Input: P.
Output: ℂ and Θ to align P.
Select N_C, N_Itr, the initial ℂ (N_C points randomly
  picked from P), and the initial Θ (zero motions).
l ← 1.
while l ≤ N_Itr do
    Based on ℂ, calculate θ_i^(new) of each ℙ_i by (5)-(8)
      to constitute Θ^(new).
    Based on T(P, Θ^(new)), calculate ℂ^(new) by (9).
    Let ℂ ← ℂ^(new), Θ ← Θ^(new), and l ← l + 1.
```

B. FuzzyQA

FuzzyQA checks the registration quality based on the spatial properties of the transformed point sets with respect to the fuzzy clusters estimated by FuzzyGReg. For simplicity, in the following, we use $T\mathbb{P}_i$ and $T\boldsymbol{p}_{ij}$ to denote $T(\mathbb{P}_i, \boldsymbol{\theta}_i)$ and $T(\boldsymbol{p}_{ij}, \boldsymbol{\theta}_i)$, respectively. Considering two consecutive point sets in $T(\mathcal{P}, \boldsymbol{\Theta})$, $T\mathbb{P}_i$ and $T\mathbb{P}_{i+1}$ ($1 \le i \le N_P - 1$), they usually partially overlap each other, and their overlapping regions will display the same spatial/geometric properties when they are aligned. Thus, for each pair of $T\mathbb{P}_i$ and $T\mathbb{P}_{i+1}$, FuzzyQA finds out their overlapping parts and calculates the fuzzy

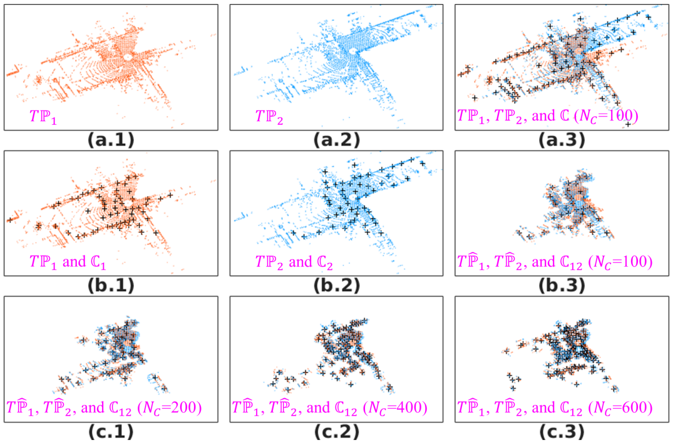


Fig. 3. Examples of selecting (main) overlapping parts of point sets, where each black "+" denotes a fuzzy cluster center. **(a.1)**-**(b.3)** show the selection process with $N_C = 100$, and **(c.1)**-**(c.3)** show different selected results of the same point sets with different N_C, where $\mu_{th} = 1/\sqrt{N_C}$.

Algorithm 2: FuzzyQA

Input: $\mathcal{P}$ and the results of FuzzyGReg ($\mathbb{C}$, $\boldsymbol{\Theta}$, and fuzzy memberships $\mu_{\boldsymbol{c}_k}(T\boldsymbol{p}_{ij})$).

Output: $\boldsymbol{\Theta}$ (updated if necessary) to align $\mathcal{P}$.

Select μ_{th} (can be $1/\sqrt{N_C}$) and $\bar{d}_{ub}$ (can be 0.015) .
$i \leftarrow 1$.
while $i \leq N_P - 1$ **do**
  Based on the fuzzy memberships of the points of $T\mathbb{P}_i$ and $T\mathbb{P}_{i+1}$, select $\mathbb{C}_{i(i+1)}$ from $\mathbb{C}$.
  Based on $\mathbb{C}_{i(i+1)}$ and μ_{th}, select $T\hat{\mathbb{P}}_i$ and $T\hat{\mathbb{P}}_{i+1}$.
  Calculate $\boldsymbol{F}_{ik}$, $\boldsymbol{F}_{(i+1)k}$, and $d_{i(i+1)}(k)$ for each $\boldsymbol{c}_k$ in $\mathbb{C}_{i(i+1)}$ using (10) and (11).
  Calculate $\bar{d}_{i(i+1)}$ by (12).
  if $\bar{d}_{i(i+1)} > \bar{d}_{ub}$ **then**
    Re-align $\mathbb{P}_i$ and $\mathbb{P}_{i+1}$ by FuzzyGReg, and then update $\boldsymbol{\Theta}$ accordingly.
  $i \leftarrow i + 1$.

cluster-based spatial properties of the overlapping parts; Then, the registration quality is assessed based on the similarity degrees of the spatial properties. If the spatial properties are diverse, FuzzyGReg is applied to re-align $\mathbb{P}_i$ and $\mathbb{P}_{i+1}$.

To derive the overlapping regions of two consecutive point sets in $T(\mathcal{P}, \boldsymbol{\Theta})$, we design an approach based on the fuzzy memberships calculated in the last iteration of FuzzyGReg as follows. For $T\mathbb{P}_i$, we assign each of its points to the fuzzy cluster in which the point has the largest fuzzy membership. Then, we know the number of points $T\boldsymbol{p}_{ij}$ in each of the N_C fuzzy clusters and the average value of the N_C numbers. Next, we pick out the fuzzy clusters with the points $T\boldsymbol{p}_{ij}$ more than the average number and denote the centers of these fuzzy clusters as $\mathbb{C}_i$ ($\mathbb{C}_i \subset \mathbb{C}$). Given two consecutive point sets $T\mathbb{P}_i$ and $T\mathbb{P}_{i+1}$, we have $\mathbb{C}_{i(i+1)} = \mathbb{C}_i \cap \mathbb{C}_{i+1}$. In each of the fuzzy clusters centered at $\mathbb{C}_{i(i+1)}$, the points of both $T\mathbb{P}_i$ and $T\mathbb{P}_{i+1}$ are more than their associated average numbers. We set a threshold value of fuzzy memberships, denoted as μ_{th}, to define the main areas of the fuzzy clusters centered at $\mathbb{C}_{i(i+1)}$; and then we select the points of $T\mathbb{P}_i$ and $T\mathbb{P}_{i+1}$ lying in these main areas, where the points have the fuzzy memberships greater than μ_{th} in at least one of the fuzzy clusters centered at $\mathbb{C}_{i(i+1)}$. The selected points, denoted as $T\hat{\mathbb{P}}_i$ and $T\hat{\mathbb{P}}_{i+1}$ ($T\hat{\mathbb{P}}_i \subset T\mathbb{P}_i$, $T\hat{\mathbb{P}}_{i+1} \subset T\mathbb{P}_{i+1}$), are considered as the (main) overlapping regions of $T\mathbb{P}_i$ and $T\mathbb{P}_{i+1}$.

The appropriate value of μ_{th} varies with different N_C. Since the sum of the N_C fuzzy memberships of each point is 1, when a point is considered to be lying in the main area of a certain fuzzy cluster, its fuzzy membership in this fuzzy cluster should be at least greater than $1/N_C$. However, if we set $\mu_{th} = 1/N_C$, this threshold value is not high enough to define the main areas of fuzzy clusters since a point far away from all the fuzzy cluster centers may have a certain fuzzy membership slightly greater than $1/N_C$. Therefore, μ_{th} should be greater than $1/N_C$. An empirical value is $\mu_{th} = 1/\sqrt{N_C}$. As the examples shown in Fig. 3, using $\mu_{th} = 1/\sqrt{N_C}$ provides satisfactory results in selecting overlapping points and selects similar overlapping regions when N_C changes.

Next, we calculate the fuzzy cluster-based spatial properties of the selected overlapping regions. For each of the fuzzy clusters centered at $\mathbb{C}_{i(i+1)}$, based on the points $T\hat{\boldsymbol{p}}_{ij}$ of $T\hat{\mathbb{P}}_i$ and their fuzzy memberships, $\mu_{\boldsymbol{c}_k}(T\hat{\boldsymbol{p}}_{ij})$, calculated in the last iteration of FuzzyGReg, a fuzzy covariance matrix can be obtained by the following function [46]:

$$\boldsymbol{F}_{ik} = \frac{\sum_{j=1}^{N_{\hat{P}_i}} \mu_{\boldsymbol{c}_k}(T\hat{\boldsymbol{p}}_{ij})^m \cdot (T\hat{\boldsymbol{p}}_{ij} - \boldsymbol{c}_k) \cdot (T\hat{\boldsymbol{p}}_{ij} - \boldsymbol{c}_k)^T}{\sum_{j=1}^{N_{\hat{P}_i}} \mu_{\boldsymbol{c}_k}(T\hat{\boldsymbol{p}}_{ij})^m} \tag{10}$$

where $N_{\hat{P}_i}$ is the number of points in $T\hat{\mathbb{P}}_i$. The fuzzy covariance matrix $\boldsymbol{F}_{ik}$ indicates the orientation and shape information of the points $T\hat{\boldsymbol{p}}_{ij}$ around the center $\boldsymbol{c}_k$. By using the way in (10), we can obtain another fuzzy covariance matrix, $\boldsymbol{F}_{(i+1)k}$, for $\boldsymbol{c}_k$ based on the points $T\hat{\boldsymbol{p}}_{(i+1)j}$ of $T\hat{\mathbb{P}}_{i+1}$. If $T\mathbb{P}_i$ and $T\mathbb{P}_{i+1}$ are aligned, $T\hat{\mathbb{P}}_i$ and $T\hat{\mathbb{P}}_{i+1}$ will have the same or similar properties in terms of orientation and shape around each element of $\mathbb{C}_{i(i+1)}$. Consequently, $\boldsymbol{F}_{ik}$ and $\boldsymbol{F}_{(i+1)k}$ will be close to each other. We use the following covariance matrix distance function [47] to measure the similarity between $\boldsymbol{F}_{ik}$ and $\boldsymbol{F}_{(i+1)k}$:

$$d_{i(i+1)}(k) = 1 - \frac{trace(\boldsymbol{F}_{ik} \cdot \boldsymbol{F}_{(i+1)k})}{\|\boldsymbol{F}_{ik}\|_f \cdot \|\boldsymbol{F}_{(i+1)k}\|_f} \tag{11}$$

where $\|\cdot\|_f$ is the Frobenius norm. $d_{i(i+1)}(k) \in [0,\ 1]$ is zero if $\boldsymbol{F}_{ik}$ and $\boldsymbol{F}_{(i+1)k}$ are equal (up to a scaling factor) and one if the two matrices are different to a maximum extent [47]. With (10) and (11), we can obtain $d_{i(i+1)}(k)$ of each element $\boldsymbol{c}_k$ of $\mathbb{C}_{i(i+1)}$ based on $T\hat{\mathbb{P}}_i$ and $T\hat{\mathbb{P}}_{i+1}$. Then, we can have the average value, $\bar{d}_{i(i+1)}$, by the following:

$$\bar{d}_{i(i+1)} = \frac{\sum_{k=1}^{N_{C_{i(i+1)}}} d_{i(i+1)}(k)}{N_{C_{i(i+1)}}} \tag{12}$$

where $N_{C_{i(i+1)}}$ denotes the number of elements in $\mathbb{C}_{i(i+1)}$. Based on $\bar{d}_{i(i+1)}$, we design the following method:

Registration quality assessment and local re-alignment. *Given the group-wise registration results of FuzzyGReg, for*

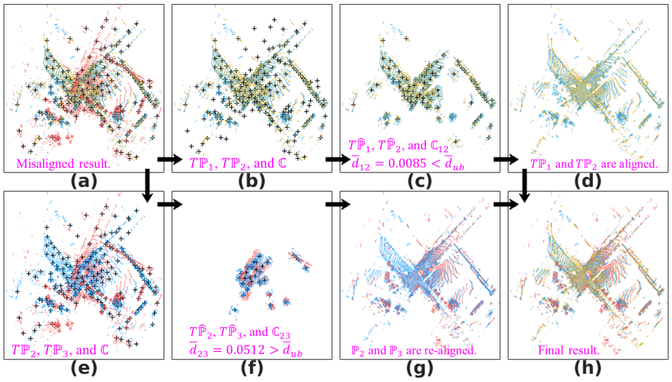


Fig. 4. An example of applying FuzzyQA to three scans $\mathbb{P}_1$ (yellow), $\mathbb{P}_2$ (blue), and $\mathbb{P}_3$ (red), where $N_C = 200$, the fuzzy cluster centers are denoted by black "+", $\mu_{th} = 0.0707$, and $\overline{d}_{ub} = 0.015$. **(a)** shows the misaligned result. **(b)** shows $T\mathbb{P}_1$, $T\mathbb{P}_2$, and $\mathbb{C}$. **(c)** shows the selected overlapping parts, including $T\hat{\mathbb{P}}_1$, $T\hat{\mathbb{P}}_2$, and $\mathbb{C}_{12}$, where $\overline{d}_{12} = 0.0085$. **(d)** shows that $T\mathbb{P}_1$ and $T\mathbb{P}_2$ are regarded as aligned since $\overline{d}_{12} < \overline{d}_{ub}$. **(e)** shows $T\mathbb{P}_2$, $T\mathbb{P}_3$ and $\mathbb{C}$. **(f)** shows the selected overlapping parts, including $T\hat{\mathbb{P}}_2$, $T\hat{\mathbb{P}}_3$, and $\mathbb{C}_{23}$, where $\overline{d}_{23} = 0.0512$. **(g)** shows that $\mathbb{P}_2$ and $\mathbb{P}_3$ are re-aligned since $\overline{d}_{23} > \overline{d}_{ub}$. **(h)** shows the final result, where the three point sets are aligned.

any two consecutive point sets $T\mathbb{P}_i$ and $T\mathbb{P}_{i+1}$ ($1 \leq i \leq N_P - 1$), they are considered as aligned if they satisfy:

$$\overline{d}_{i(i+1)} \leq \overline{d}_{ub} \tag{13}$$

where the upper bound $\overline{d}_{ub}$ is a small positive number. If $T\mathbb{P}_i$ and $T\mathbb{P}_{i+1}$ do not satisfy (13), FuzzyGReg is applied to $\mathbb{P}_i$ and $\mathbb{P}_{i+1}$ to derive new $\boldsymbol{\theta}_i$ and $\boldsymbol{\theta}_{i+1}$ for a local re-alignment, and then $\boldsymbol{\Theta}$ is updated accordingly.

The above process of FuzzyQA is summarized in Algorithm 2, and an example is illustrated in Fig. 4. If the upper bound $\overline{d}_{ub}$ is chosen to be a relatively large value, FuzzyQA may consider an incorrect result as a correct alignment (false positive); and if $\overline{d}_{ub}$ is chosen to be a too small value, FuzzyQA may treat a correct result as misalignment, and then a re-alignment is performed. The latter case takes extra costs due to the unnecessary re-alignment but will not affect registration accuracy. Thus, it would be better to choose a smaller $\overline{d}_{ub}$. The value of $\overline{d}_{ub}$ can be set between $[0.01, 0.015]$. The example shown in Fig. 4 uses $\overline{d}_{ub} = 0.015$.

C. Discussion

In this discussion, first, we describe the advantages of the proposed method. Then, we present some implementation strategies for our method to attain better registration performance in terms of efficiency and practicality.

Comparisons with related methods. The proposed method has several advantages compared to the existing group-wise registration methods. Unlike the multiple pair-wise registration-based methods [18]–[23], FuzzyGReg treats all the point sets on an equal footing to give an unbiased result. Different from the cycle graph-based and motion averaging-based methods [24]–[28], FuzzyGReg considers all the point sets to be generated from a model of fuzzy clusters, which implicitly imposes a loop constraint. Thus, it does not need to predefine closed loops of point sets. Unlike the true correspondence-based and deep neural network-based methods [34]–[36], FuzzyGReg is a true-correspondence-free method, and it neither needs lengthy training phases nor suffers from the generalization issue. Regarding the joint group-wise registrations, compared with JPR [33], FuzzyGReg has much fewer open parameters. Besides, it does not include normals of points and thus is more robust against noises. Compared with the joint probabilistic methods [30]–[32] where each probability component has a series of parameters (mean, variance, mixing proportion, etc.) needing to be calculated, FuzzyGReg is based on fuzzy clusters that are defined by fuzzy cluster centers only. Besides, no expensive operations, like exponentials, are involved in the calculation of FuzzyGReg. Thus, the proposed method has lower computational complexity.

In addition, FuzzyGReg has FuzzyQA to automatically detect misalignment and improve registration accuracy. FuzzyQA is performed based on the results of FuzzyGReg and takes low costs in time and computation. Unlike the CorAl in [17], FuzzyQA does not require a pre-training phase to learn parameters. Different from the certifiable algorithm in [6], FuzzyQA does not rely on true correspondences of point sets. Compared with the fuzzy cluster-based pair-wise assessment in [15] that is used to give a coarse registration result, FuzzyQA is used to check whether the result is a precise alignment and can work without knowing the non-overlapping ratios of point sets. Compared with the uncertainty quantification for group-wise registration in [33] based on the assumption of knowing part of ground truth correspondences, FuzzyQA does not involve impractical assumptions.

The metric of FuzzyGReg in (4) has a similar expression to that of the fuzzy cluster-based pair-wise registration [15], but there are essential differences between the two methods. In the fuzzy cluster-based pair-wise registration [15], the fuzzy cluster centers of one point set, obtained by applying FCM to this point set, are fixed as the reference during the registration process. Then, the pair-wise registration metric, constructed based on these reference centers, is minimized to obtain the transformation that moves the other point set to fit the reference centers and achieve the alignment. When the two point sets partially overlap each other, the pair-wise method [15] may need to know the non-overlapping ratio such that it can perform a trimming on the moved point set to obtain an accurate result. In FuzzyGReg, the fuzzy cluster centers in the metric $J(T(\mathcal{P}, \boldsymbol{\Theta}), \mathbb{C})$ describe all the point sets instead of one point set, and $\mathbb{C}$ is iteratively updated rather than fixed during the registration process, as shown in Fig. 2. Besides, FuzzyGReg does not need point trimming.

Regarding the computational complexity, the fuzzy cluster-based pair-wise method [15] is applied in a coarse-to-fine fashion, in which the fine registration is based on all point-to-point distances (point sets may be down-sampled in this phase); while FuzzyGReg is based on all point-to-center distances. Consequently, to align two point sets with N_{P_1} and N_{P_2} points, respectively, the fine registration of the pair-wise method [15] takes $O(N_{P_1} \cdot N_{P_2})$ operations; while FuzzyGReg using N_C fuzzy clusters takes $O(N_C \cdot N_{P_1} + N_C \cdot N_{P_2})$ operations. To align a group of point sets $\mathcal{P} = \{\mathbb{P}_i\}_{i=1}^{N_P}$, when the pair-wise method [15] is applied to align and combine multiple point set pairs sequentially, like Fig. 1(b), the fine reg-

Algorithm 3: The coarse-to-fine registration strategy

Input: $\mathcal{P}$.

Output: $\mathbf{\Theta}$ to align $\mathcal{P}$.

1 Apply FuzzyGReg with a relatively small N_C to $\mathcal{P}$ to obtain a result $\mathbf{\Theta}$ for a coarse alignment.

2 Apply FuzzyGReg with a relatively large N_C to the coarse result $T(\mathcal{P}, \mathbf{\Theta})$ to obtain a refined $\mathbf{\Theta}$.

3 Apply FuzzyQA to the refined result of FuzzyGReg to update $\mathbf{\Theta}$ if necessary.

Algorithm 4: The local-to-global registration strategy

Input: $\mathcal{P}$.

Output: $\mathbf{\Theta}$ to align $\mathcal{P}$.

1 Split $\mathcal{P}$ into several sequential subgroups, and every two consecutive subgroups share a common point set.

2 Apply the coarse-to-fine strategy in Algorithm 3 to align each subgroup and link all the aligned subgroups together based on the transformations of the common point sets to obtain a result $\mathbf{\Theta}$ for $\mathcal{P}$.

3 Apply FuzzyGReg with a relatively large N_C to the linked result $T(\mathcal{P}, \mathbf{\Theta})$ to obtain a refined $\mathbf{\Theta}$.

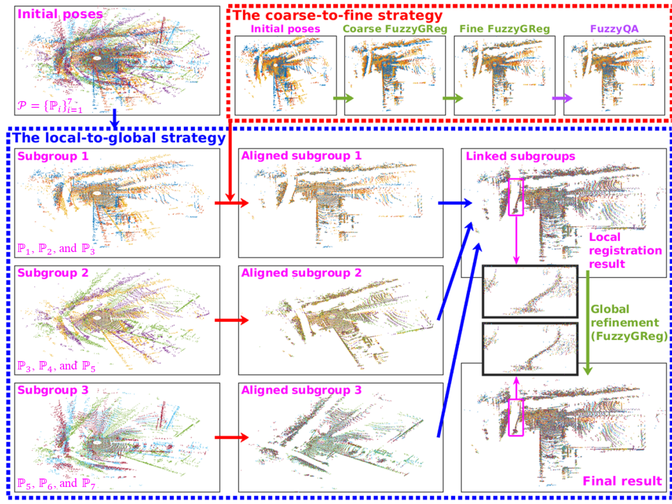


Fig. 5. Examples of Algorithms 3 and 4. The local-to-global strategy is applied to register a group containing seven scans, whose initial poses are shown in the top left sub-figure. This group is split into three subgroups, and every two consecutive subgroups share a common scan. The coarse-to-fine strategy is applied to align each subgroup, and the process of aligning subgroup 1 is shown in the top right sub-figures. The three aligned subgroups are linked together based on the poses of the common scans to obtain the local registration result. Then, FuzzyGReg with a relatively large N_C is applied to refine the local result and give the final result. The enlarged parts of the local and final results show that the global refinement increases the alignment accuracy.

istrations of the pair-wise method need $O(\sum_{i=2}^{N_P} N_{P_{i-1}} \cdot N_{P_i})$ operations; while FuzzyGReg using N_C fuzzy clusters takes $O(N_C \cdot \sum_{i=1}^{N_P} N_{P_i})$ operations. Usually, N_C is much smaller than the point number N_{P_i} of each point set. Therefore, the computational cost of FuzzyGReg can be significantly lower than that of the pair-wise method [15].

Implementation strategies. There is a trade-off issue in FuzzyGReg between accuracy and cost. When N_C is a relatively small number (e.g., dozens), FuzzyGReg takes low time and computational costs but often gives a coarse alignment only; When N_C is a relatively large number (e.g., hundreds or thousands), FuzzyGReg can achieve a precise alignment but requires relatively high costs in time and computation. To relieve this issue, similar to the pair-wise method in [15], the proposed method can also be implemented in a *coarse-to-fine* fashion as follows: First, FuzzyGReg is applied with a relatively small N_C to obtain a coarse result; then, FuzzyGReg is applied again with a relatively large N_C to refine the coarse result; finally, FuzzyQA is applied to the refined result. With this fashion, a coarse alignment can be obtained fast to pose a good initialization for the fine registration; then, the fine registration can converge to the precise alignment with low costs. This coarse-to-fine strategy is described in Algorithm 3.

The selection of N_C depends on the properties of the given point sets, such as the overlapping degrees and the number of point sets (N_P). N_C can be a smaller/greater number when the point sets have relatively high/low overlapping degrees; and N_C can increase/decrease accordingly when N_P increases/decreases. There is no explicit function to choose N_C that can guarantee an accurate alignment for a group of point sets. Nevertheless, when FuzzyGReg is applied in the coarse-to-fine fashion to register two point sets, like the local re-alignment in FuzzyQA, we can specify the following: N_C can be dozens (e.g., 60) in the coarse registration and can be a few hundred (e.g., 200) in the fine registration. Besides, when N_C is a smaller/greater number, the iteration numbers N_{Itr} can be increased/decreased accordingly, such as using $N_{Itr} = 100$ and $N_{Itr} = 80$ in the coarse and fine registrations, respectively. Usually, this selection allows FuzzyGReg to work well in the local re-alignments under Assumption 1. Consequently, even if FuzzyGReg does not give the optimal performance in aligning multiple point sets, we can have a satisfactory result after applying FuzzyQA.

Given a relatively large number of scans collected by a depth sensor moving in a relatively large environment (the origin of the coordinate frame of each scan is the sensor), even though the neighboring scans satisfy Assumption 1, the non-neighboring scans collected at different places of the environment may not have overlaps. If all these scans are directly put in one coordinate frame with zero motions as the initial $\mathbf{\Theta}$, FuzzyGReg usually cannot correctly align them. In this case, the proposed method can be implemented in a *local-to-global* fashion as follows: First, $\mathcal{P}$ is split into a number of sequential subgroups, where each subgroup contains a relatively small number of scans and every two consecutive subgroups share a common scan (the last scan in one subgroup is the first scan in the next subgroup). Next, in the local phase, the coarse-to-fine strategy in Algorithm 3 is applied to register each subgroup with zero motions as the initial $\mathbf{\Theta}$. In this case, FuzzyGReg can generally achieve correct alignments owing to the small number of scans in each subgroup. Even if FuzzyGReg does not align some scans, FuzzyQA can detect the local misalignment and re-align the scans. Since every two consecutive subgroups share a common scan, we can link all

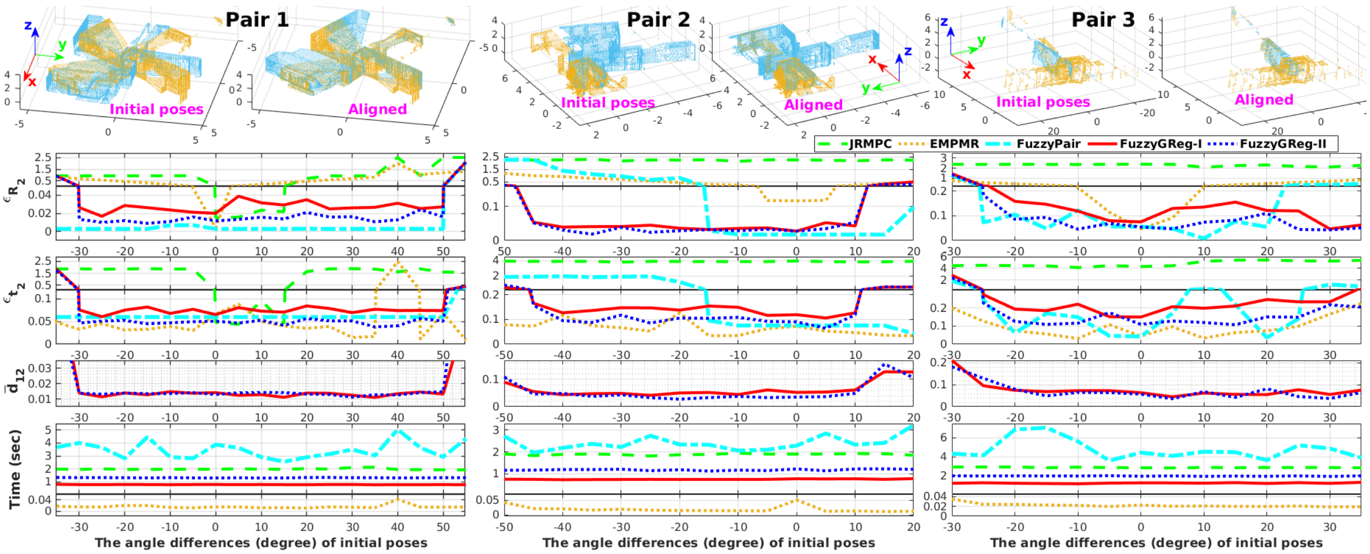


Fig. 6. Results of Test 1, where JRMPC [30], EMPMR [32], FuzzyPair [15], and FuzzyGReg are applied to register three point set pairs selected from [48]. The angle difference of an initial pose means the angle of $\mathbb{P}_2$ (blue) rotated about z-axis to deviate from its alignment with $\mathbb{P}_1$ (yellow) for creating this initial pose. For the aligned Pair 1, the overlapping points takes 73.8% of $\mathbb{P}_1$ and 68.7% of $\mathbb{P}_2$, where $N_{P_1} = 2440$ and $N_{P_2} = 2254$. For the aligned Pair 2, the overlapping points takes 52.2% of $\mathbb{P}_1$ and 34.4% of $\mathbb{P}_2$, where $N_{P_1} = 1578$ and $N_{P_2} = 2635$. For the aligned Pair 3, the overlapping points takes 22.8% of $\mathbb{P}_1$ and 45.5% of $\mathbb{P}_2$, where $N_{P_1} = 4237$ and $N_{P_2} = 1778$.

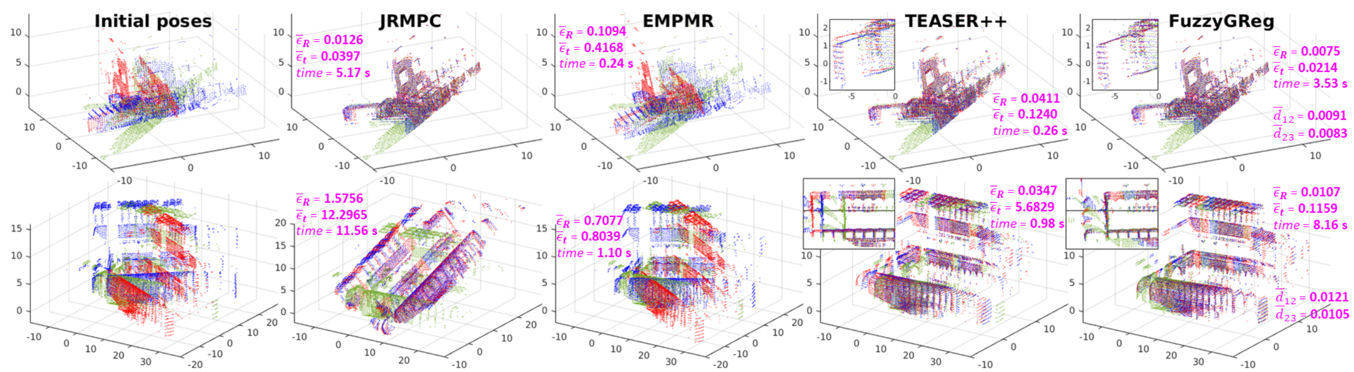


Fig. 7. Results of Test 2, where JRMPC [30], EMPMR [32], TEASER++ [6], and FuzzyGReg are applied to register two groups of point sets selected from [48]. Each group contains three scans. In the first group (in the first row), each N_{P_i} is about 3300 points; the overlapping ratios of the neighboring scans are about 90%; the initial angle differences of every two consecutive scans are about $50°$. In the second group (in the second row), each N_{P_i} is about 6200 points; the overlapping ratios of the neighboring scans are 62% and 85%, and the initial angle differences of every two consecutive scans are about $30°$.

the aligned subgroups based on the transformations of their common scans. Consequently, all the scans can be moved to the poses close to the ground truths without needing to know the sensor positions collecting these scans. Finally, in the global phase, FuzzyGReg with a relatively large N_C (e.g., thousands) is applied to all the moved point sets for further refinement. In this global refinement, N_{Itr} can be a relatively small number (e.g., 30) since the point sets transformed and linked in the local phase are generally close to being aligned. This local-to-global strategy is described in Algorithm 4.

Some examples of the two strategies are shown in Fig. 5.

V. Experiment

In this section, we use different real-world scans to test and compare the proposed method with several state-of-the-art registration approaches, including JRMPC [30], EMPMR [32], the fuzzy cluster-based pair-wise method [15] (called "FuzzyPair" for simplicity), TEASER++ [6], and JPR [33]. For JRMPC, the number of components of its GMM will be specified in each test, and the other parameters use the default setting, where the iteration number is 100. For EMPMR, the parameters are chosen as its default setting, where the iteration number is also 100. For FuzzyPair, it is applied to sequentially align every two consecutive point sets and combine the results. To give fair comparisons, FuzzyPair uses local optimizations only in its coarse-to-fine registration process, where the fuzzy cluster number in the coarse registration is 100, and no point down-sampling is applied in the fine registration. For TEASER++, which is a true correspondence-based pair-wise registration method, it is also applied in the sequential align-and-combine fashion, where the true correspondences are built using the FPFH features [37]. For JPR, there are a number of parameters needing to be manually selected, such as different weights and iteration numbers. We use its example setting for the weights and let the iteration number be 100.

For simplicity, we name the coarse-to-fine strategy in Algorithm 3 as "FuzzyGReg+QA(C2F)" and the local-to-global strategy in Algorithm 4 as "FuzzyGReg+QA(L2G)". The parameters of the proposed method are chosen as follows: N_C is specified in each test; in FuzzyGReg+QA(C2F), $N_{Itr} = 100$ and $N_{Itr} = 80$ for the coarse and fine registrations, respectively; in FuzzyGReg+QA(L2G), $N_{Itr} = 30$ for the global refinement; in FuzzyQA, $\mu_{th} = 1/\sqrt{N_C}$ and $\overline{d}_{ub} = 0.015$; in the local re-alignment of FuzzyQA, FuzzyGReg is applied in the coarse-to-fine fashion to register two consecutive point sets, where $N_C = 60$ and $N_C = 200$ are used in the coarse and fine registrations, respectively.

Given a transformation $\boldsymbol{\theta}_i = (\boldsymbol{R}_i, \boldsymbol{t}_i)$, the rotation and translation errors are defined as $\epsilon_{\boldsymbol{R}_i} = \|\boldsymbol{R}_i - \boldsymbol{R}_i^{(gt)}\|_f$ and $\epsilon_{\boldsymbol{t}_i} = \|\boldsymbol{t}_i - \boldsymbol{t}_i^{(gt)}\|$, respectively, where $\boldsymbol{\theta}_i^{(gt)} = (\boldsymbol{R}_i^{(gt)}, \boldsymbol{t}_i^{(gt)})$ denotes the ground truth of $\boldsymbol{\theta}_i$. For a group of point sets $\mathcal{P}$, we express all the transformations in terms of the first point set in $\mathcal{P}$, and thus only the results of the second to the last point sets are presented. We also use average errors to evaluate the performance as follows: $\bar{\epsilon}_{\boldsymbol{R}} = \sum_{i=2}^{N_P} \epsilon_{\boldsymbol{R}_i}/(N_P - 1)$ and $\bar{\epsilon}_{\boldsymbol{t}} = \sum_{i=2}^{N_P} \epsilon_{\boldsymbol{t}_i}/(N_P - 1)$. All the tests are conducted on a computer with the Intel Core i9-10885H CPU.

A. Tests of sensitivities to overlaps and initial poses

This subsection presents Tests 1 and 2 that evaluate the sensitivities of FuzzyGReg with respect to different overlapping ratios and initial poses of point sets. In these tests, FuzzyGReg is performed in the coarse-to-fine fashion described in Algorithm 3. Then, FuzzyQA only calculates $\overline{d}_{i(i+1)}$ of the results of FuzzyGReg and does not re-align point sets.

Test 1 is based on point set pairs. We select three pairs of point sets from the ETH data set [48] to evaluate and compare FuzzyGReg with JRMPC [30], EMPMR [32], and FuzzyPair [15]. The selected point set pairs are shown in Fig. 6 with their point numbers and overlapping ratios stated in the caption. For each pair, we manually rotate one point set about z-axis by different angles to deviate it from its alignment with the other set such that a series of misaligned initial poses can be created. We then apply all the registration methods to the point sets with these poses. In each test case, JRMPC uses 300 components; FuzzyPair uses no trimming for registering Pair 1 and 50% trimming for registering Pairs 2 and 3; and FuzzyGReg is performed twice with different N_C as follows: FuzzyGReg-I uses 60 and 200 fuzzy clusters in the coarse and fine registrations, respectively; and FuzzyGReg-II uses 80 and 300 fuzzy clusters in the coarse and fine registrations, respectively. The results of Test 1 are shown in Fig. 6.

For Pair 1 with the largest overlapping ratio, JRMPC works when the initial angles are within $[0, 15°]$, which is a relatively narrow convergence basin; EMPMR takes the lowest time costs but also gives a narrow convergence basin where the initial angles are within $[-5°, 5°]$; FuzzyPair without trimming has the broadest convergence range owing to the sufficient overlaps; FuzzyGReg provides the second broadest convergence range and needs lower costs than JRMPC and FuzzyPair do. By using more fuzzy clusters, FuzzyGReg-II gives more accurate results while taking more time than FuzzyGReg-I does. Besides, within the convergence basin, FuzzyGReg gives satisfactory transformations, where $\overline{d}_{12} < \overline{d}_{ub}$. For Pairs 2 and 3 with small overlaps (around or below 50%), JRMPC cannot align them at all; EMPMR only works when the initial angles are around 0; FuzzyPair needs trimming in these cases and gives relatively small convergence ranges. FuzzyGReg has the broadest convergence basin, and its time costs are still lower than that of JRMPC and FuzzyPair. Also, both accuracy and costs of FuzzyGReg-II are higher than that of FuzzyGReg-I. Note that FuzzyGReg does not give optimal results for Pairs 2 and 3 due to the small overlapping ratios, and $\overline{d}_{12}$ indicates this fact since its values are greater than $\overline{d}_{ub}$.

Test 2 is based on two groups of point sets still selected from the ETH data set [48], where each group contains three scans. We apply JRMPC [30], EMPMR [32], TEASER++ [6], and FuzzyGReg to align the two groups. JRMPC uses 300 components; and FuzzyGReg uses 80 and 300 fuzzy clusters in the coarse and fine registrations, respectively. The results are shown in Fig. 7. For the first group with relatively high overlapping ratios (about 90%) and relatively large initial angle differences (about 50°), JRMPC can align the point sets while EMPMR cannot; TEASER++ requires a low time cost but leaves a small misalignment; FuzzyGReg achieves the highest accuracy and takes less time than JRMPC does. For the second group with lower overlapping ratios (62% and 85%) and smaller initial angle differences (about 30°), JRMPC and EMPMR cannot work; TEASER++ leaves a relatively large translation error due to the repeated structures in the scans (as shown in the enlarged part in Fig. 7); FuzzyGReg aligns the point sets, and the values of $\overline{d}_{i(i+1)}$ are below $\overline{d}_{ub}$.

From Tests 1 and 2, unlike the pair-wise methods FuzzyPair and TEASER++, FuzzyGReg neither needs to trim point sets nor needs to know true correspondences. Besides, FuzzyGReg can be fast than FuzzyPair and can provide more accurate alignments than TEASER++ does. When compared with the

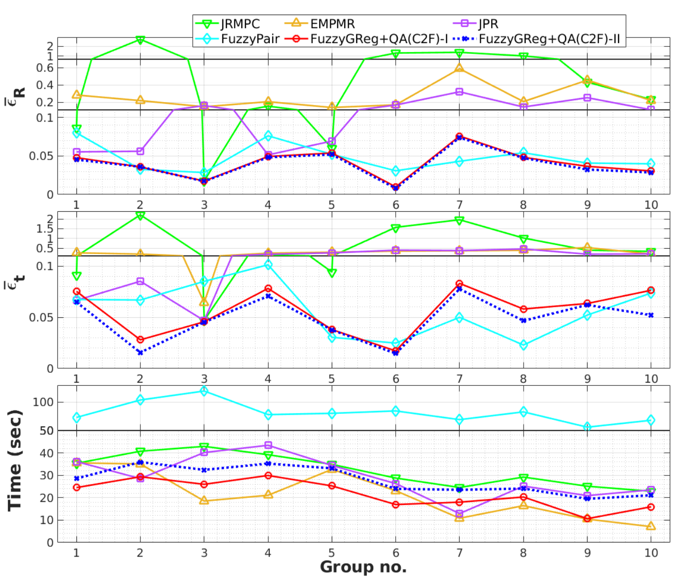


Fig. 8. Results of Test 3, where JRMPC [30], EMPMR [32], JPR [33], FuzzyPair [15], and FuzzyGReg+QA(C2F) are applied to register 10 groups of point sets selected from [49] and [50]. Each of Groups 1-5 contains 20 point sets, and each of Groups 6-10 contains 15 point sets. The number of points in each of these point sets ranges from 1200 to 3300.

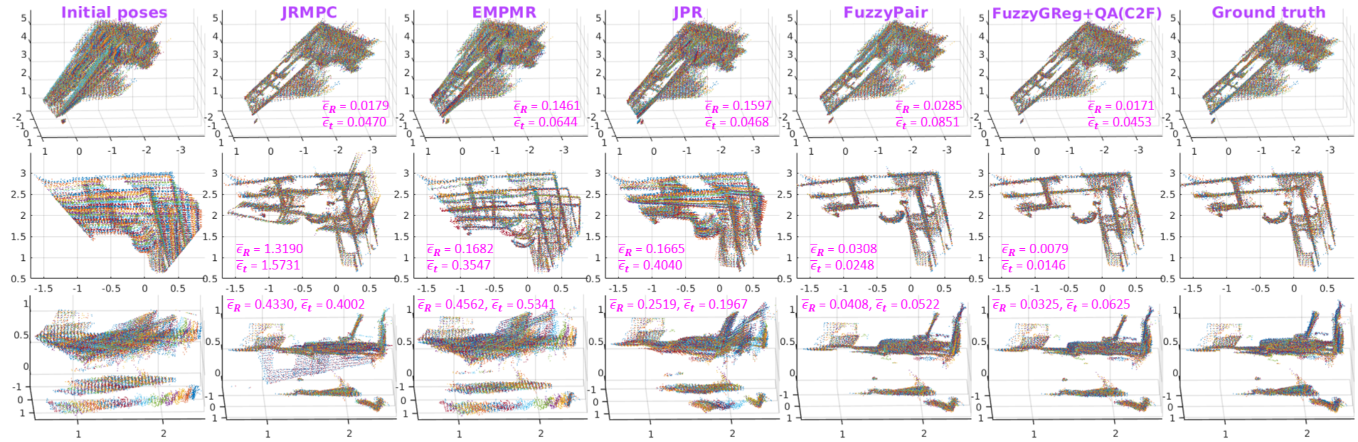


Fig. 9. Results of Groups 3, 6, and 9 in Test 3. For Groups 3 and 6 shown in the first and second rows, respectively, FuzzyGReg gives correct alignments in the group-wise registration; while for Group 9 shown in the third row, FuzzyGReg does not well align some point sets in the group-wise registration, and FuzzyQA detects and re-aligns the point sets to provide the correct final result. The re-alignments of FuzzyQA for Group 9 are shown in Fig. 10.

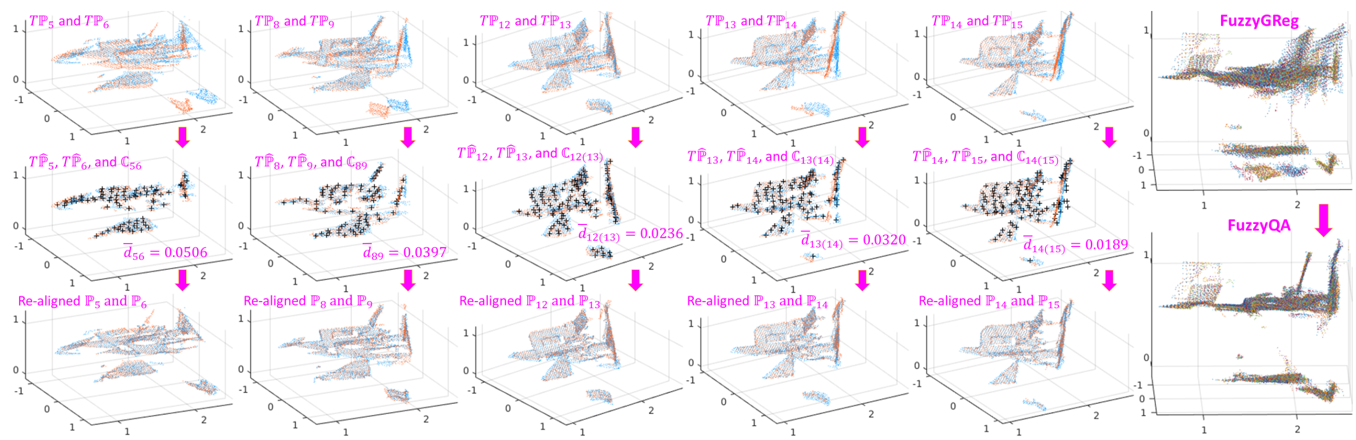


Fig. 10. Re-alignments of FuzzyQA for Group 9 in Test 3. From left to right, in each of columns 1-5, the top sub-figure shows two neighboring point sets $T\mathbb{P}_i$ (blue) and $T\mathbb{P}_{i+1}$ (orange) that are not well aligned by FuzzyGReg in the group-wise registration; the middle sub-figure shows the selected main overlapping parts, including $T\hat{\mathbb{P}}_i$ (blue), $T\hat{\mathbb{P}}_{i+1}$ (orange), and $\mathbb{C}_{i(i+1)}$ (black "+"), where $\bar{d}_{i(i+1)} > \bar{d}_{ub}$; the bottom sub-figure shows the re-aligned $\mathbb{P}_i$ and $\mathbb{P}_{i+1}$. In the rightmost column, the top sub-figure shows all the point sets transformed by FuzzyGReg in the group-wise registration, which are not well aligned; and the bottom sub-figure shows the final poses of the point sets after performing FuzzyQA, which are well aligned.

group-wise methods JRMPC and EMPMR, FuzzyGReg can be more robust against small overlapping ratios and poor initial poses. Besides, the time costs of FuzzyGReg are lower than that of JRMPC. The computational complexity of FuzzyGReg can be lower than that of EMPMR since FuzzyGReg based on fuzzy clusters needs to calculate fewer parameters compared to EMPMR based on Gaussian distributions. The reason that EMPMR takes low time costs is that EMPMR traps into local optima quickly. When the number of fuzzy clusters increases, FuzzyGReg can achieve higher accuracy but also requires more costs in time and computation. When the initial pose differences are large, or the overlapping ratios of point sets are too small (e.g., below 50%), FuzzyGReg usually cannot provide the optimal alignment or even fails, while FuzzyQA based on (13) can detect the misaligned results.

B. Tests on relatively small groups of point sets

This subsection presents Tests 3 and 4 that are based on relatively small groups of point sets ($N_P \leq 20$) to evaluate the performance of FuzzyGReg+QA(C2F) in Algorithm 3.

Test 3 is based on 10 groups of point sets selected from the SUN3D data set [49] and the TUM data set [50], where each of the first 5 groups contains 20 point sets, and each of the rest 5 groups contains 15 point sets. We apply JRMPC [30], EMPMR [32], JPR [33], FuzzyPair [15], and FuzzyGReg+QA(C2F) to register the 10 groups. For each group, JRMPC uses 500 components; FuzzyPair does not use trimming; and FuzzyGReg+QA(C2F) is performed twice with different N_C as follows: FuzzyGReg+QA(C2F)-I uses 100 and 400 fuzzy clusters in the coarse and fine registrations, respectively; FuzzyGReg+QA(C2F)-II uses 100 and 500 fuzzy clusters in the coarse and fine registrations, respectively. The results of this test are presented in Fig. 8. JRMPC achieves small errors for Group 3 while leaving relatively large errors or erroneous results for the other 9 groups; EMPMR does not work in all the 10 groups; JPR provides relatively small errors in rotation for Groups 1, 2, 4, and 5 and relatively small errors in translation for Groups 1, 2, and 3; FuzzyPair, without trimming, works for all the 10 groups and achieves the smallest errors in rotation or translation for some groups, but it requires relatively high time costs for all the groups; FuzzyGReg+QA(C2F) provides satisfactory alignments for all the 10 groups and attains the highest accuracy in both rotation and translation with the lowest time cost for a large portion of these groups. In addition, with more fuzzy clusters, FuzzyGReg+QA(C2F)-II achieves smaller errors while taking higher time costs than FuzzyGReg+QA(C2F)-I does.

We select Groups 3, 6, and 9 of Test 3 to present more details. The results of all the methods for the three groups are shown in Fig. 9. For Group 3, JRMPC, FuzzyPair, and the proposed method (FuzzyGReg+QA(C2F)-II) align the 20 point sets while EMPMR and JPR leave relatively large errors, and the proposed method achieves the highest accuracy. For Group 6, FuzzyPair and the proposed method align the 15 point sets while the others do not, and the proposed method still gives the smallest errors. For Group 9, same as that for Group 6, FuzzyPair and the proposed method work while the others fail, and the errors of FuzzyPair and the proposed method are comparable. Note that for Groups 3 and 6, FuzzyGReg provides correct results in the group-wise registration, and FuzzyQA does not apply any re-alignment; while for Group 9, FuzzyGReg does not well align some point sets in the group-wise registration, and FuzzyQA detects and corrects the local misalignment, as shown in Fig. 10.

Test 4 is based on a group containing 10 point sets selected from the TUM data set [50]. We apply FuzzyPair without trimming and FuzzyGReg+QA(C2F) to align these point sets, where FuzzyGReg uses 100 and 300 fuzzy clusters in the coarse and fine registrations, respectively. The results are shown in Fig. 11. In this group, some point sets have relatively small overlaps. FuzzyPair leaves a misaligned part due to the adverse effects of the non-overlapping points, while FuzzyGReg+QA(C2F) achieves satisfactory results and takes much lower time costs than FuzzyPair does.

From Tests 3 and 4, compared with the group-wise methods JRMPC, EMPMR, and JPR, the proposed FuzzyGReg+QA(C2F) provides more satisfactory performances. In some cases, like Groups 3 and 6 in Test 3, FuzzyGReg can well align all the point sets, and FuzzyQA can know the fact and does not apply re-alignments. In some cases, like Group 9 in Test 3 and the group in Test 4, although FuzzyGReg leaves some misaligned parts, FuzzyQA can detect and re-align these misaligned parts. Note that JPR provides the covariance of its output for uncertainty quantification. However, the covariance is used under the condition that the point sets are aligned or are close to being aligned. Besides, it is used to assist user intervention in 3D reconstruction [33]. Thus, JPR cannot perceive misalignment based on the covariance alone. In contrast, FuzzyQA gives accurate registration quality assessments and improves registration accuracy in the absence of ground truth and user intervention, offering a higher level of automation. In addition, unlike JPR, the proposed method has much fewer parameters needing to be manually determined and thus is more convenient to be implemented. When compared with FuzzyPair that may not work well for partial overlapping point sets without trimming, the proposed method is more robust against the non-overlapping parts and can provide more accurate results with lower time costs.

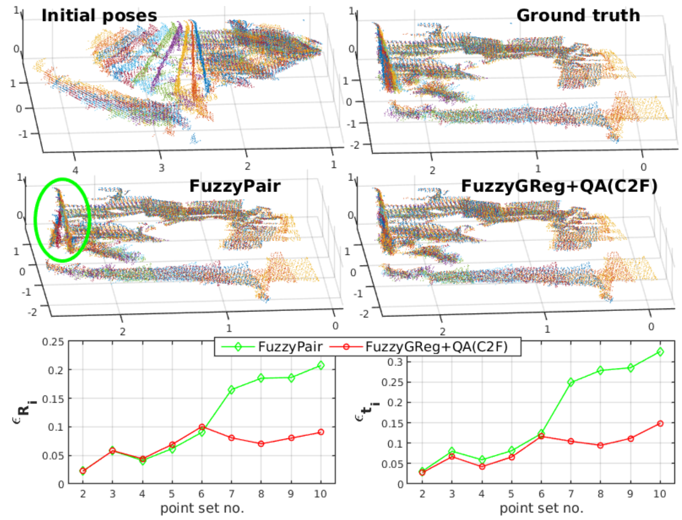


Fig. 11. Results of Test 4, where FuzzyPair [15] and FuzzyGReg+QA(C2F) are applied to register a group of point sets with $N_P = 10$ selected from [50]. In this group, each N_{P_i} is around 2700. The results of FuzzyPair (no trimming) are $\bar{\epsilon}_{\boldsymbol{R}} = 0.1132$, $\bar{\epsilon}_{\boldsymbol{t}} = 0.1680$, and time $= 45.53s$. The green ellipse marks the misaligned part. The results of FuzzyGReg+QA(C2F) are $\bar{\epsilon}_{\boldsymbol{R}} = 0.0684$, $\bar{\epsilon}_{\boldsymbol{t}} = 0.0856$, and time $= 15.72s$, where FuzzyGReg takes 8.93 s and FuzzyQA takes 6.79 s.

C. *Tests on relatively large groups of point sets*

This subsection presents Tests 5 and 6 that are based on relatively large groups of point sets ($N_P \geq 200$) to evaluate the performance of FuzzyGReg+QA(L2G) in Algorithm 4.

Test 5 is based on a group of point sets with $N_P = 200$ selected from the KITTI data set [51]. We apply JRMPC [30], FuzzyPair [15], and FuzzyGReg+QA(L2G) to register these point sets. The group is split into 50 sequential subgroups, each of which contains 5 point sets (every two consecutive subgroups share a common point set, and the last subgroup has 4 point sets only). In the local phase, FuzzyGReg+QA(C2F) is applied to register each subgroup, where FuzzyGReg uses 70 and 150 fuzzy clusters in the coarse and fine registrations, respectively; and in the global phase, FuzzyGReg uses 1500 fuzzy clusters in the refinement. For fair comparisons, JRMPC is also performed in the local-to-global fashion that first registers each of the subgroups, then links the subgroups, and finally globally refines all the poses. In the local phase, JRMPC uses 150 components to register each subgroup; and in the global phase, JRMPC uses 1500 components in the refinement. FuzzyPair is performed with the trimming ratios fixed as 0.2. The results are presented in Fig. 12 and Table I.

In Test 5, JRMPC and FuzzyPair do not align the point sets. JRMPC gives incorrect results in registering some subgroups and thus leads to an erroneous final result. FuzzyPair using fixed trimming ratio cannot well handle the point sets with different overlapping ratios and suffers from cumulative error. In addition, the quality assessment of FuzzyPair is used to select coarse alignments rather than precise alignments and may not give valid results when the trimming ratio is smaller than the true value [15]. In contrast, FuzzyGReg+QA(L2G) gives satisfactory results with the lowest values in both errors and time costs. Note that when these $N_P = 200$ point sets are

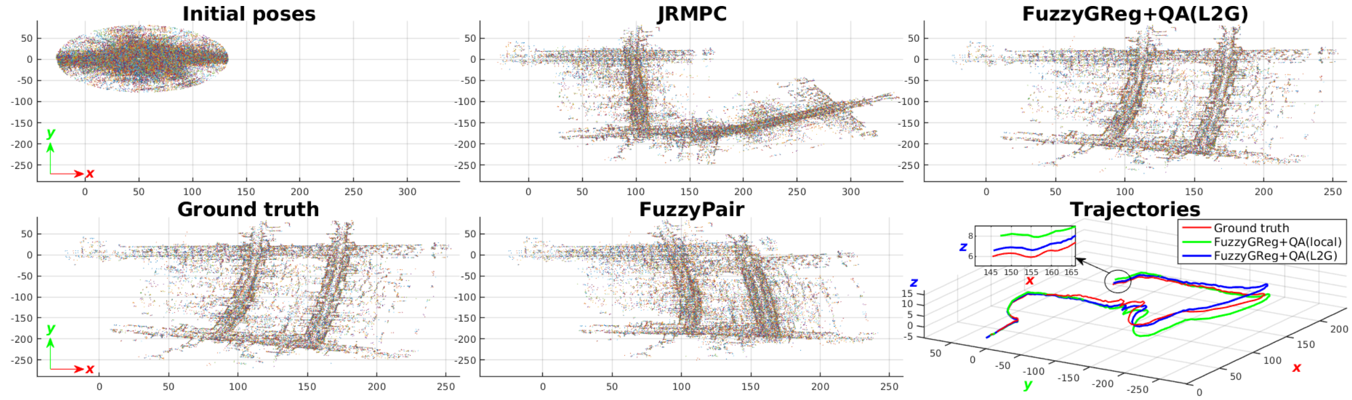


Fig. 12. Results of Test 5, where JRMPC [30], FuzzyPair [15], and FuzzyGReg+QA(L2G) are applied to register a group of point sets with $N_P = 200$ selected from [51]. In this group, each N_{P_i} is about 2100.

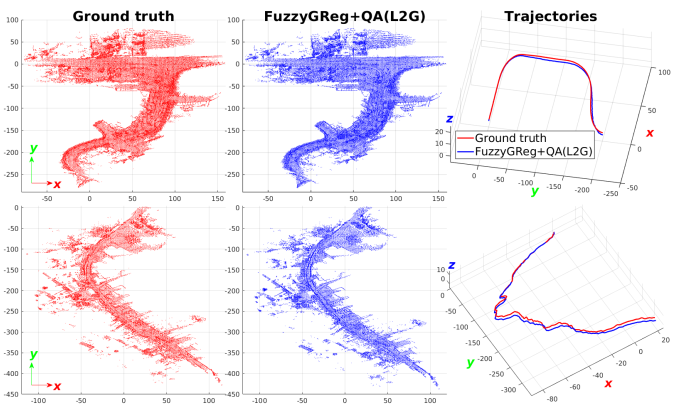


Fig. 13. Results of Test 6, where FuzzyGReg+QA(L2G) is applied to register two groups of point sets selected from [51]. The results of Group 1 are shown in the first row, where $N_P = 300$ and N_{P_i} ranges from 900 to 2200. The results of Group 2 are shown in the second row, where $N_P = 400$ and N_{P_i} ranges from 800 to 3100.

TABLE I
ERRORS AND TIME COSTS OF THE METHODS IN TEST 5

	$\bar{\epsilon}_R$	$\bar{\epsilon}_t$	time
JRMPC	2.2207	163.3611	230.08 s
FuzzyPair	0.3435	19.9442	396.72 s
FuzzyGReg+QA(L2G)	**0.0436**	**2.3170**	**165.20** s

aligned, they form a closed loop. From the sub-figure showing trajectories in Fig. 12, after locally registering and linking subgroups (FuzzyGReg+QA(local)), the end of the trajectory (green) is about 2 meters away from the ground truth (red). Then, after performing the global refinement to obtain the final result of FuzzyGReg+QA(L2G), the end of the trajectory (blue) is much closer to the ground truth (red). The reason for this fact is that the local registration of the proposed method has roughly drawn the closed loop, where the point sets at the loop closure place, although not well aligned, are moved to be close to one another. Then, based on these rough results, the global refinement can move the point sets further toward the exact alignment.

Test 6 is based on two groups of point sets still selected from [51]. We apply FuzzyGReg+QA(L2G) with different N_C to register the two groups. Group 1 with $N_P = 300$ is split into 60 subgroups, each of which contains 6 point sets (the last subgroup contains 5 point sets only); in the local phase, FuzzyGReg+QA(C2F) uses 70 and 170 fuzzy clusters in the coarse and fine registrations, respectively; and in the global refinement, FuzzyGReg uses 1000 fuzzy clusters. Group 2 with $N_P = 400$ is split into 57 subgroups, each of which contains 8 point sets; in the local phase, FuzzyGReg+QA(C2F) uses 70 and 200 fuzzy clusters in the coarse and fine registrations, respectively; and in the global refinement, FuzzyGReg uses 1200 fuzzy clusters. The results are given in Fig. 13 and Table II. Note that the initial poses of these two groups are similar to that of the group in Fig. 12. Without the prior knowledge of the sensor positions collecting the scans in the relatively large scenes, the proposed method can move the scans to be close to their actual poses, and the obtained trajectories are close to the ground truth trajectories.

TABLE II
ERRORS AND TIME COSTS OF THE PROPOSED METHOD IN TEST 6

FuzzyGReg	Group 1			Group 2		
+QA(L2G)	$\bar{\epsilon}_R$	$\bar{\epsilon}_t$	time	$\bar{\epsilon}_R$	$\bar{\epsilon}_t$	time
Local phase	0.0591	3.11	91.0 s	0.0721	6.06	179.6 s
Global phase	0.0396	1.83	58.9 s	0.0633	4.25	96.7 s

From Tests 5 and 6, FuzzyGReg+QA(L2G) works well for the groups with relatively large numbers of point sets and provides the alignments close to the ground truths. The refinement in the global phase can improve the alignment accuracy but also needs relatively high time costs, because it takes $O(\sum_{i=1}^{N_P} N_C \cdot N_{P_i})$ operations with a relatively large N_C. To reduce the costs, users can down-sample the point sets appropriately in the global refinement phase.

VI. CONCLUSION

This study develops FuzzyGReg and FuzzyQA for group-wise point set registration. Given a group of point sets, FuzzyGReg creates a model of fuzzy clusters and equally considers all the points as the elements of the fuzzy clusters. Then, it applies a fuzzy clustering algorithm to identify the parameters of the fuzzy clusters while jointly transforming all the point sets to achieve an alignment. Based on the identified fuzzy clusters, FuzzyQA calculates the spatial properties of the overlapping parts of every two consecutive point sets and checks whether the point sets are aligned based on the similarity degrees of the spatial properties. When a local misalignment is detected, a local re-alignment is applied to improve registration accuracy. For higher efficiency and practicality, we develop a coarse-to-fine strategy and a local-to-global strategy to perform FuzzyGReg and FuzzyQA. Compared with the existing pair-wise registration methods, the proposed method treats all the point sets on an equal footing to avoid biased results. Compared with the state-of-the-art group-wise registration methods, the proposed method is simpler in calculation and is more convenient in implementation. In addition, the proposed method has a reliable quality assessment method to detect and correct misalignment in the absence of ground truth and user intervention, providing a higher level of automation. The experiments utilize different real-world scans to test and compare the proposed method with state-of-the-art registration techniques. The experimental results demonstrate the effectiveness of our method.

References

[1] P. J. Besl and N. D. McKay, "A method for rgistration of 3-d shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, pp. 239–256, 1992.

[2] D. Chetverikov, D. Stepanov, and P. Krsek, "Robust euclidean alignment of 3d point sets: The trimmed iterative closest point algorithm," *Image and Vision Computing*, vol. 23, no. 3, pp. 299–309, 2005.

[3] S. Bouaziz, A. Tagliasacchi, and M. Pauly, "Sparse iterative closest point," in *Computer Graphics Forum*, Wiley Online Library, vol. 32, 2013, pp. 113–123.

[4] J. Zhang, Y. Yao, and B. Deng, "Fast and robust iterative closest point," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[5] C. Olsson, F. Kahl, and M. Oskarsson, "Branch-and-bound methods for euclidean registration problems," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 5, pp. 783–794, 2009.

[6] H. Yang, J. Shi, and L. Carlone, "Teaser: Fast and certifiable point cloud registration," *IEEE Transactions on Robotics*, vol. 37, no. 2, pp. 314–333, 2021.

[7] Z. J. Yew and G. H. Lee, "3dfeat-net: Weakly supervised local 3d features for point cloud registration," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 607–623.

[8] Z. Gojcic, C. Zhou, J. D. Wegner, and A. Wieser, "The perfect match: 3d point cloud matching with smoothed densities," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5545–5554.

[9] Y. Aoki, H. Goforth, R. A. Srivatsan, and S. Lucey, "Pointnetlk: Robust & efficient point cloud registration using pointnet," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7163–7172.

[10] G. D. Pais, S. Ramalingam, V. M. Govindu, J. C. Nascimento, R. Chellappa, and P. Miraldo, "3dregnet: A deep neural network for 3d point registration," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 7193–7203.

[11] M. El Banani, L. Gao, and J. Johnson, "Unsupervisedr&r: Unsupervised point cloud registration via differentiable rendering," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 7129–7139.

[12] M. Magnusson, A. Lilienthal, and T. Duckett, "Scan registration for autonomous mining vehicles using 3d-ndt," *Journal of Field Robotics*, vol. 24, no. 10, pp. 803–827, 2007.

[13] B. Jian and B. C. Vemuri, "Robust point set registration using gaussian mixture models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1633–1645, 2011.

[14] T. Stoyanov, M. Magnusson, H. Andreasson, and A. J. Lilienthal, "Fast and accurate scan registration through minimization of the distance between compact 3d ndt representations," *The International Journal of Robotics Research*, vol. 31, no. 12, pp. 1377–1393, 2012.

[15] Q. Liao, D. Sun, and H. Andreasson, "Point set registration for 3d range scans using fuzzy cluster-based metric and efficient global optimization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 9, pp. 3229–3246, 2020.

[16] ——, "Fuzzypsreg: Strategies of fuzzy cluster-based point set registration," *IEEE Transactions on Robotics*, 2021.

[17] D. Adolfsson, M. Magnusson, Q. Liao, A. J. Lilienthal, and H. Andreasson, "Coral–are the point clouds correctly aligned?" In *2021 European Conference on Mobile Robots (ECMR)*, IEEE, 2021, pp. 1–7.

[18] Y. Chen and G. Medioni, "Object modelling by registration of multiple range images," *Image and Vision Computing*, vol. 10, no. 3, pp. 145–155, 1992.

[19] G. Blais and M. D. Levine, "Registering multiview range data to create 3d computer objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 8, pp. 820–824, 1995.

[20] T. Masuda, "Generation of geometric model by registration and integration of multiple range images," in *Proceedings Third International Conference on 3-D Digital Imaging and Modeling*, IEEE, 2001, pp. 254–261.

[21] R. Bergevin, M. Soucy, H. Gagnon, and D. Laurendeau, "Towards a general multi-view registration technique," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 5, pp. 540–547, 1996.

[22] D. F. Huber and M. Hebert, "Fully automatic registration of multiple 3d data sets," *Image and Vision Computing*, vol. 21, no. 7, pp. 637–650, 2003.

[23] S. Choi, Q.-Y. Zhou, and V. Koltun, "Robust reconstruction of indoor scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 5556–5565.

[24] G. C. Sharp, S. W. Lee, and D. K. Wehe, "Multiview registration of 3d scenes by minimizing error between coordinate frames," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 8, pp. 1037–1050, 2004.

[25] S.-W. Shih, Y.-T. Chuang, and T.-Y. Yu, "An efficient and accurate method for the relaxation of multiview registration error," *IEEE Transactions on Image Processing*, vol. 17, no. 6, pp. 968–981, 2008.

[26] V. M. Govindu and A. Pooja, "On averaging multiview relations for 3d scan registration," *IEEE Transactions on Image Processing*, vol. 23, no. 3, pp. 1289–1302, 2013.

[27] Z. Li, J. Zhu, K. Lan, C. Li, and C. Fang, "Improved techniques for multi-view registration with motion averaging," in *2014 2nd International Conference on 3D Vision*, IEEE, vol. 1, 2014, pp. 713–719.

[28] R. Guo, J. Zhu, Y. Li, D. Chen, Z. Li, and Y. Zhang, "Weighted motion averaging for the registration of multi-view range scans," *Multimedia Tools and Applications*, vol. 77, no. 9, pp. 10 651–10 668, 2018.

[29] X. Mateo, X. Orriols, and X. Binefa, "Bayesian perspective for the registration of multiple 3d views," *Computer Vision and Image Understanding*, vol. 118, pp. 84–96, 2014.

[30] G. D. Evangelidis and R. Horaud, "Joint alignment of multiple point sets with batch and incremental expectation-

maximization," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 6, pp. 1397–1410, 2018.

[31] N. Ravikumar, A. Gooya, S. Çimen, A. F. Frangi, and Z. A. Taylor, "Group-wise similarity registration of point sets using student's t-mixture model for statistical shape models," *Medical image analysis*, vol. 44, pp. 156–176, 2018.

[32] J. Zhu, R. Guo, Z. Li, J. Zhang, and S. Pang, "Registration of multi-view point sets under the perspective of expectation-maximization," *IEEE Transactions on Image Processing*, vol. 29, pp. 9176–9189, 2020.

[33] X. Huang, Z. Liang, and Q. Huang, "Uncertainty quantification for multi-scan registration," *ACM Transactions on Graphics (TOG)*, vol. 39, no. 4, pp. 130–1, 2020.

[34] S. Krishnan, P. Y. Lee, J. B. Moore, and S. Venkatasubramanian, "Optimisation-on-a-manifold for global registration of multiple 3d point sets," *International Journal of Intelligent Systems Technologies and Applications*, vol. 3, no. 3-4, pp. 319–340, 2007.

[35] L. Ding and C. Feng, "Deepmapping: Unsupervised map estimation from multiple point clouds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8650–8659.

[36] Z. Gojcic, C. Zhou, J. D. Wegner, L. J. Guibas, and T. Birdal, "Learning multiview 3d point cloud registration," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1759–1769.

[37] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (fpfh) for 3d registration," in *2009 IEEE International Conference on Robotics and Automation*, IEEE, 2009, pp. 3212–3217.

[38] C. Zach, A. Penate-Sanchez, and M.-T. Pham, "A dynamic programming approach for fast and robust object pose recognition from range images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 196–203.

[39] Q.-F. Liao, D. Sun, W.-J. Cai, S.-Y. Li, and Y.-Y. Wang, "Type-1 and type-2 effective takagi-sugeno fuzzy models for decentralized control of multi-input-multi-output processes," *Journal of Process Control*, vol. 52, pp. 26–44, 2017.

[40] Q.-F. Liao and D. Sun, "Interaction measures for control configuration selection based on interval type-2 takagi–sugeno fuzzy model," *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 5, pp. 2510–2523, 2018.

[41] Q. Liao and D. Sun, "Sparse and decoupling control strategies based on takagi–sugeno fuzzy models," *IEEE Transactions on Cybernetics*, vol. 51, no. 2, pp. 947–960, 2019.

[42] D. Sun, Q. Liao, T. Stoyanov, A. Kiselev, and A. Loutfi, "Bilateral telerobotic system using type-2 fuzzy neural network based moving horizon estimation force observer for enhancement of environmental force compliance and human perception," *Automatica*, vol. 106, pp. 358–373, 2019.

[43] D. Sun, Q. Liao, and A. Loutfi, "Type-2 fuzzy model-based movement primitives for imitation learning," *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2462–2480, 2022.

[44] J. C. Bezdek, R. Ehrlich, and W. Full, "Fcm: The fuzzy c-means clustering algorithm," *Computers & geosciences*, vol. 10, no. 2-3, pp. 191–203, 1984.

[45] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 04, pp. 376–380, 1991.

[46] D. E. Gustafson and W. C. Kessel, "Fuzzy clustering with a fuzzy covariance matrix," in *IEEE Conference on Decision and Control including the 17th Symposium on Adaptive Processes*, IEEE, 1979, pp. 761–766.

[47] M. Herdin, N. Czink, H. Ozcelik, and E. Bonek, "Correlation matrix distance, a meaningful measure for evaluation of non-stationary mimo channels," in *2005 IEEE 61st Vehicular Technology Conference*, IEEE, vol. 1, 2005, pp. 136–140.

[48] F. Pomerleau, M. Liu, F. Colas, and R. Siegwart, "Challenging data sets for point cloud registration algorithms," *The International Journal of Robotics Research*, vol. 31, no. 14, pp. 1705–1711, 2012.

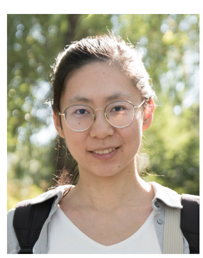
[49] J. Xiao, A. Owens, and A. Torralba, "Sun3d: A database of big spaces reconstructed using sfm and object labels," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 1625–1632.

[50] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of rgb-d slam systems," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2012, pp. 573–580.

[51] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2012, pp. 3354–3361.

Qianfang Liao received a B.Eng. degree in automation from Wuhan University, China, in 2006, an M.Eng. degree in automation from Shanghai Jiao Tong University, China, in 2009, and a Ph.D. degree in electrical engineering from Nanyang Technological University, Singapore, in 2015. She is currently a tenured researcher with Örebro University, Sweden. Her research interests include fuzzy-based learning, perception, robotics, and control theory.

Da Sun received B.Eng. and Ph.D. degrees in mechatronics from the University of Wollongong, Australia, in 2012 and 2016, respectively. He is currently a tenured researcher with the Center for Applied Autonomous Sensor Systems (AASS), Örebro University, Sweden. His research interests include robot learning and control, and human-robot interaction.



Shiyu Zhang received a B.Eng degree in automation from Hefei University of Technology in 2014 and Ph.D. degree in navigation, guidance and control from Beihang University in 2019. She was a Postdoctoral Researcher with the Centre for Applied Autonomous Sensor Systems, Örebro University from 2019 to 2022. She is currently an Associate Researcher with the School of Modern Post (School of Automation), Beijing University of Posts and Telecommunications. Her research interests include multi-robot coordination and human-robot interaction.

Amy Loutfi received the Ph.D. degree in computer science from Örebro University, Örebro, Sweden in 2006. She is currently a professor in information technology at Örebro University with the Center for Applied Autonomous Sensor Systems (AASS), Sweden. Her general interests are in the integration of artificial intelligence with autonomous systems, and over the years, she has looked into applications where robots closely interact with humans in both industrial and domestic environments. Her thesis focused on the integration of artificial olfaction into robotic and intelligent systems to enable good human-robot interactions. Since then, she has examined various cases of human-robot interactions, including applications of telepresence.

Henrik Andreasson received a Master degree in mechatronics from the Royal Institute of Technology, Sweden, in 2001 and a Ph.D. degree in computer science from Örebro University, Sweden, in 2008. He is currently an associate professor with the Center for Applied Autonomous Sensor Systems (AASS), Örebro University, Sweden. His research interests include mobile robotics, computer vision, and machine learning.