

# FuzzyPSReg: Strategies of Fuzzy Cluster-Based Point Set Registration

Qianfang Liao<sup>ID</sup>, Da Sun<sup>ID</sup>, and Henrik Andreasson<sup>ID</sup>

**Abstract**—This article studies the fuzzy cluster-based point set registration (FuzzyPSReg). First, we propose a new metric based on Gustafson-Kessel (GK) fuzzy clustering to measure the alignment of two point clouds. Unlike the metric based on fuzzy c-means (FCM) clustering in our previous work, the GK-based metric includes orientation properties of the point clouds, thereby providing more information for registration. We then develop the registration quality assessment of the GK-based metric, which is more sensitive to small misalignments than that of the FCM-based metric. Next, by effectively combining the two metrics, we design two FuzzyPSReg strategies with global optimization. 1) *FuzzyPSReg-SS*, which extends our previous work and aligns two similar-sized point clouds with greatly improved efficiency. 2) *FuzzyPSReg-O2S*, which aligns two point clouds with a relatively large difference in size and can be used to estimate the pose of an object in a scene. In the experiment, we use different point clouds to test and compare the proposed method with state-of-the-art registration approaches. The results demonstrate the advantages and effectiveness of our method.

**Index Terms**—3-D point clouds, fuzzy clusters, object pose estimation, point set registration, registration quality assessment.

## I. INTRODUCTION

POINT set registration finds the optimal spatial transformation to align two sets of points (point clouds). It is an essential technique in many areas, such as computer vision, pattern recognition, image processing, and robotics. This study focuses on 3-D rigid point set registration, where the transformation consists of 3-D rotation and translation.

To develop an effective registration method, the first task is to select a metric that can correctly measure the alignment of point clouds. Various metrics have been proposed, such as the iterative closest point (ICP) [1] and its variants [2]–[5], point correspondence-based metrics [6]–[8], probability distribution-based metrics [9]–[13], neural network-based metrics [14]–[17], and fuzzy cluster-based metrics [18]. Optimizing the metric to

derive the optimal transformation is another major task. Different optimization algorithms are available, including local optimizations like singular value decomposition [1], gradient-based algorithms [9], [12], expectation-maximization (EM) [10], [13], etc., and global optimizations like stochastic algorithms [19], [20], deterministic branch-and-bound (BnB) algorithms [6], [18], [21], correspondence-based algorithms [6]–[8], etc. It is nontrivial to achieve high robustness and efficiency for registration. To this end, many techniques have been studied, such as data structures (e.g., kd-tree) to speed up the nearest neighbor search [21], [22], outlier trimming to suppress noises [2], and automatic registration quality assessments to reduce time and computational costs [18].

Most existing registration methods aim to align two point clouds with sufficient overlaps, meaning that the point clouds must have similar sizes. These methods may not be suitable for applications that need to align two point clouds with a relatively large size difference, like aligning an object model to a relatively large scene scan to estimate the pose of the object in the scene. Existing studies of object pose estimation are usually devoted to providing a good precondition/initialization, like a coarse alignment or sufficient correspondences between point clouds, for registration methods to complete the task. For example, with RGB-D data, some methods [23]–[25] employ deep learning to encode features for object detection, and some methods [26]–[29] utilize specified features (e.g., ellipses) to recognize objects. However, in many scenarios (e.g., underwater exploration), color cues are unreliable or even unavailable. Then, these methods generally cannot work with depth data alone since depth data are texture-less and miss some crucial information for their feature extractions in contrast to color images. Some techniques can be found that extract features from depth-only data, which also include deep learning (e.g., 3-D ShapeNet [30], PointNet [31], and those used by neural network-based registration methods [14]–[17]) and handcrafted feature descriptors (e.g., Hough transform [32], spin images [33], and fast point feature histograms (FPFH) [34]). Deep learning is popular and has demonstrated its viability in many perception applications. However, it requires a long training phase, and its performance is restricted by the training data. Handcrafted features are generally designed to work with relatively clean range data like laser scans, and may not work very well with the scans collected by commodity depth sensors [35]. In addition, the 3-D information provided by handcrafted features may be relatively uninformative (many flat surfaces or similar curves). Consequently, one may not find a sufficient number of reliable

RGB-D 借助于RGB进行分割的缺点

Manuscript received July 31, 2021; accepted October 14, 2021. This work was supported in part by VINNOVA under the Project TAMMP 2019-05878 and Project LUNAR 2020-04483, and in part by FORMAS under the project RoboClean 2019-02264. This paper was recommended for publication by Associate Editor J. Bohg and Editor F. Chaumette upon evaluation of the reviewers' comments. (Qianfang Liao and Da Sun are co-first authors.) (Corresponding author: Da Sun.)

The authors are with the Centre for Applied Autonomous Sensor Systems, Örebro University, 701 82 Örebro, Sweden (e-mail: qianfang.liao@oru.se; da.sun@oru.se; henrik.andreasson@oru.se).

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/TRO.2021.3123898>.

Digital Object Identifier 10.1109/TRO.2021.3123898

features [35]. The above facts motivate us to develop effective strategies to align point clouds of both similar and different sizes, which can achieve high accuracy at low or acceptable costs, work with depth data collected by commodity sensors, and neither require long-time training nor rely on handcrafted feature extractions.

This article studies the fuzzy cluster-based point set registration (FuzzyPSReg). In our recent work [18], a registration metric based on fuzzy c-means (FCM) clustering [36] is designed and is minimized by a global optimization algorithm to align point clouds. Besides, the registration quality assessment of the FCM-based metric is proposed to boost computational efficiency. In this study, first, we design a new metric based on a different fuzzy clustering: Gustafson–Kessel (GK) method [37]. Unlike the FCM-based metric, the GK-based metric includes fuzzy covariance matrices to provide orientation information for registration. Afterward, we develop the registration quality assessment of the GK-based metric, which is more sensitive to small misalignments than that of the FCM-based metric in [18]. Next, by effectively combining the two fuzzy cluster-based metrics, we develop two FuzzyPSReg strategies. The first strategy aligns two point clouds with similar sizes and is called *FuzzyPSReg-SS* (registering similar-sized scans). FuzzyPSReg-SS integrates the GK-based metric into the global registration method of [18] and greatly improves computational efficiency. The second strategy aligns point clouds with a relatively large size difference and can be used for object pose estimation. We call it *FuzzyPSReg-O2S* (registering an object to a scene). FuzzyPSReg-O2S is designed as a process of elimination that (iteratively) selects possible solutions using multistart optimization algorithms and then checks whether any solution gives the correct alignment using the registration quality assessments. It also employs FuzzyPSReg-SS for orientation correction or further refinement. In addition, it includes human-computer collaboration and provides shared autonomy to complete tasks without putting a heavy burden on users. The FuzzyPSReg strategies work well with the data collected by commodity depth sensors and do not require long-time training or handcrafted feature extractions. In the experiments, different point clouds are employed to test and compare the proposed method with state-of-the-art registration techniques. The results demonstrate the accuracy, robustness, and effectiveness of our method.

The rest of this article is organized as follows. Section II reviews the related work. Section III presents the fuzzy cluster-based metrics. Section IV describes FuzzyPSReg-SS and FuzzyPSReg-O2S. In Section V, experimental results are given. Finally, Section VI concludes this article.

## II. RELATED WORK

In this section, we review some relevant point set registration methods. ICP [1] is the best-known method in this area. Starting from an initial estimate, it performs in a direct and intuitive way that iteratively minimizes the distances between the closest points of two point clouds until convergence. Currently, various modifications and extensions of ICP can be found, including

revised metrics for higher accuracy and robustness, like the generalized expression [3], the  $l_p$ -norm distance loss [4], the robust version via Welsch's function [5], etc.; different approaches for better convergence, like the EM algorithm [38], the Levenberg–Marquardt algorithm [22], the Anderson acceleration [5], [39], etc.; and outlier handling schemes for noise reduction, like the trimmed ICP [2]. The limitation of these ICPs lies in the local optimizations that will fail without a good initial estimate. To overcome this limitation, global optimizations have been developed without relying on initial estimates, such as genetic algorithms [19] and BnB-based algorithms [21].

Some studies have investigated correspondence-based registration. They use feature descriptors [33], [34] to build correspondences between two point clouds, and then optimize the correspondence-based metric to obtain the correct transformation. For example, in [6], a BnB-based algorithm is designed to globally minimizes the distances between correspondences to align point clouds; in [7], a correspondence-based metric is developed based on the Geman–McClure estimator, and the graduated nonconvexity is employed to achieve the global optimization for registration; and in [8], an approach called TEASER++ is proposed that uses a decoupling scheme to globally minimizes the truncated least squares distances between correspondences to realize point cloud matching.

Unlike the above methods taking point distances between two point clouds as the metric, some other methods represent one or each of the point clouds by a certain model and then optimize the model-based metric to achieve the alignment. Probability distribution-based models, like the Gaussian mixture model (GMM) and the normal distributions transform (NDT), are widely used to represent point clouds in these methods, such as the coherent point drift [10], GMM-Reg [11], JRMP (joint registration of multiple point clouds) [13], the point-to-distribution NDT [9], and the distribution-to-distribution NDT [12]. As deep learning became popular, neural networks have emerged to describe point clouds for registration. For example, PointNetLK [14] uses PointNet to encode point clouds and applies a modified L&K algorithm for registration. 3DSmoothNet [15] matches point clouds with a siamese deep learning architecture and fully convolutional layers. 3DRegNet [16] utilizes a neural network to identify inliers from a set of correspondences between two scans and then calculates the transformation for scan alignment. The feature-metric registration (FMR) [17] trains a neural network to extract features from point clouds, and then minimizes the feature-metric projection error to realize registration without correspondences. As described before, the main problem of the deep learning-based methods is that they need long-time training and cannot guarantee the generalization of the trained neural network to match unseen point clouds. In our recent work [18], different from the above methods, fuzzy clusters are used to describe each point cloud and then form the FCM-based metric. In addition, an efficient optimization algorithm is designed to globally minimize the FCM-based metric for registration. Unlike ICPs, the FCM-based metric is more robust to noise, has a broader convergence basin, and does not need to search for the nearest neighbors. Unlike the correspondence-based approaches, the FCM-based metric is a correspondence-free

method and saves the correspondence construction. Unlike the probability-based metrics, the FCM-based metric does not involve expensive calculations (exponentials, etc.), and thus, its implementation is simpler and more cost-effective. Unlike the deep learning methods, the FCM-based metric neither needs any training phase nor suffers from the generalization issue.

Most existing registration methods, including ours in [18], are designed to register similar-sized point clouds and are generally not suitable for aligning point clouds with a relatively large difference in size. For this problem, similar to the object pose estimation methods mentioned before, some learned or hand-crafted feature-based registration approaches may be applied when the features provide good preconditions. Nevertheless, as described before, the learned or handcrafted features may not provide a good precondition for every point cloud pair. Therefore, alternative strategies are desired.

### III. FUZZY CLUSTER-BASED REGISTRATION METRICS

Fuzzy clusters are effective and robust to model various systems and data, including 3-D scans for registration [18]. Given two 3-D point clouds/range scans: one is the fixed set denoted as  $\mathbb{P}_F = \{\mathbf{p}_{Fi}, i = 1, \dots, N_{P_F}\}$  and the other is the moving set denoted as  $\mathbb{P}_M = \{\mathbf{p}_{Mj}, j = 1, \dots, N_{P_M}\}$ , where  $\mathbf{p}_{Fi}, \mathbf{p}_{Mj} \in \mathbb{R}^3$  are point coordinates, and  $N_{P_F}$  and  $N_{P_M}$  are point numbers, a rigid point set registration must find the 3-D transformation  $\lambda = (\mathbf{r}, \mathbf{t}) \in \mathbb{R}^6$ , where  $\mathbf{r}, \mathbf{t} \in \mathbb{R}^3$  are rotation and translation parameters, respectively, such that the transformed moving set, denoted by  $T(\lambda, \mathbb{P}_M) = \{T(\lambda, \mathbf{p}_{Mj}), j = 1, \dots, N_{P_M}\}$ , can be correctly aligned with  $\mathbb{P}_F$ . The transformed point  $T(\lambda, \mathbf{p}_{Mj})$  is derived by

$$T(\lambda, \mathbf{p}_{Mj}) = \mathbf{R}(\mathbf{r}) \cdot \mathbf{p}_{Mj} + \mathbf{t} \quad (1)$$

where  $\mathbf{R}(\mathbf{r}) \in SO(3)$  is the rotation matrix. In this study,  $\mathbf{r}$  is expressed by the axis-angle representation, where the axis is  $\mathbf{r}/\|\mathbf{r}\|$  and the angle is  $\|\mathbf{r}\|$ . In this section, first, we briefly introduce the FCM-based registration metric developed in our previous work [18]. Afterward, we propose a new registration metric based on GK clustering [37]. Finally, we present a theoretical discussion on the two fuzzy cluster-based metrics.

#### A. FCM-Based Registration Metric

Given a 3-D point set  $\mathbb{P} = \{\mathbf{p}_j \in \mathbb{R}^3, j = 1, \dots, N_P\}$ , we suppose that it is described by  $N_C$  fuzzy clusters, whose centers/prototypes/centroids are denoted as  $\mathbb{C} = \{\mathbf{c}_i \in \mathbb{R}^3, i = 1, \dots, N_C\}$ . FCM clustering [36] locates  $\mathbb{C}$  by iteratively minimizing the following objective function:

$$\min_{\mathbb{C}} \left\{ J_{fcm}(\mathbb{P}, \mathbb{C}) = \sum_{j=1}^{N_P} \sum_{i=1}^{N_C} \mu_{\mathbf{c}_i}(\mathbf{p}_j)^m \cdot \|\mathbf{p}_j - \mathbf{c}_i\|^2 \right\} \quad (2)$$

where  $m > 1$  controls the fuzziness of fuzzy clusters: A greater  $m$  leads to a fuzzier partition, and a general choice is  $m = 2$ ;  $\mu_{\mathbf{c}_i}(\mathbf{p}_j)$  denotes the fuzzy membership of  $\mathbf{p}_j$  in the fuzzy cluster centered at  $\mathbf{c}_i$ , which satisfies  $0 \leq \mu_{\mathbf{c}_i}(\mathbf{p}_j) \leq 1$  and

$\sum_{i=1}^{N_C} \mu_{\mathbf{c}_i}(\mathbf{p}_j) = 1$ , and is calculated by

$$\mu_{\mathbf{c}_i}(\mathbf{p}_j) = \begin{cases} 1, & \text{if } \|\mathbf{p}_j - \mathbf{c}_i\| = 0 \\ 0, & \text{if } \exists_{k \neq i} \|\mathbf{p}_j - \mathbf{c}_k\| = 0 \\ \frac{1}{\sum_{k=1}^{N_C} \left( \frac{\|\mathbf{p}_j - \mathbf{c}_k\|^2}{\|\mathbf{p}_j - \mathbf{c}_i\|^2} \right)^{\frac{1}{m-1}}}, & \text{else.} \end{cases} \quad (3)$$

In our previous work [18], based on the fuzzy cluster centers of  $\mathbb{P}_F$  and  $\mathbb{P}_M$  derived by FCM clustering, denoted as  $\mathbb{C}_F = \{\mathbf{c}_{Fi}, i = 1, \dots, N_{C_F}\}$  and  $\mathbb{C}_M = \{\mathbf{c}_{Mj}, j = 1, \dots, N_{C_M}\}$ , respectively, a metric is designed to measure the alignment of  $\mathbb{P}_F$  and  $\mathbb{P}_M$  using the following logic: when  $\mathbb{P}_F$  and  $T(\lambda, \mathbb{P}_M)$  are aligned, in their overlapping region, the points or fuzzy cluster centers of  $T(\lambda, \mathbb{P}_M)$  describe the same spatial properties/surfaces as the points of  $\mathbb{P}_F$  do. As a result, the points or fuzzy cluster centers of  $T(\lambda, \mathbb{P}_M)$  in the overlapping region can be regarded as the elements of  $\mathbb{P}_F$ 's fuzzy clusters, and thus, their associated FCM clustering objective function with respect to  $\mathbb{C}_F$  in the form of (2) is minimized. The FCM-based registration metric is given in (4), minimizing which can derive the transformation  $\lambda$  to align  $\mathbb{P}_F$  and  $\mathbb{P}_M$  [18]

$$\min_{\lambda} \left\{ J_{fcm}(\lambda, \mathbb{C}'_M, \mathbb{C}_F) = \sum_{j=1}^{N_{C'_M}} J_{fcm}(\lambda, \mathbf{c}'_{Mj}, \mathbb{C}_F) = \sum_{j=1}^{N_{C'_M}} \sum_{i=1}^{N_{C_F}} \mu_{\mathbf{c}_{Fi}}(T(\lambda, \mathbf{c}'_{Mj}))^m \cdot \|T(\lambda, \mathbf{c}'_{Mj}) - \mathbf{c}_{Fi}\|^2 \right\} \quad (4)$$

where  $\mu_{\mathbf{c}_{Fi}}(T(\lambda, \mathbf{c}'_{Mj}))$  denotes the fuzzy membership grade of  $T(\lambda, \mathbf{c}'_{Mj})$  in the fuzzy cluster centered at  $\mathbf{c}_{Fi}$ ;  $\mathbb{C}'_M = \{\mathbf{c}'_{Mj}, j = 1, \dots, N_{C'_M}\} \subseteq \mathbb{C}_M$  is selected in the following way [18]: when  $\mathbb{P}_M$  is fully overlapped by  $\mathbb{P}_F$ ,  $\mathbb{C}'_M = \mathbb{C}_M$ . When  $\mathbb{P}_M$  is partially overlapped by  $\mathbb{P}_F$ , we estimate the nonoverlapping portion of  $\mathbb{P}_M$  to have a trimming ratio  $\xi \in (0, 1)$  and  $N_{C'_M} = N_{C_M} \cdot (1 - \xi)$ ; then, at each calculation of a new transformation,  $\mathbb{C}'_M$  is composed of  $N_{C'_M}$  elements in  $\mathbb{C}_M$  having the smallest per-element distance losses  $J_{fcm}(\lambda, \mathbf{c}_{Mj}, \mathbb{C}_F)$ , where  $\lambda$  is the current transformation.

Another contribution of [18] is a registration quality assessment. The two point clouds can be considered as aligned or coarsely aligned if the following condition is satisfied:

$$\rho_{fcm}(\lambda) = \frac{\bar{J}_{fcm}(\lambda, \mathbb{C}'_M, \mathbb{C}_F)}{\bar{J}_{fcm}(\mathbb{P}_F, \mathbb{C}_F)} < 1 \quad (5)$$

where  $\bar{J}_{fcm}(\lambda, \mathbb{C}'_M, \mathbb{C}_F)$  and  $\bar{J}_{fcm}(\mathbb{P}_F, \mathbb{C}_F)$  are average per-element distance losses calculated by

$$\begin{cases} \bar{J}_{fcm}(\lambda, \mathbb{C}'_M, \mathbb{C}_F) = J_{fcm}(\lambda, \mathbb{C}'_M, \mathbb{C}_F) / N_{C'_M} \\ \bar{J}_{fcm}(\mathbb{P}_F, \mathbb{C}_F) = J_{fcm}(\mathbb{P}_F, \mathbb{C}_F) / N_{P_F} \end{cases}$$

which indicate the disposition and dispersion of the elements around the fuzzy cluster centers  $\mathbb{C}_F$  [18]. The ratio  $\rho_{fcm}(\lambda) < 1$  means that the disposition and dispersion of  $T(\lambda, \mathbb{C}'_M)$  around  $\mathbb{C}_F$  are similar to that of the points of  $\mathbb{P}_F$  around  $\mathbb{C}_F$  [18].

Consequently, the two point clouds,  $\mathbb{P}_F$  and  $T(\lambda, \mathbb{P}_M)$ , can be regarded as aligned or at least coarsely aligned.

*Remark 1:* When the number of fuzzy clusters  $N_{C_F}$  is close to or equal to the number of points  $N_{P_F}$  the value of  $\bar{J}_{fcm}(\mathbb{P}_F, \mathbb{C}_F)$  will be close to or equal to 0. In this case,  $\bar{J}_{fcm}(\mathbb{P}_F, \mathbb{C}_F)$  cannot correctly describe the disposition and dispersion of the points of  $\mathbb{P}_F$  around  $\mathbb{C}_F$  since there are not sufficient points in each fuzzy cluster. Hence, the quality assessment (5) should be applied when  $N_{C_F}$  is much smaller than  $N_{P_F}$  [18], such that  $\bar{J}_{fcm}(\mathbb{P}_F, \mathbb{C}_F)$  is valid.

The FCM-based metric utilizes a weighted sum of Euclidean distances between the two point sets' fuzzy cluster centers to measure the alignment, where one-to-all correspondences are employed. Next, we design a new fuzzy cluster-based metric using Mahalanobis distances for the measurement.

计算马氏距离

### B. GK-Based Registration Metric

In this section, we present a new registration metric based on GK fuzzy clustering [37]. Unlike FCM clustering, GK clustering uses adaptive distance norm. For a 3-D point set  $\mathbb{P}$ , GK clustering locates  $N_C$  fuzzy cluster centers  $\mathbb{C}$  by iteratively minimizing the following objective function:

$$\min_{\mathbb{C}} \left\{ J_{gk}(\mathbb{P}, \mathbb{C}) = \sum_{j=1}^{N_P} \sum_{i=1}^{N_C} \mu_{c_i}(\mathbf{p}_j)^m \cdot \|\mathbf{p}_j - \mathbf{c}_i\|_{\mathbf{A}_i}^2 \right\} \quad (6)$$

where  $m > 1$ ;  $\|\mathbf{p}_j - \mathbf{c}_i\|_{\mathbf{A}_i}^2 = (\mathbf{p}_j - \mathbf{c}_i)^T \mathbf{A}_i (\mathbf{p}_j - \mathbf{c}_i)$  is the Mahalanobis distance between  $\mathbf{p}_j$  and  $\mathbf{c}_i$ ;  $\mathbf{A}_i$ , as an element of  $\mathbf{A} = \{\mathbf{A}_i \in \mathbb{R}^{3 \times 3}, i = 1, \dots, N_C\}$ , denotes the norm-inducing matrix of the  $i$ th fuzzy cluster and is calculated by

$$\mathbf{A}_i = \det(\mathbf{K}_i)^{1/3} \cdot \mathbf{K}_i^{-1} \quad (7)$$

where  $\mathbf{K}_i \in \mathbb{R}^{3 \times 3}$  is the fuzzy covariance matrix of the  $i$ th fuzzy cluster, and is calculated by

$$\mathbf{K}_i = \frac{\sum_{j=1}^{N_P} \mu_{c_i}(\mathbf{p}_j)^m \cdot (\mathbf{p}_j - \mathbf{c}_i) \cdot (\mathbf{p}_j - \mathbf{c}_i)^T}{\sum_{j=1}^{N_P} \mu_{c_i}(\mathbf{p}_j)^m}. \quad (8)$$

The objective function of GK clustering in (6) shares the same form of the objective function of FCM clustering in (2), except that Mahalanobis instead of Euclidean distances are used. Also, the fuzzy membership grades  $\mu_{c_i}(\mathbf{p}_j)$  in (6) and (8) are calculated by (3) with each Euclidean distance  $\|\cdot\|$  replaced by the corresponding Mahalanobis distance  $\|\cdot\|_{\mathbf{A}_i}$ . Using the similar logic of the FCM-based metric (4), we develop the following GK-based registration metric:

$$\begin{aligned} \min_{\lambda} \left\{ J_{gk}(\lambda, \mathbb{P}'_M, \mathbb{C}_F) = \sum_{j=1}^{N_{P'_M}} J_{gk}(\lambda, \mathbf{p}'_{Mj}, \mathbb{C}_F) = \right. \\ \left. \sum_{j=1}^{N_{P'_M}} \sum_{i=1}^{N_{C_F}} \mu_{c_{Fi}}(T(\lambda, \mathbf{p}'_{Mj}))^m \cdot \|T(\lambda, \mathbf{p}'_{Mj}) - \mathbf{c}_{Fi}\|_{\mathbf{A}_{Fi}}^2 \right\} \end{aligned} \quad (9)$$

where  $\mathbf{A}_{Fi} \in \mathbf{A}_F$  is the norm-inducing matrix of the  $i$ th fuzzy cluster of  $\mathbb{P}_F$ ;  $\mu_{c_{Fi}}(T(\lambda, \mathbf{p}'_{Mj}))$  is calculated in the way of

(3) with each  $\|\cdot\|$  replaced by the corresponding  $\|\cdot\|_{\mathbf{A}_{Fi}}$ ;  $\mathbb{P}'_M = \{\mathbf{p}'_{Mj}, j = 1, \dots, N_{P'_M}\} \subseteq \mathbb{P}_M$  is determined in a way similar to that of selecting  $\mathbb{C}'_M$  described in Section III-A: When  $\mathbb{P}_M$  is fully overlapped by  $\mathbb{P}_F$ ,  $\mathbb{P}'_M = \mathbb{P}_M$ . When  $\mathbb{P}_M$  is partially overlapped by  $\mathbb{P}_F$ , we estimate a trimming ratio  $\xi \in (0, 1)$  for  $\mathbb{P}_M$  to have  $N_{P'_M} = N_{P_M} \cdot (1 - \xi)$ ; then, at each calculation of a new transformation,  $\mathbb{P}'_M$  is composed of  $N_{P'_M}$  points in  $\mathbb{P}_M$  having the smallest per-element distance losses  $J_{gk}(\lambda, \mathbf{p}'_{Mj}, \mathbb{C}_F)$ , where  $\lambda$  is the current transformation.

The GK-based registration metric in (9) is a weighted sum of adaptive distances between  $\mathbb{P}_F$ 's fuzzy cluster centers and  $\mathbb{P}_M$ 's points. Similar to the FCM-based metric (4), it utilizes one-to-all correspondences with fuzzy membership grades as the weights. Also, we can have a registration quality assessment for the GK-based metric (9). Similar to (5), a ratio, denoted by  $\rho_{gk}(\lambda)$ , is calculated as follows:

$$\rho_{gk}(\lambda) = \frac{\bar{J}_{gk}(\lambda, \mathbb{P}'_M, \mathbb{C}_F)}{\bar{J}_{gk}(\mathbb{P}_F, \mathbb{C}_F)} = \frac{J_{gk}(\lambda, \mathbb{P}'_M, \mathbb{C}_F)/N_{P'_M}}{J_{gk}(\mathbb{P}_F, \mathbb{C}_F)/N_{P_F}}. \quad (10)$$

An important difference between  $\rho_{gk}(\lambda)$  of (10) and  $\rho_{fcm}(\lambda)$  of (5) is that in (5), the numerator  $\bar{J}_{fcm}(\lambda, \mathbb{C}'_M, \mathbb{C}_F)$  is an average center-to-center distance loss, and the denominator  $\bar{J}_{fcm}(\mathbb{P}_F, \mathbb{C}_F)$  is an average point-to-center distance loss; while in (10), both of the numerator and denominator are average point-to-center distance losses. Due to this fact, the registration quality assessment using  $\rho_{gk}(\lambda)$  is different from (5):  $\mathbb{P}_F$  and  $T(\lambda, \mathbb{P}_M)$ , can be regarded as aligned if

$$\rho_{gk}(\lambda) \leq 1 + \Delta J \quad (11)$$

where  $\Delta J > 0$  is a small number. The reason is as follows: when  $\mathbb{P}_F$  and  $\mathbb{P}_M$  are aligned by  $\lambda$ , their points in the overlapping region describe the same surfaces/spatial properties, and thus, the points of  $T(\lambda, \mathbb{P}'_M)$  and  $\mathbb{P}_F$  are very similar in terms of the disposition and dispersion around  $\mathbb{C}_F$ . Consequently, the ratio of their average per-element distance losses with respect to  $\mathbb{C}_F$ ,  $\rho_{gk}(\lambda)$ , is around 1 to have  $\rho_{gk}(\lambda) \leq 1 + \Delta J$ . On the other hand, fuzzy cluster centers of a point cloud are generally located at the representative positions, and thus, each  $T(\lambda, \mathbf{c}'_{Mj})$  is generally near a  $\mathbf{c}_{Fi}$  when the two point clouds are aligned [18]. As a result, the average per-element distance loss of  $T(\lambda, \mathbb{C}'_M)$  with respect to  $\mathbb{C}_F$  is smaller than that of the points  $\mathbf{p}_{Fi}$  with respect to  $\mathbb{C}_F$  to have  $\rho_{fcm}(\lambda) < 1$ .

*Remark 2:* As stated in [18], the FCM-based registration quality assessment (5) may not work in the following cases: i)  $N_{C'_M}$  is too small; ii) the trimming ratio  $\xi$  is chosen to be too large; and iii) the two scans have a large size difference or a small overlap ratio. Similarly, the GK-based registration quality assessment (11) may be affected in cases ii) and iii). To deal with these issues, some choices and assumptions are made in [18]: For case i), a generally safe choice for 3-D range scan registration is given as  $N_{C'_M} \geq 50$ ; for case ii),  $\xi$  can be manually determined to be an appropriate value; and for case iii), an assumption is made that the overlap of the two scans is at least 50%. However, the assumption for case iii) may not hold in some applications. For example, when aligning an object model with a relatively large scene scan for object pose estimation, the

scene scan can be several times greater than the object model. In this situation, when the two point clouds are aligned, (5) or (11) will be satisfied; but when a transformation  $\lambda$  satisfies (5) or (11), the two point clouds may not be correctly aligned. In Section IV, a strategy is developed to resolve this problem.

### C. Discussion of the Two Fuzzy Cluster-Based Metrics

In this section, we present the mathematical analysis and theoretical comparison of the two fuzzy cluster-based metrics.

1) *Mathematical Analysis of the Metrics*: The following two lemmas from [40] will be used in the analysis.

**Lemma 1:** Given a set of real numbers  $d_i \geq 0$  for  $i = 1, \dots, N - 1$ , and a variable  $d_N \geq 0$ , then,

$$J = \left( \sum_{i=1}^{N-1} d_i^{-v} + d_N^{-v} \right)^{-1/v}, \quad v \geq 1 \quad (12)$$

decreases as  $d_N$  decreases.

**Lemma 2:** If  $d_i \geq 0$  for  $i = 1, \dots, N$ , and  $v \geq 1$ , then

$$\min_{i \in \{1, \dots, N\}} \frac{d_i}{N^{1/v}} \leq \left( \sum_{i=1}^N d_i^{-v} \right)^{-1/v} \leq \min_{i \in \{1, \dots, N\}} d_i. \quad (13)$$

By submitting the fuzzy membership calculation (3) to (4), we can rewrite the FCM-based metric in (4) to be

$$\begin{aligned} J_{fcm}(\lambda, \mathbb{C}'_M, \mathbb{C}_F) &= \sum_{j=1}^{N_{C'_M}} J_{fcm}(\lambda, c'_{Mj}, \mathbb{C}_F) \\ &= \sum_{j=1}^{N_{C'_M}} \left( \sum_{i=1}^{N_{C_F}} (\|T(\lambda, c'_{Mj}) - c_{Fi}\|^2)^{\frac{1}{1-m}} \right)^{1-m}. \end{aligned} \quad (14)$$

By denoting  $D_{ji}^2(\lambda) = \|T(\lambda, c'_{Mj}) - c_{Fi}\|^2$  and  $J_{fcmj} = J_{fcm}(\lambda, c'_{Mj}, \mathbb{C}_F)$ , we have the following from (14):

$$J_{fcmj} = \left( \sum_{i=1}^{N_{C_F}} D_{ji}^2(\lambda)^{\frac{1}{1-m}} \right)^{1-m}. \quad (15)$$

By setting  $v = 1/(m-1)$ , according to Lemmas 1 and 2, we can know that  $J_{fcmj}$  decreases as  $D_{ji}^2(\lambda)$  decreases, and satisfies the following when  $m \in (1, 2]$ :

$$\min_{i \in \{1, \dots, N_{C_F}\}} \frac{D_{ji}^2(\lambda)}{N_{C_F}^{m-1}} \leq J_{fcmj} \leq \min_{i \in \{1, \dots, N_{C_F}\}} D_{ji}^2(\lambda). \quad (16)$$

From (16), if  $m \rightarrow 1^+$ , then  $J_{fcmj} \rightarrow \min_{i \in \{1, \dots, N_{C_F}\}} D_{ji}^2(\lambda)$ .

Thus, the per-element distance loss  $J_{fcmj}$  can be considered as an approximation of the minimum of  $D_{ji}^2(\lambda)$ ,  $i = 1, \dots, N_{C_F}$ . Then, the FCM-based metric,  $J_{fcm}(\lambda, \mathbb{C}'_M, \mathbb{C}_F)$ , can be considered as an alternative measure of  $J(\lambda)$  in (17) to calculate  $\lambda$  for point set registration and satisfies  $J(\lambda)/N_{C_F}^{m-1} \leq J_{fcm}(\lambda, \mathbb{C}'_M, \mathbb{C}_F) \leq J(\lambda)$

$$J(\lambda) = \sum_{j=1}^{N_{C'_M}} \left( \min_{i \in \{1, \dots, N_{C_F}\}} D_{ji}^2(\lambda) \right). \quad (17)$$

Note that when the trimming ratio  $\xi$  is zero that  $\mathbb{C}'_M = \mathbb{C}_M$ ,  $J(\lambda)$  in (17) is the standard ICP metric for  $\mathbb{C}_F$  and  $\mathbb{C}_M$ ; and when  $\xi > 0$  that  $\mathbb{C}'_M$  is a subset of  $\mathbb{C}_M$ ,  $J(\lambda)$  in (17) is the metric of the trimmed ICP [2] for  $\mathbb{C}_F$  and  $\mathbb{C}_M$ .

Similarly, we can rewrite the GK-based metric in (9) to be

$$\begin{aligned} J_{gk}(\lambda, \mathbb{P}'_M, \mathbb{C}_F) &= \sum_{j=1}^{N_{P'_M}} J_{gk}(\lambda, p'_{Mj}, \mathbb{C}_F) \\ &= \sum_{j=1}^{N_{P'_M}} \left( \sum_{i=1}^{N_{C_F}} (\|T(\lambda, p'_{Mj}) - c_{Fi}\|^2_{A_{Fi}})^{\frac{1}{1-m}} \right)^{1-m}. \end{aligned} \quad (18)$$

By denoting  $D_{jAi}^2(\lambda) = \|T(\lambda, p'_{Mj}) - c_{Fi}\|^2_{A_{Fi}}$  and  $J_{gkj} = J_{gk}(\lambda, p'_{Mj}, \mathbb{C}_F)$ , we have the following from (18):

$$J_{gkj} = \left( \sum_{i=1}^{N_{C_F}} D_{jAi}^2(\lambda)^{\frac{1}{1-m}} \right)^{1-m}. \quad (19)$$

According to Lemmas 1 and 2,  $J_{gkj}$  decreases as  $D_{jAi}^2(\lambda)$  decreases, and satisfies the following when  $m \in (1, 2]$ :

$$\min_{i \in \{1, \dots, N_{C_F}\}} \frac{D_{jAi}^2(\lambda)}{N_{C_F}^{m-1}} \leq J_{gkj} \leq \min_{i \in \{1, \dots, N_{C_F}\}} D_{jAi}^2(\lambda). \quad (20)$$

Then, the GK-based metric,  $J_{gk}(\lambda, \mathbb{P}'_M, \mathbb{C}_F)$ , can be considered as an alternative measure of  $J_{\mathbb{A}_F}(\lambda)$  in (21) to calculate  $\lambda$  for point set registration and satisfies  $J_{\mathbb{A}_F}(\lambda)/N_{C_F}^{m-1} \leq J_{gk}(\lambda, \mathbb{P}'_M, \mathbb{C}_F) \leq J_{\mathbb{A}_F}(\lambda)$

$$J_{\mathbb{A}_F}(\lambda) = \sum_{j=1}^{N_{P'_M}} \left( \min_{i \in \{1, \dots, N_{C_F}\}} D_{jAi}^2(\lambda) \right). \quad (21)$$

Different from the point-to-point ICP metric in (17),  $J_{\mathbb{A}_F}(\lambda)$  in (21) can be regarded as the point-to-plane ICP metric [3] including the orientation information of the fixed set.

Compared with the ICP metrics, the FCM- and GK-based metrics have the following advantages. 1) They have registration quality assessments. 2) They are differentiable such that the gradient-based algorithms (e.g., quasi-Newton) can be applied for optimization. 3) They are weighted sums of distances using one-to-all correspondence, which avoid the nearest correspondence search, smooth out the function landscape, and widen the convergence basin. Consequently, the two fuzzy cluster-based metrics are more robust to a poor initialization in local optimization, and can converge to the optima with fewer steps in global optimization.

2) *Comparison of the Metrics*: The main difference between the two fuzzy clustering algorithms is that FCM imposes a spherical shape on the clusters regardless of the actual point distributions, while GK using adaptive distances can detect clusters of different shapes and orientations, and thus, can describe the spatial/geometric properties of a range scan more accurately. In Fig. 1, the Stanford bunny [41] is taken as an example to show the results of the two fuzzy clustering algorithms. With the same fuzzy cluster numbers, GK clustering (Fig. 1(c)) describes the shape of the bunny better than FCM (Fig. 1(b)) does.

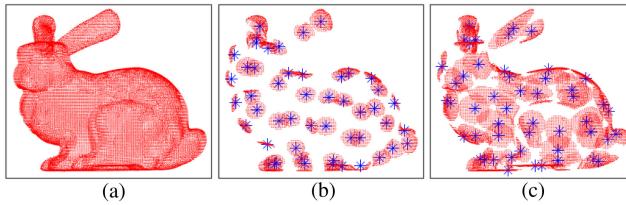


Fig. 1. FCM and GK clustering for the bunny model in (a). Their fuzzy cluster numbers are equal, 50. (b) and (c) show the fuzzy cluster centers (marked by blue “\*”) derived by the two clustering algorithms, together with the points of the model (in red) whose largest fuzzy membership grades are greater than 0.5. These points can reflect the shapes and main ranges of the clusters.

The above difference leads to different usages of the two fuzzy cluster-based metrics. For the FCM-based metric, its registration accuracy is related to the number of fuzzy clusters. When the point clouds are described by dozens of fuzzy clusters, the FCM-based metric usually only gives a coarse alignment (this is a tradeoff issue and will be described in the next section). From (14), the FCM-based metric is composed of Euclidean distances between  $\mathbb{C}_F$  and  $T(\lambda, \mathbb{C}_M)$ . Thus, it can directly take a relatively large number (e.g., thousands) of points of each point cloud as the fuzzy cluster centers to perform a fine registration and achieve a precise alignment [18]. For the GK-based metric, it cannot directly take the points of  $\mathbb{P}_F$  as the fuzzy cluster centers due to the unknown norm-inducing matrix of each point. Therefore, when using the GK-based metric (9) for registration, one needs to apply GK clustering to  $\mathbb{P}_F$  to derive the fuzzy cluster centers and the norm-inducing matrices. However, with the norm-inducing matrices, the GK-based metric can provide an exact alignment when  $\mathbb{P}_F$  is described by only dozens of fuzzy clusters. In addition, the time costs can be much lower than that of the fine registration of the FCM-based metric. Note that the accuracy of the GK-based metric using dozens of fuzzy clusters to describe  $\mathbb{P}_F$  may not be as high as that of the FCM-based metric using thousands of points as the fuzzy cluster centers of  $\mathbb{P}_F$  and  $\mathbb{P}_M$ . Hence, for the applications needing sequential registration, such as 3-D model reconstruction from a batch of partial scans, the GK-based metric may lead to a greater cumulative error than the FCM-based metric does. Nevertheless, this study focuses on pair-wise registration, and the cumulative error handling is not considered here.

Besides, since FCM clustering imposes a spherical shape on the clusters, the average distance losses  $\bar{J}_{fcm}(\mathbb{P}_F, \mathbb{C}_F)$  utilizes equal weights to describe the point dispersion in all directions around  $\mathbb{C}_F$ . By contrast, the average distance losses  $\bar{J}_{gk}(\mathbb{P}_F, \mathbb{C}_F)$  of GK clustering gives different weights to the point dispersion in different directions around  $\mathbb{C}_F$  using norm-inducing matrices (fuzzy covariances). In a surface scan, the point dispersion generally varies greatly in different directions. Taking a plane scan as an example, the points are unlikely to be dispersed in the normal direction of this plane. In this case, FCM still uses spheres to describe the plane, whereas GK can detect the plane shape since the eigenvalue of the fuzzy covariance matrix in the normal direction will be much smaller than that in other directions. As a result, when aligning two plane scans,

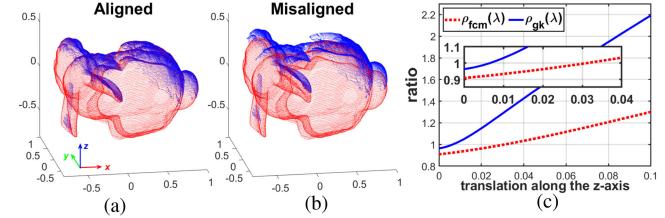


Fig. 2. Registration quality assessments of the two fuzzy cluster-based metrics. (a) Two aligned bunny point clouds, where  $\mathbb{P}_F$  (red) fully overlaps  $\mathbb{P}_M$  (blue). (b) Misalignment is produced by moving  $\mathbb{P}_M$  along the z-axis. (c)  $\rho_{fcm}(\lambda)$  and  $\rho_{gk}(\lambda)$  of the misaligned point clouds in (b) under different translations along the z-axis, where the number of fuzzy clusters for a point cloud is 50. From (c), when the translation is greater than 0.032,  $\rho_{fcm}(\lambda) > 1$ ; and when the translation is greater than 0.006,  $\rho_{gk}(\lambda) > 1$ . Thus,  $\rho_{gk}(\lambda)$  is more sensitive to small misalignments than  $\rho_{fcm}(\lambda)$ .

more sensitive to error  
if given the same number of clusters  
这个地方需要再了解一下

compared with the FCM-based metric, the GK-based metric will produce a much greater distance loss for a small misalignment in the normal directions of the two planes. Therefore, the GK-based quality assessment in (11) can be more sensitive to small errors than the FCM-based quality assessment in (5). An example is shown in Fig. 2.

On the other hand, compared with the FCM-based metric, the GK-based metric has some shortages. First, the time cost of GK fuzzy clustering is generally higher than that of FCM clustering due to the calculation of norm-inducing matrices. Second, the convergence basin of the GK-based metric may not be as broad as that of the FCM-based metric, which makes a good initialization more important to the GK-based metric in local optimization. Nevertheless, the GK-based metric has a broader convergence basin when compared with some other state-of-the-art methods, such as the fast and robust ICP (FRICP) [5], NDT [12], and JRMP [13]. The above will be demonstrated by the tests in Section V-A.

#### IV. FUZZYPSREG STRATEGIES

In this section, the two fuzzy cluster-based metrics in (4) and (9) are effectively combined to develop FuzzyPSReg-SS and FuzzyPSReg-O2S. FuzzyPSReg-SS improves the FCM-based registration method of our previous work [18] to align two similar-sized point clouds, and FuzzyPSReg-O2S aligns two point clouds with a relatively large difference in size.

##### A. FuzzyPSReg-SS

Before presenting FuzzyPSReg-SS, we introduce a tradeoff issue existing in the registration using the FCM-based metric (4). When the fuzzy cluster numbers of  $\mathbb{P}_F$  and  $\mathbb{P}_M$ ,  $N_{CF}$ , and  $N_{CM}$ , are relatively small (e.g., less than 100), the registration takes low time and computational costs, but only coarsely aligns the point clouds; when  $N_{CF}$  and  $N_{CM}$  are relatively large (e.g., thousands), the registration can achieve high accuracy but is computationally expensive. Also, we briefly introduce our previous registration method in [18]. It is a coarse-to-fine method using the FCM-based metric (4) that can globally align two point clouds with sufficient overlaps. The method includes the following three steps:

clusters的数目是否进行一个自动的选择?

Step i) *Fuzzy clustering*: FCM clustering is applied separately to  $\mathbb{P}_F$  and  $\mathbb{P}_M$  to derive  $\mathbb{C}_F$  and  $\mathbb{C}_M$  to compose the FCM-based metric (4), where  $N_{C_F}$  and  $N_{C_M}$  are equal and chosen to be relatively small numbers (e.g., less than 100).

Step ii) *Coarse registration*: A deterministic optimization algorithm, combining BnB-based global search and gradient-based local convergence, is applied to the FCM-based metric to derive the global minimum. It utilizes the registration quality assessment (5) to evaluate the quality of the solution. The minimum gives a coarse alignment of  $\mathbb{P}_F$  and  $\mathbb{P}_M$  since the point clouds are described by small numbers of fuzzy clusters.

Step iii) *Fine registration*: A relatively large number (e.g., more than 1000) of points of each point cloud are directly taken as the fuzzy cluster centers to update the FCM-based metric, where  $N_{C_F}$  and  $N_{C_M}$  do not need to be equal. Then, the gradient-based algorithm is applied to the updated metric to refine the coarse alignment and give a precise result.

The above coarse-to-fine method effectively addresses the tradeoff issue of the FCM-based metric. The coarse registration roughly aligns the two point clouds at a low cost, posing a good initialization for the fine registration. Then, the fine registration using local optimization can achieve the exact alignment at an acceptable cost. In addition, in the coarse registration,  $N_{C_F}$  and  $N_{C_M}$  are usually much smaller than  $N_{P_F}$  and  $N_{P_M}$  of the two point clouds. Consequently, the registration quality assessment (5) is valid according to Remark 1. By using  $\rho_{fcm}(\lambda)$  of (5) to detect whether a good solution is obtained during the global optimization, the computational efficiency can be greatly improved [18].

To further improve the efficiency, we extend our previous work [18] to develop FuzzyPSReg-SS that combines the FCM- and the GK-based metrics to align point clouds regardless of the initialization. As presented in Algorithm 1, line 1 performs the same FCM clustering described in Step i) above; lines 2–4 perform the same coarse registration described in Step ii); while lines 5 and 6 carry out a different fine registration from Step iii): GK clustering is applied to the fixed set to derive new fuzzy clustering results; then, the GK-based metric (9) replaces the FCM-based metric (4) in the local optimization to refine the coarse registration result.

*Discussion of FuzzyPSReg-SS*: As stated in Section III-C, different from the FCM-based metric, the GK-based metric can give an exact alignment when the fixed set  $\mathbb{P}_F$  is described by only dozens of fuzzy clusters. In addition, the time and computational costs of the registration using the GK-based metric are much lower than that of the fine registration of the FCM-based metric. Therefore, the efficiency of FuzzyPSReg-SS is improved when compared with our previous method in [18]. Besides, in line 5 of Algorithm 1, the GK clustering applied to  $\mathbb{P}_F$  takes the fuzzy cluster centers derived by FCM clustering as the initialization. Unlike the general GK clustering procedure that randomly initializes the fuzzy cluster centers, taking FCM clustering results as the initialization allows the objective function of GK clustering (6) to converge to the minimum in fewer iterations. Consequently, the online calculating costs of FuzzyPSReg-SS are further reduced.

fine使用的是GK-distance

---

#### Algorithm 1: FuzzyPSReg-SS.

---

**Input:**  $\mathbb{P}_F$ ,  $\mathbb{P}_M$ ,  $N_{C_F}$ ,  $N_{C_M}$ , and an estimated trimming ratio  $\xi$ , where  $N_{C_F} = N_{C_M}$ .  
**Output:**  $\lambda^* = (\mathbf{r}^*, \mathbf{t}^*)$  moving  $\mathbb{P}_M$  to align with  $\mathbb{P}_F$ .

- 1 Apply FCM clustering separately to  $\mathbb{P}_F$  and  $\mathbb{P}_M$  to derive  $N_{C_F}$  fuzzy cluster centers  $\mathbb{C}_F$  and  $N_{C_M}$  fuzzy cluster centers  $\mathbb{C}_M$ , as well as the average distance loss of  $\mathbb{P}_F$ ,  $\bar{J}_{fcm}(\mathbb{P}_F, \mathbb{C}_F)$ .
- 2 Taking zero as the initial estimate, apply the gradient-based algorithm to the FCM-based metric  $J_{fcm}(\lambda, \mathbb{C}'_M, \mathbb{C}_F)$ , where  $N_{C'_M} = N_{C_M} \cdot (1 - \xi)$ , to obtain a transformation and let it be  $\lambda^*$ .
- 3 **if**  $\rho_{fcm}(\lambda^*) > 1$  **then**
- 4     Apply the deterministic global optimization algorithm developed in [18] to  $J_{fcm}(\lambda, \mathbb{C}'_M, \mathbb{C}_F)$  to obtain a new transformation and let it be  $\lambda^*$ .
- 5 Taking  $\mathbb{C}_F$  as the initialization, apply GK clustering to  $\mathbb{P}_F$  to derive new fuzzy cluster centers  $\mathbb{C}_F$  and the associated  $\mathbf{A}_{Fi}, i = 1, \dots, N_{C_F}$ .
- 6 Taking  $\lambda^*$  as the initial estimate, apply the gradient-based algorithm to the GK-based metric  $J_{gk}(\lambda, \mathbb{P}'_M, \mathbb{C}_F)$ , where  $N_{P'_M} = N_{P_M} \cdot (1 - \xi)$ , to obtain a refined transformation and let it be  $\lambda^*$ .

---

On the other hand, same as [18], FuzzyPSReg-SS needs the two point clouds to have similar sizes and a sufficient overlap. In the next section, we present FuzzyPSReg-O2S to align two point clouds with a relatively large difference in size.

## B. FuzzyPSReg-O2S

Some applications need to align a small point cloud with a relatively large point cloud. For example, to estimate the pose of an object in a scene, the object model, denoted as  $\mathbb{P}_o$ , needs to be aligned with the scene scan, denoted as  $\mathbb{P}_s$ . In these applications, FuzzyPSReg-SS, as well as many existing registration methods, may not directly work for the alignment. Besides, the scene is usually cluttered with other items, producing outliers in the scene scan. In addition,  $\mathbb{P}_s$  is usually taken by a depth sensor and only shows the view from the sensor's angle, meaning that  $\mathbb{P}_s$  is a partial scan of the scene that only contains a part of the object's surface. On the other hand,  $\mathbb{P}_o$  can be a full model of the object derived from the CAD model or reconstructed from a batch of partial scans of the object. Therefore, besides the size difference, it is common that  $\mathbb{P}_s$  and  $\mathbb{P}_o$  partially overlap each other.

In this section, we introduce FuzzyPSReg-O2S that combines the two fuzzy cluster-based metrics to align the full model of an object,  $\mathbb{P}_o$ , with a partial scan of the scene containing this object,  $\mathbb{P}_s$ . We first define the following registration quality assessment extended from (11).

Extended registration quality assessment: Define two numbers  $\Delta\bar{J}_1$  and  $\Delta\bar{J}_2$ , where  $0 < \Delta\bar{J}_1 < \Delta\bar{J}_2$ , and define a variable  $q_{gk}(\lambda) = \{-1, 0, 1\}$  to indicate different assessment

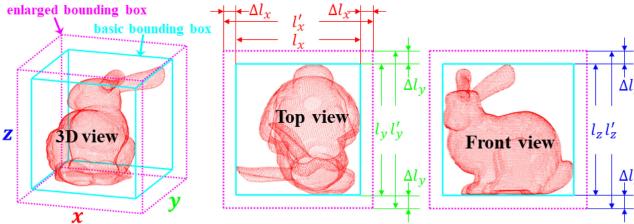


Fig. 3. Two bounding boxes of a point cloud. The side length of the basic bounding box (in cyan) along the  $i$ -axis ( $i = x, y, z$ ), denoted as  $l_i$ , is defined as the distance between the maximum and minimum values of this point cloud on the  $i$ -axis. The enlarged bounding box (in magenta) is a concentric cuboid of the basic bounding box, and its side length along the  $i$ -axis, denoted as  $l'_i$ , is calculated by  $l'_i = l_i + 2 \times \Delta l_i$ , where  $\Delta l_i = 0.1 \times l_i$ .

results. Given  $\lambda$  for  $\mathbb{P}_F$  and  $\mathbb{P}_M$ ,  $q_{gk}(\lambda)$  is derived by

$$q_{gk}(\lambda) = \begin{cases} -1, & \text{if } \rho_{gk}(\lambda) > 1 + \Delta \bar{J}_2 \\ 0, & \text{if } 1 + \Delta \bar{J}_1 < \rho_{gk}(\lambda) \leq 1 + \Delta \bar{J}_2 \\ 1, & \text{else} \end{cases} \quad (22)$$

where  $\rho_{gk}(\lambda)$  is calculated by (10);  $\Delta \bar{J}_1$  is a small value, and  $\Delta \bar{J}_2$  is a relatively greater number. The usage of this extended quality assessment will be explained later.

In addition, we define the basic and enlarged bounding boxes for a point cloud, which will be used by FuzzyPSReg-O2S. The definition is shown in Fig. 3.

Now we present FuzzyPSReg-O2S. To align  $\mathbb{P}_o$  and  $\mathbb{P}_s$ , we need to resolve a global optimization problem with multiple local optima. For this issue, FuzzyPSReg-O2S is designed as a process of elimination running in a repeatable coarse-to-fine fashion. Algorithm 2 and the following steps describe the process. Further explanations are in the discussion given later.

*Step 1: Fuzzy clustering of the object model:* This step, relating to lines 1–2 of Algorithm 2, is used to obtain both FCM and GK clustering results of the object model  $\mathbb{P}_o$ . A relatively small number (e.g., in [50, 100]) is chosen as the fuzzy cluster number  $N_{C_o}$  for  $\mathbb{P}_o$ . Afterward, FCM clustering is applied to  $\mathbb{P}_o$  to derive  $N_{C_o}$  fuzzy cluster centers, denoted as  $\mathbb{C}_o^{(fcm)} = \{\mathbf{c}_{oi}^{(fcm)}, i = 1, \dots, N_{C_o}\}$ , and the average distance loss  $\bar{J}_{fcm}(\mathbb{P}_o, \mathbb{C}_o^{(fcm)})$ . Next, GK clustering is applied to  $\mathbb{P}_o$  to derive another group of  $N_{C_o}$  fuzzy cluster centers, denoted as  $\mathbb{C}_o^{(gk)} = \{\mathbf{c}_{oi}^{(gk)}, i = 1, \dots, N_{C_o}\}$ , and the norm-inducing matrices  $\mathbb{A}_o = \{\mathbf{A}_{oi}, i = 1, \dots, N_{C_o}\}$ , as well as the average distance loss  $\bar{J}_{gk}(\mathbb{P}_o, \mathbb{C}_o^{(gk)})$ .

*Step 2: Coarse registration:* This step, relating to lines 3–10 of Algorithm 2, is used to find a rough pose of the object in the scene. If the number of points in the scene scan  $\mathbb{P}_s$  is smaller than a predefined threshold  $\underline{N}_{P_s}$ , we will collect a new scene scan as  $\mathbb{P}_s$ . In this step, a registration is performed using the FCM-based metric (4), where the fixed and moving sets, as well as their fuzzy cluster centers, are denoted as  $\mathbb{P}_F^{(1)}$ ,  $\mathbb{P}_M^{(1)}$ ,  $\mathbb{C}_F^{(1)}$ , and  $\mathbb{C}_M^{(1)}$ , respectively. We let  $\mathbb{P}_F^{(1)} = \mathbb{P}_s$  and  $\mathbb{P}_M^{(1)} = \mathbb{P}_o$ . For  $\mathbb{P}_F^{(1)}$ , FCM clustering is applied online to derive  $\mathbb{C}_F^{(1)} = \{\mathbf{c}_{Fi}^{(1)}, i = 1, \dots, N_{C_F^{(1)}}\}$  and  $\bar{J}_{fcm}(\mathbb{P}_F^{(1)}, \mathbb{C}_F^{(1)})$ , where  $N_{C_F^{(1)}}$  is selected as a relatively small number (e.g., in [50, 100]).

## Algorithm 2: FuzzyPSReg-O2S.

```

Input: The object model  $\mathbb{P}_o$ , the scene scan  $\mathbb{P}_s$ , the number of fuzzy clusters for  $\mathbb{P}_o$ ,  $N_{C_o}$ , and the number of fuzzy clusters for  $\mathbb{P}_s$ ,  $N_{C_s^{(1)}}$ .
Output:  $\lambda^* = (\mathbf{r}^*, \mathbf{t}^*)$  moving  $\mathbb{P}_o$  to align with  $\mathbb{P}_s$ .
1 Apply FCM clustering to  $\mathbb{P}_o$  to obtain  $N_{C_o}$  fuzzy cluster centers  $\mathbb{C}_o^{(fcm)}$  and  $\bar{J}_{fcm}(\mathbb{P}_o, \mathbb{C}_o^{(fcm)})$ .
2 Apply GK clustering to  $\mathbb{P}_o$  to obtain  $N_{C_o}$  fuzzy cluster centers  $\mathbb{C}_o^{(gk)}$ ,  $\mathbb{A}_o$ , and  $\bar{J}_{gk}(\mathbb{P}_o, \mathbb{C}_o^{(gk)})$ .
3 while True do
4   if the number of points in  $\mathbb{P}_s$  is below  $\underline{N}_{P_s}$  then
5     | Re-scan the scene to collect a new  $\mathbb{P}_s$ .
6   Set  $\mathbb{P}_F^{(1)} = \mathbb{P}_s$ ,  $\mathbb{P}_M^{(1)} = \mathbb{P}_o$ , and  $\mathbb{C}_M^{(1)} = \mathbb{C}_o^{(fcm)}$ .
7   Apply FCM clustering to  $\mathbb{P}_F^{(1)}$  to obtain  $N_{C_F^{(1)}}$  fuzzy cluster centers  $\mathbb{C}_F^{(1)}$  and  $\bar{J}_{fcm}(\mathbb{P}_F^{(1)}, \mathbb{C}_F^{(1)})$ .
8   Apply the multi-start optimization to the metric  $J_{fcm}(\lambda, \mathbb{C}_M^{(1)}, \mathbb{C}_F^{(1)})$  with the trimming ratio  $\xi_1$  ( $\xi_1 = 0.4$ ) to derive  $\lambda^{(1)}$  and the segment  $\mathbb{P}_{\lambda^{(1)}}$ .
9   if  $\rho_{fcm}(\lambda^{(1)}) > 1$  then
10    | Remove  $\mathbb{P}_{\lambda^{(1)}}$  from  $\mathbb{P}_s$ .
11   else
12    | Set  $\mathbb{P}_F^{(2)} = \mathbb{P}_o$ ,  $\mathbb{P}_M^{(2)} = T(\lambda_{inv}^{(1)}, \mathbb{P}_{\lambda^{(1)}})$ ,
13      |  $\mathbb{C}_F^{(2)} = \mathbb{C}_o^{(gk)}$ ,  $\mathbb{A}_F^{(2)} = \mathbb{A}_o$ , and
14      |  $\bar{J}_{gk}(\mathbb{P}_F^{(2)}, \mathbb{C}_F^{(2)}) = \bar{J}_{gk}(\mathbb{P}_o, \mathbb{C}_o^{(gk)})$ .
15    | Apply the multi-start local optimization to the metric  $J_{gk}(\lambda, \mathbb{P}_M^{(2)}, \mathbb{C}_F^{(2)})$  with the trimming ratio  $\xi_2$  ( $\xi_2 = 0$ ) to derive  $\lambda^{(2)}$ .
16    if  $q_{gk}(\lambda^{(2)}) = 1$  then
17      | Calculate  $\lambda^*$  by (24), and break.
18    if  $q_{gk}(\lambda^{(2)}) = -1$  then
19      | Remove  $\mathbb{P}_{\lambda^{(1)}}$  from  $\mathbb{P}_s$ .
20    else
21      | The user is involved to check the result.
22    if  $\lambda^{(1)}$  gives a wrong position then
23      | Remove  $\mathbb{P}_{\lambda^{(1)}}$  from  $\mathbb{P}_s$ .
24    else
25      | Calculate  $\lambda^{(1,2)}$  by (24), and derive the segment  $\mathbb{P}_{\lambda^{(1,2)}}$ . Then, set  $\mathbb{P}_F^{(3)} = \mathbb{P}_o$  and  $\mathbb{P}_M^{(3)} = T(\lambda_{inv}^{(1,2)}, \mathbb{P}_{\lambda^{(1,2)}})$ , and apply FuzzyPSReg-SS to  $\mathbb{P}_F^{(3)}$  and  $\mathbb{P}_M^{(3)}$  with the user-defined  $\xi_3$  to derive  $\lambda^{(3)}$ . Finally, calculate  $\lambda^*$  and break.

```

For  $\mathbb{P}_M^{(1)}$ , we have  $\mathbb{C}_M^{(1)} = \mathbb{C}_o^{(fcm)}$  from Step 1. Since  $\mathbb{P}_F^{(1)}$  and  $\mathbb{P}_M^{(1)}$  partially overlap each other as described before, trimming is applied. The trimming ratio, denoted as  $\xi_1$ , is selected as  $\xi_1 = 0.4$  in this study. Then we have the FCM-based registration metric  $J_{fcm}(\lambda, \mathbb{C}_M^{(1)}, \mathbb{C}_F^{(1)})$ , and apply the following *multistart optimization* to the metric.

- Initialize  $N_\lambda$  transformations  $\lambda_i = (\mathbf{r}_i, \mathbf{t}_i)$ ,  $i = 1, \dots, N_\lambda$ , and calculate the distance loss

$J_{fcm}(\lambda_i, \mathbb{C}_M'^{(1)}, \mathbb{C}_F^{(1)})$  of each  $\lambda_i$ . These  $\lambda_i$  can be randomly generated as follows. The rotation  $r_i$  is selected in  $[-\pi, \pi]^3$  (the entire 3-D rotation space), and the translation  $t_i$  is selected according to the basic bounding box of  $\mathbb{P}_F^{(1)}$ .

- ii) From the  $N_\lambda$  initial  $\lambda_i$ , pick out  $N_{\lambda_f}$  transformations with the lowest distance losses. Then, refine each of them by applying the gradient-based optimization algorithm to the metric  $J_{fcm}(\lambda, \mathbb{C}_M'^{(1)}, \mathbb{C}_F^{(1)})$  to have  $N_{\lambda_f}$  refined transformations and their distance losses.
- iii) Denote the refined transformation with the lowest distance loss as  $\lambda^{(1)} = (r^{(1)}, t^{(1)})$ , and denote the part of  $\mathbb{P}_s$  lying in the basic bounding box of  $T(\lambda^{(1)}, \mathbb{P}_o)$  as  $\mathbb{P}_{\lambda^{(1)}}$ . If  $\lambda^{(1)}$  satisfies (5) that  $\rho_{fcm}(\lambda^{(1)}) < 1$ , then Step 2 is complete. Otherwise, remove  $\mathbb{P}_{\lambda^{(1)}}$  from  $\mathbb{P}_s$  to repeat Step 2.

*Step 3: Fine registration:* This step, relating to lines 11–13 of Algorithm 2, is used to refine the coarse result of Step 2. In this step, a registration is performed using the GK-based metric (9), where the fixed and moving set are denoted as  $\mathbb{P}_F^{(2)}$  and  $\mathbb{P}_M^{(2)}$ , respectively. We let  $\mathbb{P}_F^{(2)} = \mathbb{P}_o$  and  $\mathbb{P}_M^{(2)} = T(\lambda_{inv}^{(1)}, \mathbb{P}_{\lambda^{(1)}})$ , where  $\lambda_{inv}^{(1)} = (r_{inv}^{(1)}, t_{inv}^{(1)})$  denotes the inverse transformation of  $\lambda^{(1)} = (r^{(1)}, t^{(1)})$ , and is calculated by

$$\begin{cases} \mathbf{R}(r_{inv}^{(1)}) = \mathbf{R}(r^{(1)})^{-1} \\ t_{inv}^{(1)} = -\mathbf{R}(r_{inv}^{(1)}) \cdot t^{(1)}. \end{cases} \quad (23)$$

For  $\mathbb{P}_F^{(2)}$ , we already have its fuzzy cluster centers  $\mathbb{C}_F^{(2)} = \mathbb{C}_o^{(gk)}$ , norm-inducing matrices  $\mathbb{A}_F^{(2)} = \mathbb{A}_o$ , and average distance loss  $\bar{J}_{gk}(\mathbb{P}_F^{(2)}, \mathbb{C}_F^{(2)}) = \bar{J}_{gk}(\mathbb{P}_o, \mathbb{C}_o^{(gk)})$  from Step 1. If  $\lambda^{(1)}$  moves the object model to a correct position, then  $\mathbb{P}_M^{(2)}$ , the transformed  $\mathbb{P}_{\lambda^{(1)}}$ , is a partial scan of the object. Thus, ideally, this step performs a partial-to-full registration, where the trimming ratio, denoted by  $\xi_2$ , can be set as zero. Then we have the GK-based metric  $J_{gk}(\lambda, \mathbb{P}_M'^{(2)}, \mathbb{C}_F^{(2)})$ , and apply the following *multistart local optimization* to the metric.

- i) Specify  $N_r$  rotations  $r_i$ ,  $i = 1, \dots, N_r$ , including zero  $r_i = (0, 0, 0)$ , to have  $N_r$  initial estimates  $\lambda_i = (r_i, t_i)$ , where  $t_i = (0, 0, 0)$  for  $i = 1, \dots, N_r$ .
- ii) For each  $\lambda_i$  ( $i = 1, \dots, N_r$ ), refine it through applying the gradient-based optimization algorithm to the metric  $J_{gk}(\lambda, \mathbb{P}_M'^{(2)}, \mathbb{C}_F^{(2)})$  to obtain its refined transformation and the associated distance loss.
- iii) From the  $N_r$  refined transformations, select the one with the lowest distance loss to be  $\lambda^{(2)} = (r^{(2)}, t^{(2)})$ .

*Step 4: Registration quality assessment:* This step, relating to lines 14–21 of Algorithm 2, is used to check the accuracy of  $\lambda^{(2)}$  and then take corresponding actions. In Step 3, the size difference between  $\mathbb{P}_F^{(2)}$  and  $\mathbb{P}_M^{(2)}$  is not large, and thus, the registration quality assessments in (11) and (22) can be applied to evaluate the quality of  $\lambda^{(2)}$ . We calculate  $\rho_{gk}(\lambda^{(2)})$  by (10) and then  $q_{gk}(\lambda^{(2)})$  by (22). A human–computer shared method is used to deal with the three possible values of  $q_{gk}(\lambda^{(2)})$ . This method

together with the explanation of the extended registration quality assessment are presented as follows:

- i)  $q_{gk}(\lambda^{(2)}) = 1$ . In this case,  $\rho_{gk}(\lambda^{(2)})$  is smaller than  $1 + \Delta \bar{J}_1$ . According to (11),  $T(\lambda^{(2)}, \mathbb{P}_M^{(2)})$  and  $\mathbb{P}_F^{(2)}$  are correctly aligned. Hence,  $\mathbb{P}_o$  and  $\mathbb{P}_s$  are correctly aligned based on  $\lambda^{(1)}$  and  $\lambda^{(2)}$ . Note that  $\lambda^{(1)}$  moves  $\mathbb{P}_o$  to  $\mathbb{P}_s$ , and  $\lambda^{(2)}$  moves  $\mathbb{P}_s$  to  $\mathbb{P}_o$  since  $\mathbb{P}_M^{(2)}$  is a part of  $\mathbb{P}_s$ . By combining  $\lambda^{(1)}$  and inverse  $\lambda^{(2)}$  using (24), we have the transformation, denoted as  $\lambda^{(1,2)} = (r^{(1,2)}, t^{(1,2)})$ , moving  $\mathbb{P}_o$  to  $\mathbb{P}_s$ , and the final result  $\lambda^* = \lambda^{(1,2)}$

$$\begin{cases} \mathbf{R}(r^{(1,2)}) = \mathbf{R}(r^{(1)}) \cdot \mathbf{R}(r^{(2)})^{-1} \\ t^{(1,2)} = t^{(1)} - \mathbf{R}(r^{(1,2)}) \cdot t^{(2)}. \end{cases} \quad (24)$$

- ii)  $q_{gk}(\lambda^{(2)}) = -1$ . In this case,  $\rho_{gk}(\lambda^{(2)})$  exceeds  $1 + \Delta \bar{J}_2$ , which means the distance loss  $\bar{J}_{gk}(\lambda^{(2)}, \mathbb{P}_M^{(2)}, \mathbb{C}_F^{(2)})$  is too large compared to  $\bar{J}_{gk}(\mathbb{P}_F^{(2)}, \mathbb{C}_F^{(2)})$ . This case indicates an erroneous result that  $\lambda^{(1)}$  moves the object to a wrong place and  $\mathbb{P}_{\lambda^{(1)}}$  is not a part of the object. Thus, we remove  $\mathbb{P}_{\lambda^{(1)}}$  from  $\mathbb{P}_s$  and repeat Steps 2–4 with the pruned  $\mathbb{P}_s$ .
- iii)  $q_{gk}(\lambda^{(2)}) = 0$ . In this case,  $\rho_{gk}(\lambda^{(2)})$  is greater than  $1 + \Delta \bar{J}_1$  but does not exceed  $1 + \Delta \bar{J}_2$ , which has two possibilities: one is that  $\lambda^{(1)}$  moves the object to a wrong place, and the other is that  $\lambda^{(1)}$  gives an acceptable position but the pose needs further correction. To deal with this uncertainty, the user is involved to check the result. For the former possibility, same as that of  $q_{gk}(\lambda^{(2)}) = -1$ , the scene scan is pruned to repeat Steps 2–4. For the latter possibility, the following Step 5 is applied.

*Step 5: Pose correction using FuzzyPSReg-SS:* This step, relating to lines 22–23 of Algorithm 2, is used to correct the result passed from Step 4. When this step is applied, the transformation  $\lambda^{(1,2)}$  calculated by (24) moves  $\mathbb{P}_o$  to an acceptable position in  $\mathbb{P}_s$  as stated in Step 4. Then, we segment the part of  $\mathbb{P}_s$  lying in the enlarged bounding box of  $T(\lambda^{(1,2)}, \mathbb{P}_o)$  and denote this part as  $\mathbb{P}_{\lambda^{(1,2)}}$ . This step performs a registration, where the fixed and moving sets are denoted as  $\mathbb{P}_F^{(3)}$  and  $\mathbb{P}_M^{(3)}$ , respectively. We let  $\mathbb{P}_F^{(3)} = \mathbb{P}_o$  and  $\mathbb{P}_M^{(3)} = T(\lambda_{inv}^{(1,2)}, \mathbb{P}_{\lambda^{(1,2)}})$ , where  $\lambda_{inv}^{(1,2)}$  is the inverse transformation of  $\lambda^{(1,2)}$  and calculated using the way in (23). Since  $\mathbb{P}_F^{(3)}$  and  $\mathbb{P}_M^{(3)}$  have similar sizes, we use FuzzyPSReg-SS to achieve the registration. For  $\mathbb{P}_F^{(3)}$ , the FCM and GK clustering results have been obtained in Step 1. For  $\mathbb{P}_M^{(3)}$ , FuzzyPSReg-SS will apply FCM clustering online to derive its fuzzy clusters, where the fuzzy cluster number is chosen to be  $N_{C_o}$ . In addition,  $\mathbb{P}_F^{(3)}$  and  $\mathbb{P}_M^{(3)}$  may partially overlap each other. Since the user has been involved, s/he is asked to select a trimming ratio, denoted as  $\xi_3$ , for this registration. After applying FuzzyPSReg-SS, a transformation, denoted as  $\lambda^{(3)} = (r^{(3)}, t^{(3)})$ , is derived to align  $\mathbb{P}_F^{(3)}$  and  $\mathbb{P}_M^{(3)}$ . Then, the final result  $\lambda^* = (r^*, t^*)$  is calculated by (24) with  $\lambda^{(1)}$ ,  $\lambda^{(2)}$ , and  $\lambda^{(1,2)}$  replaced by  $\lambda^{(1,2)}$ ,  $\lambda^{(3)}$ , and  $\lambda^*$ , respectively.

*Discussion of FuzzyPSReg-O2S:* FuzzyPSReg-O2S effectively combines the two fuzzy cluster-based metrics to align two point clouds with a relatively large difference in size. It may

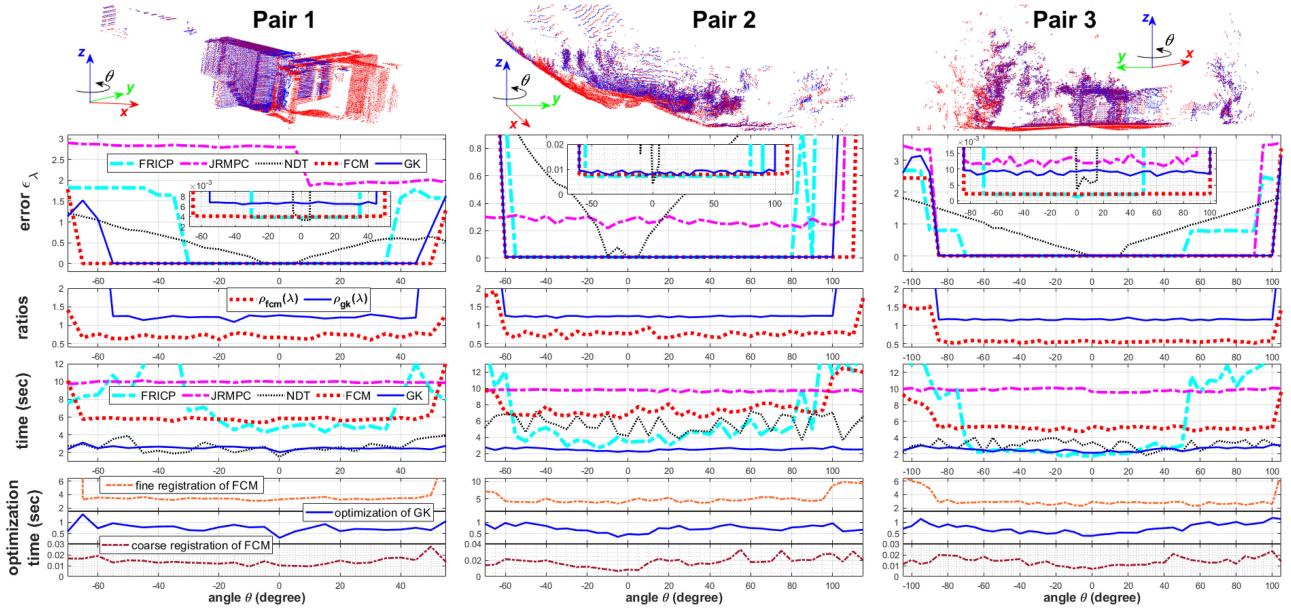


Fig. 4. Local convergence tests for five registration metrics: FRICP [5], JRMPG [13], NDT [12], the FCM-based metric [18], and the proposed GK-based metric. The first row shows the three scan pairs from [42], where Pairs 1, 2, and 3 belong to the datasets named “Stairs”, “Mountain plain,” and “Gazebo in summer,” respectively; the second row shows the registration errors, where the inset plots are the enlarged figures; the third row shows  $\rho_{fcm}(\lambda)$  and  $\rho_{gk}(\lambda)$  of the FCM- and GK-based metrics; the fourth row shows the time costs; and the fifth row shows the optimization time of the FCM- and GK-based metrics.

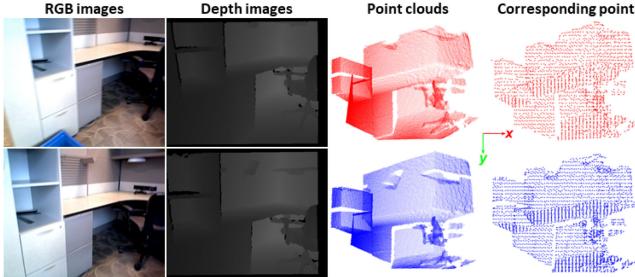


Fig. 5. Point cloud pair from the SUN3D dataset [43] used in the tests of Section V-B. The corresponding points are obtained from [16], where the number of outliers takes up about 50% of the total matches. Note that the RGB information is not used in any registration methods of this article.

need several rounds of the coarse-to-fine registration to obtain the alignment. We will take the processes shown in Figs. 15 and 16 as examples in the following discussion.

In Step 1, the fuzzy clustering of the object model  $\mathbb{P}_o$  can be performed offline and does not take up the online costs. In Step 2, the two point clouds,  $\mathbb{P}_F^{(1)} = \mathbb{P}_s$  and  $\mathbb{P}_M^{(1)} = \mathbb{P}_o$ , are described by relatively small numbers of fuzzy clusters, such as  $N_{C_F^{(1)}} = N_{C_M^{(1)}} = 50$  in Figs. 15 and 16. Besides, the FCM-based metric is used for their registration. Hence, the online computational cost is low, and the result is a coarse alignment if  $\lambda^{(1)}$  is a correct solution. To further reduce the cost, we can down-sample  $\mathbb{P}_s$  for its online FCM fuzzy clustering. Generally, the depth sensor used to collect  $\mathbb{P}_s$  is installed above the scene and facing downwards, which can view about half of the object’s surface. Thus, the trimming ratio  $\xi_1$  can be around 0.5. According to [18], it is better to choose a slightly smaller trimming ratio for global search.

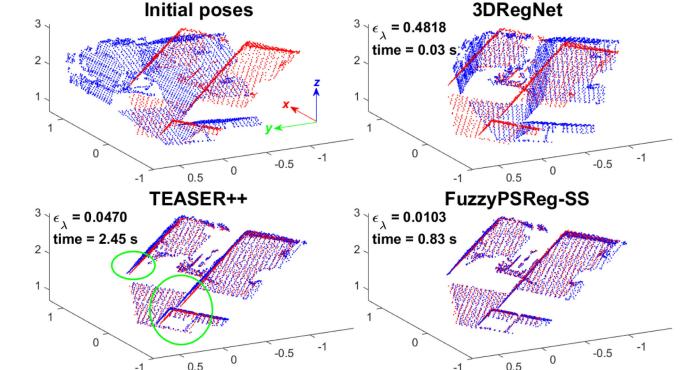


Fig. 6. Registration results of the three methods for the point cloud pair shown in Fig. 5. 3DRegNet cannot align the two point clouds. TEASER++ leaves a small misalignment as marked by the green ellipses. FuzzyPSReg-SS achieves a precise alignment. This figure uses a different view from that of Fig. 5 in order to clearly show the small misalignment of TEASER++.

Therefore,  $\xi_1 = 0.4$  is used in this study. If  $\mathbb{P}_o$  is a partial instead of full model of the object, or  $\mathbb{P}_s$  is a full map instead of a partial scan of the scene, users can select an appropriate  $\xi_1$  according to actual conditions. For example, when  $\mathbb{P}_o$  is similar to the part of the scene scan corresponding to the object,  $\xi_1$  can be zero. In the multistart optimization, the initial transformations are selected by a uniform random number generator, and thus, they are evenly distributed in the range of the scene. Owing to the broad convergence basin of the FCM-based metric [18], it is likely to locate a rough position of the object by refining some selected initial transformations, like Fig. 15(b). If some possible poses of the object in the scene are already known, the initial  $\lambda_i$  can be manually determined. According to Remark 2, (5) is a

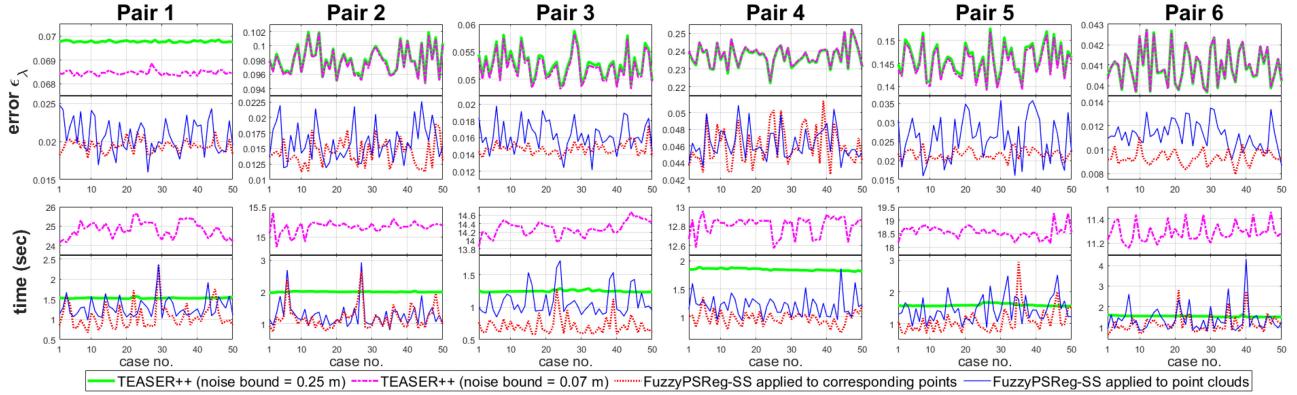


Fig. 7. Registration results of TEASER++ and FuzzyPSReg-SS for the six pairs shown in Fig. 8 with random initial poses.

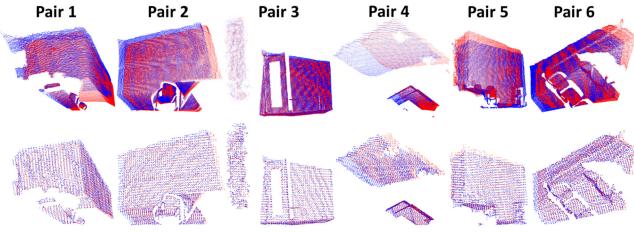
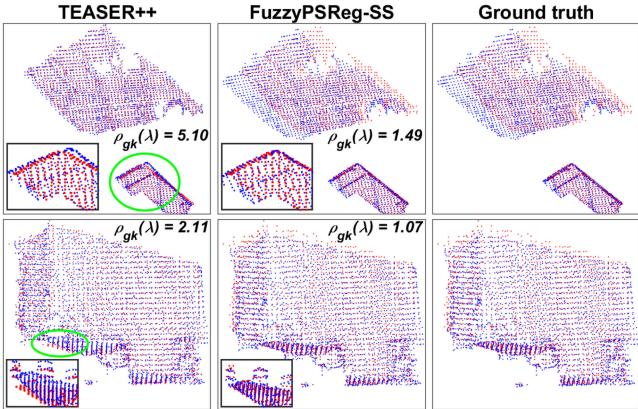


Fig. 8. Six pairs selected from the SUN3D dataset [43]. Fixed sets are in red and moving sets are in blue. In each pair, the two scans partially overlap each other. The first row shows the point clouds (each contains about 300 000 points) and the second row shows the corresponding points given by [16].

Fig. 9. Registration results of TEASER++ and FuzzyPSReg-SS for Pair 4 (in the first row) and Pair 5 (in the second row). As marked by the green ellipses, apparent misalignments remain in the results of TEASER++. Using the fuzzy clustering results of FuzzyPSReg-SS, we also calculate  $\rho_{gk}(\lambda)$  of  $\lambda$  derived by TEASER++, which is greater than  $\rho_{gk}(\lambda)$  derived by FuzzyPSReg-SS.

necessary condition for a correct solution. If (5) is not satisfied, it means that  $\lambda^{(1)}$  moves  $\mathbb{P}_o$  to a wrong place. Hence, we prune this wrong place and repeat Step 2 with the pruned  $\mathbb{P}_s$ , which can prevent this misalignment from happening again. If  $\lambda^{(1)}$  satisfies (5), it implies that  $\mathbb{P}_o$  may move to the correct place. The following steps will perform a check.

In Step 3, the GK-based metric is used for the refinement. Although  $\mathbb{P}_F^{(2)} = \mathbb{P}_o$  is described by a relatively small number of

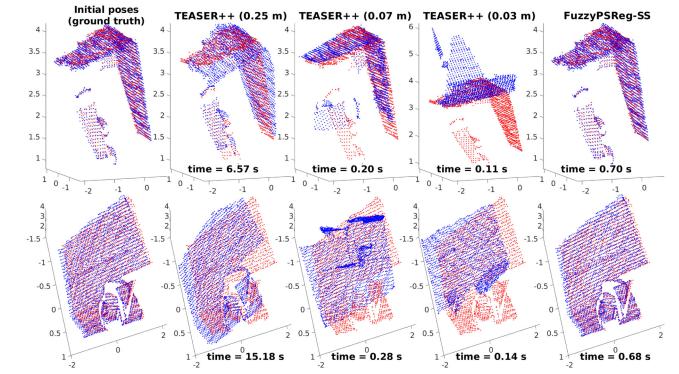


Fig. 10. Registration results of TEASER++ (using different noise bounds) and FuzzyPSReg-SS applied to the corresponding points of Pair 1 (in the first row) and Pair 2 (in the second row). The order of the points in each moving set (in blue) is randomly permuted.

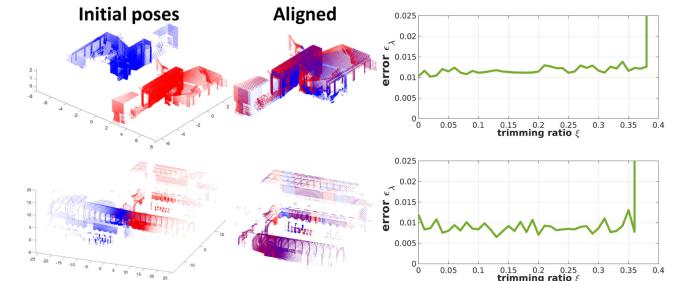


Fig. 11. Registration results of FuzzyPSReg-SS with different trimming ratios for two scan pairs from the challenging dataset [42]. The overlap ratios of the scans in the first row and the second row are about 80% and 90%, respectively.

fuzzy clusters ( $N_{C_o}$ ), according to Section III-C, the refinement can give a satisfactory result if  $\lambda^{(1)}$  is a good solution, and the computational cost is low. We can down-sample  $\mathbb{P}_M^{(2)} = T(\lambda_{inv}^{(1)}, \mathbb{P}_{\lambda^{(1)}})$  for the refinement to further reduce the cost. Besides, the quality assessment of the GK-based metric in (11) is more sensitive to small misalignments, and thus, can provide a better evaluation (used in the next step) when compared with that of the FCM-based metric in (5). Note that  $\lambda^{(1)}$  from Step 2 may give an acceptable position but an incorrect orientation of

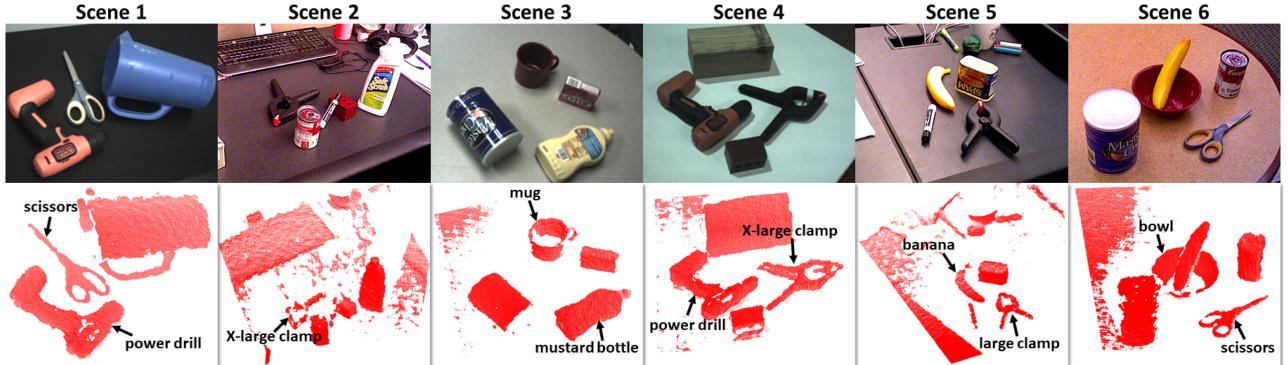


Fig. 12. Six scenes selected from the YCB-Video dataset [23]. The first and the second rows, respectively, show the pictures and the scene scans (point clouds). For each scene, we select one or two objects, and apply registration methods to align the model of each object with the scene scan for tests. The selected objects are marked in the scene scans (the second row). Note that the color images are not used in the registration tests.

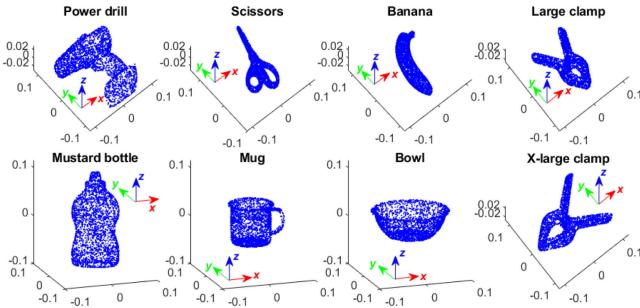


Fig. 13. Full models of the eight objects in the six scenes of Fig. 12.

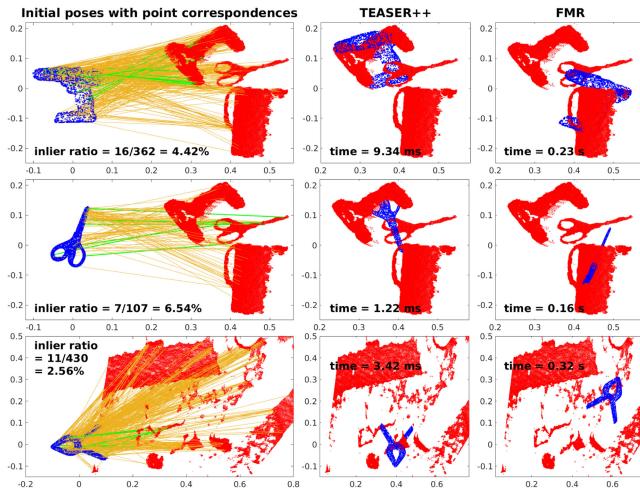


Fig. 14. Registration results of TEASER++ [8] and FMR [17] for the three cases of Scene 1 and Scene 2. FPFH features [34] are used to build point correspondences of each pair, where inliers are in green and outliers are in orange. TEASER++ is applied to the correspondences while FMR is applied to the point clouds without needing to know correspondences. In the three cases, TEASER++ and FMR do not provide correct alignments.

the object, like Fig. 15(b), and the incorrect orientation may be outside the convergence basin of the GK-based metric using zero as the initialization, like Fig. 15(c). Hence, in the multistart local optimization, in addition to zero, we also apply several different

rotations as the initial transformations for the refinement to help correct the orientation. As shown in Fig. 15(c),  $N_r = 4$  initial rotations are applied, and  $r_2 = (\pi, 0, 0)$  gives a good result. The  $N_r$  rotations can be manually selected according to the shape of the object. In the experiment, we will specify the  $N_r$  rotations used by each object. Note that in the fine registration of Step 3, the trimming ratio is  $\xi_2 = 0$ . As shown in Fig. 15(c), although some outliers are included into  $\mathbb{P}_M^{(2)}$ , the GK-based metric with zero trimmings can still provide a good result.

In Step 4, using the extended quality assessment (22), the accurate results with  $q_{gk}(\lambda^{(2)}) = 1$  and inaccurate results with  $q_{gk}(\lambda^{(2)}) = -1$  can be automatically detected. For uncertain results with  $q_{gk}(\lambda^{(2)}) = 0$ , the user will be involved to check. When the result is erroneous, like Round 2 of the second row in Fig. 16, the user will choose to prune the scan and perform a new round of the coarse-to-fine registration. When the uncertain result gives an acceptable position of the object, such as that shown in Fig. 15(d) and Round 3 in Fig. 16, the user will choose to go to Step 5, where FuzzyPSReg-SS is applied for orientation correction or further refinement.

In the registration of Step 5, the fixed set is  $\mathbb{P}_F^{(3)} = \mathbb{P}_o$ , and the moving set is a new segment  $\mathbb{P}_M^{(3)} = T(\lambda_{inv}^{(1,2)}, \mathbb{P}_{\lambda^{(1,2)}})$ . To include more points corresponding to the object into  $\mathbb{P}_{\lambda^{(1,2)}}$ , the enlarged bounding box of the object is used for the segmentation. In this way, more outliers may also be included into  $\mathbb{P}_{\lambda^{(1,2)}}$ . For example, compared to  $\mathbb{P}_{\lambda^{(1)}}$  derived by the basic bounding box in Fig. 15(b),  $\mathbb{P}_{\lambda^{(1,2)}}$  derived by the enlarged bounding box in Fig. 15(d) includes all the points of the object (the power drill) in the scene scan but also contains more outliers (corresponding to the scissors). To address this issue, the user is asked to select the trimming ratio  $\xi_3$  since he/she has been involved in checking. Note that FuzzyPSReg-SS performs robustly when the trimming ratio varies in a relatively large interval (a test is shown in Fig. 11). Therefore, the user can roughly choose  $\xi_3$  for FuzzyPSReg-SS to align  $\mathbb{P}_F^{(3)}$  and  $\mathbb{P}_M^{(3)}$ . As shown in Figs. 15 and 16, FuzzyPSReg-SS using user-defined  $\xi_3$  provides accurate results. To improve efficiency,  $\mathbb{P}_M^{(3)}$  can be down-sampled for the registration.

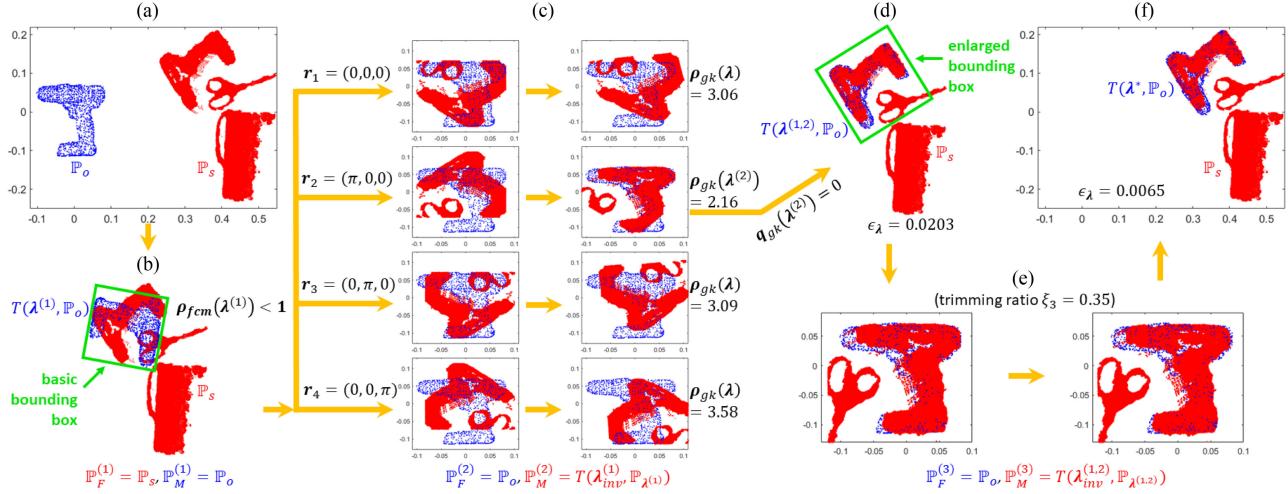


Fig. 15. Process of FuzzyPSReg-O2S aligning the power driller's model (in blue) with Scene 1s scan (in red). (a) Initial poses of  $\mathbb{P}_o$  and  $\mathbb{P}_s$ . (b) Coarse registration result, where  $\rho_{fcm}(\lambda^{(1)}) < 1$ . The part of the scene scan lying in the basic bounding box of the transformed object model,  $\mathbb{P}_{\lambda^{(1)}}$ , is segmented for the fine registration. (c) Fine registration result, where the refined transformation with  $r_2$  as the initialization gives the lowest distance loss and is selected as  $\lambda^{(2)}$ . (d)  $\lambda^{(2)}$  gives  $q_{gk}(\lambda^{(2)}) = 0$  according to (22), and the user is involved to check the result. The user chooses to run FuzzyPSReg-SS. Then, the enlarged bounding box of the transformed model is applied to derive a new segment,  $\mathbb{P}_{\lambda^{(1,2)}}$ , and the user chooses  $\xi_3 = 0.35$ . (e) FuzzyPSReg-SS gives a more precise alignment. (f) Final poses of the two point clouds. The time costs for the coarse registration, the fine registration, and FuzzyPSReg-SS are 0.20, 1.56, and 0.25 s, respectively. The total time cost of this case (excluding the checking time taken by the user) is 2.01 s.

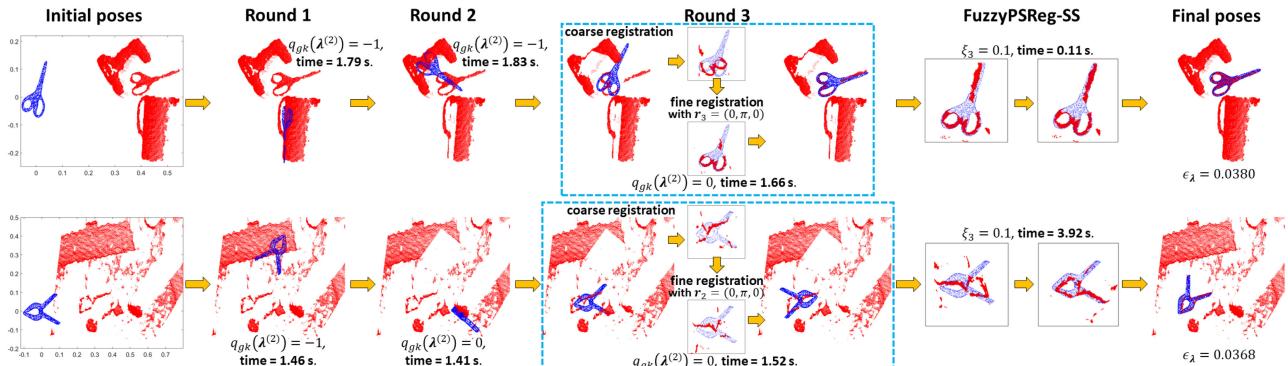


Fig. 16. Registration results of FuzzyPSReg-O2S. In the first row, the scissors' model is aligned with Scene 1s scan. In the second row, the X-large clamp's model is aligned with Scene 2s scan. In each case, three rounds of the coarse-to-fine registration are utilized, and FuzzyPSReg-O2S gives a correct alignment. Excluding the checking time taken by the user, the time costs of FuzzyPSReg-O2S in the cases of the first and second rows are 5.39 and 8.31 s, respectively.

Regarding the extended registration quality assessment (22), if  $\Delta\bar{J}_1$  is chosen to be overlarge, it will give a false positive evaluation; if  $\Delta\bar{J}_2$  is chosen to be too small, the part of the object may be pruned from the scene scan. When  $\Delta\bar{J}_1$  is chosen to be too small or  $\Delta\bar{J}_2$  is chosen to be too large, the frequency of involving the user in checking will increase, but the alignment result will not be affected.

## V. EXPERIMENT

This section tests and compares the proposed method<sup>1</sup> with state-of-the-art registration techniques. In each test, we denote the ground truth transformation as  $\lambda_{gt}$  and define the error of the result  $\lambda$  to be:  $\epsilon_\lambda = \|\lambda - \lambda_{gt}\|$ . A rotation error of  $1^\circ$

<sup>1</sup>Source code is [Online]. Available: <https://gitsvn-nt.oru.se/qianfang.liao/FuzzyPSReg>

will cause an increment of  $\epsilon_\lambda$  to be 0.0175. For the two fuzzy clustering algorithms and the two fuzzy cluster-based metrics, we choose  $m = 2$ . The fuzzy clustering in the two FuzzyPSReg strategies is performed in the following way: FCM clustering uses random initialization and runs 100 iterations to minimize its objective function (2); GK clustering takes the FCM clustering results as the initialization and runs 30 iterations to minimize its objective function (6). Other parameter settings are as follows. In FuzzyPSReg-SS,  $N_{C_F}$  and  $N_{C_M}$  will be specified in each test. To improve efficiency,  $\mathbb{P}_F$  and  $\mathbb{P}_M$  may be down-sampled for the fuzzy clustering and the fine registration. The parameters will be specified when the down-sampling is applied. In FuzzyPSReg-O2S, for Step 1, we choose  $N_{C_o} = 50$ . For Step 2, we set the threshold  $\underline{N}_{P_s}$  as 3000.  $\mathbb{P}_s$  is down-sampled for its online FCM clustering using box grid filter with the grid step as 0.005 m, and  $N_{C_F^{(1)}} = 50$ . In the multi-start optimization,

$N_\lambda = 100$  and  $N_{\lambda_f} = 10$ . For Step 3, the  $N_r$  rotations  $\mathbf{r}_i$  of each object in the multi-start local optimization will be given.  $\mathbb{P}_M^{(2)}$  is down-sampled for each refinement using box grid filter with the grid step as 0.005 m. For Step 4, we choose  $\Delta \bar{J}_1 = 0.05$  and  $\Delta \bar{J}_2 = 2$  for the extended registration quality assessment. For Step 5,  $\mathbb{P}_M^{(3)}$  is down-sampled for FuzzyPSReg-SS using box grid filter with the grid step as 0.005 m. All the tests are implemented on a computer with the Intel Core i9-8950HK CPU.

### A. Tests of Local Convergences

In this section, we compare the local convergence basins of the two fuzzy cluster-based metrics and three state-of-the-art registration methods: FRICP [5], JRMPC [13], and NDT [12]. Three range scan pairs selected from the challenging dataset [42] are employed for the tests, as shown in the first row of Fig. 4. In each pair, the fixed set (red) and the moving set (blue) have around 15 300 and 12 300 points, respectively, and they are normalized with the same scaling factor to fit the cube  $[-1, 1]^3$ . In each test, the moving set rotates about the  $z$ -axis by an angle  $\theta$  (as shown in Fig. 4) to deviate from its alignment with the fixed set. Then, we apply each of the five registration methods using local optimization to this misalignment to see whether the two point clouds can be registered.

To test the FCM-based metric (4), the coarse-to-fine method in [18] is applied using local optimization only: first, each scan is described by 80 fuzzy clusters through FCM clustering, where random initialization is utilized and the iteration number is 100. Afterward, the gradient-based optimization algorithm is applied to the FCM-based metric with zero as the initialization to derive a coarse alignment. Next, each scan is down-sampled to around 2000 points, and these points are directly taken as the fuzzy cluster centers to update the metric. Finally, the gradient-based algorithm is applied to the updated FCM-based metric to refine the coarse alignment. To test the GK-based metric (9), first, the following two steps are applied to derive the GK clustering results of  $\mathbb{P}_F$ : Step i) FCM clustering is applied to  $\mathbb{P}_F$  with  $N_{C_F} = 80$ , where random initialization is employed and the iteration number is 70; Step ii) taking the FCM clustering results of  $\mathbb{P}_F$  as the initialization, GK clustering is applied to  $\mathbb{P}_F$  with  $N_{C_F} = 80$ , where the iteration number is 30. The total iteration number of fuzzy clustering in these two steps is  $70 + 30 = 100$ , which is the same as that to test the FCM-based metric. Consequently, we can give a fair comparison between the two fuzzy cluster-based metrics. After fuzzy clustering, we down-sample  $\mathbb{P}_M$  to around 2000 points, and form the GK-based metric based on the GK clustering results of  $\mathbb{P}_F$  and the down-sampled  $\mathbb{P}_M$ . Finally, the gradient-based algorithm is applied to the GK-based metric with zero as the initialization to derive a transformation for  $\mathbb{P}_M$ .

The other three metrics also take zero as the initialization of their local optimizations in each test. For FRICP, its robust point-to-point metric using Welsch's function and Anderson acceleration is applied; for JRMPC [13], the number of components is selected as 200, and the other parameters use the default setting; and for NDT [12], the grid steps are selected

between 0.03 and 0.06 such that the fixed set and the moving set have around 500 and 450 components, respectively.

To give a fair comparison, the point clouds are not down-sampled in the fuzzy clustering when testing the two fuzzy cluster-based metrics. Besides, no trimming is applied in these tests. The results of the registration under different  $\theta$  are shown in the second to the fifth rows of Fig. 4.

From the second row of Fig. 4, the FCM-based metric gives the broadest convergence basin among the five metrics. In addition, its accuracy is comparable to FRICP and better than the other three metrics. Although the GK-based metric does not yield a convergence basin as broad as that of the FCM-based metric, and its accuracy based on 80 fuzzy clusters generally cannot be as high as that of the FCM-based metric using the coarse-to-fine method, it outperforms FRICP, JRMPC, and NDT in terms of convergence basin with fewer components, and its results within the convergence basin can be considered as satisfactory alignments (the rotation errors are below  $0.6^\circ$ ). Note that JRMPC cannot give correct alignments for Pair 1 and Pair 2 in all the tests.

From the third row of Fig. 4,  $\rho_{fcm}(\lambda)$  is smaller than 1 within the convergence basin of the FCM-based metric, and  $\rho_{gk}(\lambda)$  is a little greater than 1 within the convergence basin of the GK-based metric. The reason for  $\rho_{gk}(\lambda)$  being greater than 1 is that the two scans of each pair partially overlap each other, but no trimming is used in the registration. As a result, when they are aligned, the nonoverlapping part of the moving set, although takes up a small portion, will lead to an average distance loss  $\bar{J}_{gk}(\lambda, \mathbb{P}'_M, \mathbb{C}_F)$  greater than  $\bar{J}_{gk}(\mathbb{P}_F, \mathbb{C}_F)$ , where  $\mathbb{P}'_M = \mathbb{P}_M$  due to the zero trimming.

From the fourth row of Fig. 4, the GK-based metric requires low time costs. Especially for Pair 2, it takes the lowest time costs among the five metrics. The fifth row (last row) of Fig. 4 shows the time that the two fuzzy cluster-based metrics spend on optimization. The coarse registration of the FCM-based metric requires very low time costs and the fine registration of the FCM-based metric needs relatively high time costs. In addition, the optimization time of the GK-based metric is much shorter than the fine registration time of the FCM-based metric. Taking Pair 2 in Fig. 4 as an example, the fine registration time of the FCM-based metric is about 5 s, while the optimization time of the GK-based metric is less than 0.5 s in some test cases. The total time used by the FCM- or GK-based metric for registration, as shown in the fourth row of Fig. 4, is the sum of the optimization time and the fuzzy clustering time. For a fixed set with around 15 300 points, FCM clustering takes about 1.4 s, and GK clustering takes about 1.9 s including the time of FCM clustering to derive the initialization. For a moving set with around 12 300 points, FCM clustering takes about 1.2 s. If the point clouds are down-sampled for the fuzzy clustering, the two fuzzy cluster-based metrics can achieve the same results with reduced time costs.

### B. Tests of FuzzyPSReg-SS

In this section, we test and compare FuzzyPSReg-SS with two recently developed registration methods: 3DRegNet [16] and TEASER++ [8]. In [16], 13 different scenes are selected

TABLE I  
REGISTRATION COMPARISONS AMONG 3DREGNET, TEASER++, AND FUZZYPSREG-SS USING THE SUN3D DATA

Scene no.	3DRegNet		TEASER++		FuzzyPSReg-SS	
	$\epsilon_{\lambda}$ (mean / max)	time (mean / max)	$\epsilon_{\lambda}$ (mean / max)	time (mean / max)	$\epsilon_{\lambda}$ (mean / max)	time (mean / max)
1	0.1262 / 0.3982	0.0305 / 0.3812 s	0.1068 / 0.3432	1.4409 / 2.3149 s	0.0751 / 0.2990	0.7094 / 0.9167 s
2	0.1117 / 0.3526	0.0281 / 0.3965 s	0.0898 / 0.2841	1.4012 / 2.7066 s	0.0571 / 0.2535	0.8514 / 1.1460 s
3	0.2347 / 0.4230	0.0279 / 0.3946 s	0.0794 / 0.3009	1.5162 / 2.7716 s	0.0381 / 0.2157	0.8562 / 1.4371 s
4	0.2000 / 0.3869	0.0286 / 0.3971 s	0.0545 / 0.2247	1.5707 / 2.8370 s	0.0257 / 0.1564	0.8631 / 1.1780 s
5	0.3509 / 0.7082	0.0279 / 0.3987 s	0.1022 / 0.4308	1.4533 / 2.8113 s	0.0650 / 0.2731	0.8565 / 1.2885 s
6	0.3319 / 0.5969	0.0285 / 0.3964 s	0.0848 / 0.2971	1.5405 / 2.7328 s	0.0424 / 0.2780	0.8332 / 1.1253 s
7	0.5662 / 0.8124	0.0276 / 0.3910 s	0.0873 / 0.3767	1.3981 / 2.7796 s	0.0472 / 0.3108	0.7893 / 1.1673 s
8	0.5368 / 0.7557	0.0282 / 0.3962 s	0.1098 / 0.3531	1.5554 / 2.6803 s	0.0638 / 0.3189	0.8882 / 1.2913 s
9	0.6944 / 0.8835	0.0286 / 0.3935 s	0.0834 / 0.3582	1.4769 / 2.3864 s	0.0456 / 0.2026	0.7145 / 0.9787 s
10	0.6932 / 0.8638	0.0290 / 0.3868 s	0.0815 / 0.2614	1.4273 / 2.4202 s	0.0526 / 0.1989	0.8764 / 1.1157 s
11	0.0782 / 0.1702	0.0419 / 1.2575 s	0.0756 / 0.1646	1.5059 / 2.2383 s	0.0476 / 0.1332	0.7324 / 1.0209 s
12	0.1051 / 0.3069	0.0284 / 0.3856 s	0.0851 / 0.2566	1.4965 / 2.6743 s	0.0466 / 0.2149	0.8353 / 1.1707 s
13	0.1059 / 0.2416	0.0288 / 0.3906 s	0.0727 / 0.2470	1.5639 / 2.8206 s	0.0398 / 0.1813	0.7993 / 1.0584 s

from the SUN3D dataset [43] to train and test 3DRegNet, where 10 scenes are used for training and 3 scenes are used for testing. The 13 scenes contain more than 3700 different point cloud pairs. For each pair, about 3000 point correspondences are extracted using FPFH features [34], where the number of outliers is about 50% of the total matches [16]. An example is shown in Fig. 5. In this section, all the point cloud pairs of the 13 scenes are employed to test the registration methods. For the first test, we make some changes to the initial poses of some moving sets: From the 13 scenes, we select ten scenes, where nine scenes belong to the training data in [16] and one scene belongs to the testing data in [16]; then, we divide the selected ten scenes into five groups, each of which has two scenes. In the  $i$ th group ( $i = 1, \dots, 5$ ), the moving set of each pair rotates about the  $z$ -axis by  $i \times 10^\circ$ . The remaining three scenes keep unchanged. Afterward, we apply 3DRegNet, TEASER++, and FuzzyPSReg-SS to align each point cloud pair.

To test 3DRegNet, the trained model given by [16] is used. To test TEASER++, the noise bound is chosen as 0.25 m, and the other parameters use the default setting. Note that 3DRegNet and TEASER++ are correspondence-based registration methods, and we employ the corresponding points given by [16] for their tests; while FuzzyPSReg-SS does not require knowing the correspondences between two point clouds. To perform a fair comparison, FuzzyPSReg-SS is also applied to the corresponding points in the following way. For each pair, we let  $\mathbb{P}_F$  and  $\mathbb{P}_M$  be the corresponding points of the two point clouds, respectively, and choose  $N_{CF} = N_{CM} = 50$  for their fuzzy clustering. Besides, neither down-sampling nor trimming of points is applied. Table I presents the results. Note that Scenes 1, 12, and 13 are the testing data in [16], and the rest of the scenes are the training data in [16]. As stated before, Scenes 1–10 are selected and divided into 5 groups, and the initial poses of their moving sets are changed. Scene  $2i - 1$  and Scene  $2i$  constitute the  $i$ th group,  $i = 1, \dots, 5$ . The point cloud pair in Fig. 5 belongs to Scene 7 (the 4th group), and its registration results are shown in Fig. 6.

In Table I, from Scene 1 to Scene 10, as the angle between the initial poses of the two point clouds grows, the error of 3DRegNet increases. For the remaining 3 scenes that are unchanged, the initial pose difference of the two point clouds in each pair is relatively small. 3DRegNet achieves the average and maximum

errors for Scene 11 lower than 0.1 and 0.2, respectively, and provides greater average and maximum errors for Scenes 12 and 13 because Scene 11 is included in its training data while Scenes 12 and 13 are not. These results demonstrate that the convergence basin of 3DRegNet is relatively narrow, and the performance is restricted by its training data. For each scene, including Scene 11, although 3DRegNet requires the lowest time cost, it gives the largest average and maximum errors among the three methods. In addition, it requires a long training phase to obtain the neural network. For TEASER++, owing to its global optimization, it can align all the pairs regardless of their initial poses. However, it may not give a precise alignment, as the example shown in Fig. 6. For all the scenes, FuzzyPSReg-SS achieves the highest accuracy, and its time costs are lower than that of TEASER++.

In the first test, FuzzyPSReg-SS does not use global optimization, because the initial pose differences of all the pairs are within the local convergence basins of the fuzzy cluster-based metrics. To further test FuzzyPSReg-SS, we select six pairs from the 13 scenes to perform the second test. Fig. 8 shows the point clouds and corresponding points given by [16] of the six pairs. In the second test, we randomly choose 50 initial poses for the moving set of each pair and then apply TEASER++ and FuzzyPSReg-SS to align it with the fixed set.

Note that in the first test, TEASER++ gives similar results under different noise bounds, but takes much lower time costs when using a relatively large noise bound like 0.25 m. Hence, we choose 0.25 m as its noise bound in the first test. However, the 13 scenes of the SUN3D data are indoor scans, where the maximum distance between true corresponding points when the point clouds are aligned is relatively small and usually does not exceed 0.07 m. Therefore, in the second test, we apply TEASER++ twice to each pose of each pair using different noise bounds: 0.25 and 0.07 m. We also apply FuzzyPSReg-SS twice in different ways: The first way is to apply FuzzyPSReg-SS to the corresponding points same as the first test; and the second way is to apply FuzzyPSReg-SS to the point clouds. As shown in Fig. 8, a point cloud contains around 300 000 points. For efficiency, we down-sample each point cloud to about 5000 points for the fuzzy clustering of FuzzyPSReg-SS, where  $N_{CF} = N_{CM} = 50$ , and down-sample each moving set to about 3000 points for the fine registration of FuzzyPSReg-SS. No trimming is used in FuzzyPSReg-SS.

TABLE II  
REGISTRATION ERRORS  $\epsilon_\lambda$  AND TIME COSTS OF TEASER++ AND FUZZYPSREG-SS IN FIG. 7

Pair no.	TEASER++ (0.25 m)		TEASER++ (0.07 m)		FuzzyPSReg-SS (correspondences)		FuzzyPSReg-SS (point clouds)	
	$\epsilon_\lambda$ (mean / max)	time (mean / max)						
1	0.0698 / 0.0699	1.53 / 1.57 s	0.0685 / 0.0689	24.8 / 25.6 s	0.0194 / 0.0214	1.07 / 2.35 s	0.0207 / 0.0248	1.29 / 2.37 s
2	0.0980 / 0.1019	2.00 / 2.03 s	0.0978 / 0.1018	15.2 / 15.4 s	0.0144 / 0.0189	1.18 / 2.65 s	0.0160 / 0.0226	1.26 / 2.92 s
3	0.0527 / 0.0588	1.25 / 1.29 s	0.0521 / 0.0582	14.3 / 14.7 s	0.0147 / 0.0175	0.71 / 1.27 s	0.0162 / 0.0198	1.08 / 1.72 s
4	0.2373 / 0.2521	1.84 / 1.89 s	0.2378 / 0.2525	12.8 / 13.0 s	0.0461 / 0.0514	0.98 / 1.35 s	0.0463 / 0.0509	1.20 / 1.86 s
5	0.1458 / 0.1526	1.58 / 1.66 s	0.1452 / 0.1519	18.6 / 19.3 s	0.0216 / 0.0246	1.09 / 2.93 s	0.0258 / 0.0358	1.45 / 2.52 s
6	0.0410 / 0.0427	1.53 / 1.62 s	0.0410 / 0.0428	11.3 / 11.5 s	0.0093 / 0.0110	1.18 / 2.79 s	0.0112 / 0.0135	1.39 / 4.31 s

Fig. 7 and Table II show the results. TEASER++ gives similar errors under the two noise bounds, but takes a much longer time when the noise bound is smaller (0.07 m). Compared with TEASER++ using 0.25 m as the noise bound, although the maximum time costs of FuzzyPSReg-SS are greater for five out of the six pairs (excluding Pair 4), the average time costs of FuzzyPSReg-SS are smaller for all the pairs. In addition, the time costs of FuzzyPSReg-SS are much lower than that of TEASER++ using the more reasonable noise bound, 0.07 m. Also, FuzzyPSReg-SS achieves higher accuracy than TEASER++ does. We choose Pairs 4 and 5 of Fig. 8 as examples to give more details. As shown in Fig. 9, each scan contains a relatively large plane (wall or ceiling), and the planes of the two scans in each pair have similar areas and shapes but partially overlap each other. In this case, although other parts of the scans (such as the parts marked by green ellipses in Fig. 9) take much smaller portions than the planes do, they are key factors to determine a precise alignment. The registration metric of TEASER++ is the truncated point-to-point distances, formed based on the maximum clique of inliers found from the given correspondences. With this design, TEASER++ treats incorrect correspondences on the planes as inliers while excluding true correspondences on the small parts. As a result, it ignores the key factors and leaves a misalignment as shown in Fig. 9. By contrast, the GK-based metric (9) utilizes the adaptive distance norm, where different weights are applied to the distances in different directions. According to Section III-C, with the GK-based metric, the distance between two planes in their normal directions has a greater weight and leads to a greater distance loss when compared with other directions. Thus, for the small but key parts of the scans, the GK-based metric highlights their misalignment. From Fig. 9,  $\rho_{gk}(\lambda)$  of the transformation  $\lambda$  given by TEASER++ is much greater than that given by FuzzyPSReg-SS. These results indicate that the GK-based metric is a more accurate measure for point cloud alignment than the metric of TEASER++, and FuzzyPSReg-SS provides a more precise scan matching than TEASER++ does. Note that the scans of each pair in Fig. 9 partially overlap each other, but no trimming is applied in the registration. Thus,  $\rho_{gk}(\lambda)$  of FuzzyPSReg-SS may not be very close to 1, because the nonoverlapping part increases the distance loss.

In most cases of Fig. 7 and Table II, the time costs of FuzzyPSReg-SS applied to point clouds are slightly higher than that applied to corresponding points. The reason is that for each point cloud, its down-sampled points are more than its corresponding points. Then, a longer time is needed for the fuzzy clustering and fine registration. In this test, when

FuzzyPSReg-SS is applied to corresponding points, the average time for fuzzy clustering is about 0.43 s and the average time for fine registration is about 0.32 s; when FuzzyPSReg-SS is applied to down-sampled point clouds, the average time for fuzzy clustering is about 0.66 s and the average time for fine registration is about 0.33 s. The time for coarse registration (global search) varies with different initial poses.

It is claimed in [8] that TEASER++ can solve problems without correspondences. Thus, we use the corresponding points of Pairs 1 and 2 of Fig. 8 to perform the third test, where the order of the points in each moving set is randomly arranged. In this test, the ground truth alignments of the two pairs are used as their initial poses. For TEASER++, three different noise bounds, 0.25, 0.07, and 0.03 m, are tried. For FuzzyPSReg-SS, the parameter setting is the same as that in the previous tests. The results are shown in Fig. 10. Different from the second test, in this test, TEASER++ using a greater noise bound takes a longer time for registration. In addition, for both pairs, TEASER++ gives erroneous transformations regardless of the noise bound. By contrast, the results of FuzzyPSReg-SS are accurate and not affected by the order of points in the moving set, because FuzzyPSReg-SS does not rely on or need to know point correspondences.

In the previous tests, the point clouds in each pair partially overlap each other, and FuzzyPSReg-SS aligns them without trimming. To test the robustness of FuzzyPSReg-SS with respect to the trimming ratio, we apply FuzzyPSReg-SS using different trimming ratios to align two scan pairs selected from the challenging dataset [42], as shown in Fig. 11. Each of the scans contains hundreds of thousands of points. To improve efficiency, each scan is down-sampled to around 5000 points for the fuzzy clustering, where  $N_{C_F} = N_{C_M} = 100$ ; and each moving set is down-sampled to around 2500 points for the fine registration. The results in Fig. 11 show that FuzzyPSReg-SS gives accurate alignments for the two pairs when the trimming ratios are between [0, 0.38] and [0, 0.36], respectively. This test demonstrates that FuzzyPSReg-SS performs robustly when the trimming ratio varies in a relatively large interval.

### C. Tests of FuzzyPSReg-O2S

In this section, we test and compare FuzzyPSReg-O2S with TEASER++ [8] and another recently developed learning-based method: feature-metric registration (FMR) [17]. Six scenes and eight objects, selected from the YCB-Video dataset [23], are used for the tests, as shown in Figs. 12 and 13. In each case of the tests, the registration methods are applied to align the model

TABLE III  
 $N_r$  ROTATIONS ( $\mathbf{r}_i, i = 1, \dots, N_r$ ) USED BY EACH OBJECT IN THE MULTISTART LOCAL OPTIMIZATION OF FUZZYPSREG-O2S

Object	$\mathbf{r}_1$	$\mathbf{r}_2$	$\mathbf{r}_3$	$\mathbf{r}_4$
Power drill ( $N_r = 4$ )	(0,0,0)	( $\pi$ ,0,0)	(0, $\pi$ ,0)	(0,0, $\pi$ )
Scissors ( $N_r = 3$ )	(0,0,0)	( $\pi$ ,0,0)	(0, $\pi$ ,0)	—
Banana ( $N_r = 4$ )	(0,0,0)	( $\pi$ ,0,0)	(0, $\pi$ ,0)	(0,0, $\pi$ )
Large clamp ( $N_r = 2$ )	(0,0,0)	( $\pi$ ,0,0)	—	—
X-large clamp ( $N_r = 2$ )	(0,0,0)	(0, $\pi$ ,0)	—	—
Bowl ( $N_r = 2$ )	(0,0,0)	( $\pi$ ,0,0)	—	—
Mug ( $N_r = 3$ )	(0,0,0)	( $\pi$ /2,0,0)	(0, $\pi$ /2,0)	—
Mustard bottle ( $N_r = 2$ )	(0,0,0)	( $\pi$ ,0,0)	—	—

of an object with a scene scan containing this object. The objects are marked in the scene scans of Fig. 12. To test TEASER++, the FPFH feature descriptor [34] is used to establish putative correspondences between the object model and the scene scan. The noise bound of TEASER++ is set as 0.02 m. To test FMR, its model trained based on the ModelNet40 dataset [30] is utilized. For FuzzyPSReg-O2S, the  $N_r$  rotations used by each object in the multistart local optimization of the fine registration are given in Table III.

The three cases of Scenes 1 and 2 are employed for the first test, and the results are shown in Figs. 14–16. From Fig. 14, in each case, hundreds of point correspondences are built based on FPFH features, where the inlier ratios are at a very low level. TEASER++ does not align the object model with the scene scan. Besides, FMR, which is directly applied to the point clouds without needing to know the correspondences, also fails to register the point clouds.

By contrast, Figs. 15 and 16 show that FuzzyPSReg-O2S aligns the point clouds in each case. From Fig. 15, FuzzyPSReg-O2S correctly aligns the power drill with Scene 1 using only one round of the coarse-to-fine registration. The result of the coarse registration,  $\lambda^{(1)}$ , achieves  $\rho_{fcm}(\lambda^{(1)}) < 1$ . As shown in Fig. 15(B),  $\lambda^{(1)}$  finds an acceptable position of the power drill. Then, the part of the scene scan lying in the basic bounding box of the transformed power drill is segmented to perform the fine registration. In the fine registration shown in Fig. 15(C), the refined pose using  $\mathbf{r}_2 = (\pi, 0, 0)$  as the initialization gives the lowest distance loss, and thus, is chosen as  $\lambda^{(2)}$ . From Fig. 15(D), the power drill's model is well aligned with the segment after the fine registration. However, due to the fact that the segment includes a nonnegligible part of outliers (corresponding to the scissors),  $\rho_{gk}(\lambda^{(2)})$  is small than  $1 + \Delta\bar{J}_2$  but greater than  $1 + \Delta\bar{J}_1$ , and leads to  $q_{gk}(\lambda^{(2)}) = 0$  according to (22) in the extended registration quality assessment. As a result, the user is involved in checking and then makes a decision to run FuzzyPSReg-SS. Next, the enlarged bounding box of the transformed power drill is used to obtain a new segment from the scene scan, and FuzzyPSReg-SS is applied to further refine the alignment of the power drill's model and the new segment, where the trimming ratio  $\xi_3 = 0.35$  is roughly chosen by the user. Afterward, a more precise registration is obtained and is taken as the final result as shown in Fig. 15(E) and (F).

From Fig. 16, in each of the two cases, FuzzyPSReg-O2S takes three rounds of the coarse-to-fine registration. In the case shown in the first row of Fig. 16, the scissors' model is moved

to incorrect positions in Rounds 1 and 2. The quality assessment detects the incorrectness and gives  $q_{gk}(\lambda^{(2)}) = -1$ . Thus, the scene scan is pruned automatically to proceed with the next round of the coarse-to-fine registration. In Round 3, a rough position of the scissors is found in the coarse registration. Then, the multi-start local optimization in the fine registration using  $\mathbf{r}_3 = (0, \pi, 0)$  as the initialization gives a refined pose with the lowest distance loss, and gives  $q_{gk}(\lambda^{(2)}) = 0$ . Subsequently, the user checks the result and chooses to run FuzzyPSReg-SS with  $\xi_3 = 0.1$ . After applying FuzzyPSReg-SS, a more accurate alignment is obtained to be the final result. Note that same to the case shown in Fig. 15, in this case, FuzzyPSReg-SS uses local optimization alone for the registration since the two point clouds have been well aligned by the coarse-to-fine registration. In addition, in Round 2, part of the points corresponding to the scissors in the scene scan is pruned. Nevertheless, the following Round 3 together with FuzzyPSReg-SS still provides a correct alignment. In the case shown in the second row of Fig. 16, the registration process is similar to that in the first row, and the main differences are the following. 1) In Round 2, the result is  $q_{gk}(\lambda^{(2)}) = 0$ , and thus, the user is involved to check. Since the X-large clamp is aligned to the wrong place, the user chooses to prune the scene scan and perform a new round. 2) Round 3 gives an acceptable position but an incorrect orientation of the X-large clamp, and the incorrectness is out of the local convergence basin of the fuzzy cluster-based registration metrics. To correct the orientation, FuzzyPSReg-SS utilizes both of its global and local optimizations for the registration, and thus, takes a longer time compared to the previous two cases.

The four cases of Scenes 3 and 4 are employed for the second test, and the results are shown in Figs. 17 and 18. Similar to the first test, TEASER++ and FMR do not align the point clouds. From Fig. 17, in the cases shown in the first, third, and fourth rows, after one or several rounds of the coarse-to-fine registration, FuzzyPSReg-O2S utilizes FuzzyPSReg-SS to correct or further refine the pose. In the case shown in the second row, FuzzyPSReg-O2S takes two rounds of the coarse-to-fine registration to locate the object pose, and the quality assessment gives  $q_{gk}(\lambda^{(2)}) = 1$ . Thus, it does not apply FuzzyPSReg-SS. Note that in the cases shown in the second and fourth rows of Fig. 17, due to the symmetries of the objects (the mustard bottle and the X-large clamp), FuzzyPSReg-O2S gives the poses with rotation errors of about  $180^\circ$  compared to the ground truths. These results are regarded as correct alignments in this study. Therefore, FuzzyPSReg-O2S aligns the point clouds in every case of this test.

The four cases of Scenes 5 and 6 are employed for the third test, and the results are shown in Figs. 19 and 20. For all the cases, TEASER++ and FMR still do not provide accurate results. From Fig. 19, in the cases shown in the first, third, and last rows, FuzzyPSReg-O2S, respectively takes one, two, and four rounds of the coarse-to-fine registration to obtain a rough pose of the object, where  $q_{gk}(\lambda^{(2)}) = 0$ . Then, FuzzyPSReg-SS refines the rough poses. In the case shown in the second row, FuzzyPSReg-O2S finds the correct pose of the object in two rounds of the coarse-to-fine registration, and the result gives  $q_{gk}(\lambda^{(2)}) = 1$ . Thus, it does not apply FuzzyPSReg-SS. Note that in the case

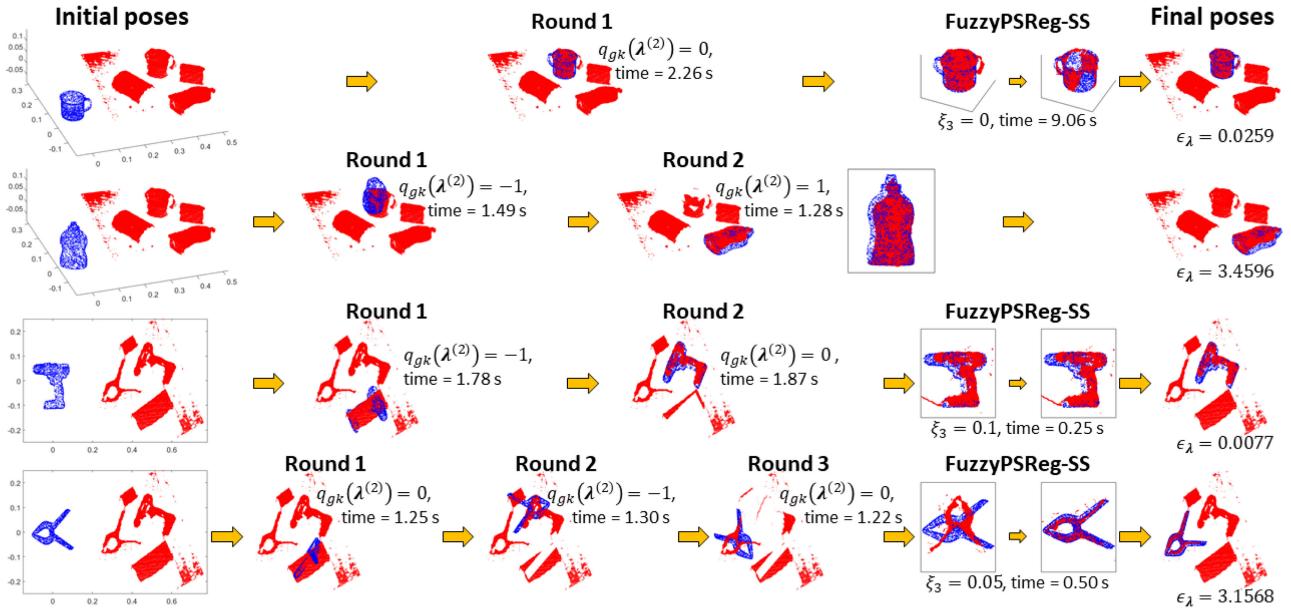


Fig. 17. Registration results of FuzzyPSReg-O2S for Scenes 3 and 4 of Fig. 12. Excluding the user's checking time, the time costs of the cases in rows 1–4 are 11.32, 2.77, 3.90, and 4.27 s, respectively. In the cases shown in rows 2 and 4, due to the symmetries of the objects (the mustard bottle and the X-large clamp), FuzzyPSReg-O2S gives the poses with rotation errors of approximately 180° compared to the ground truths. We consider them as correct alignments.

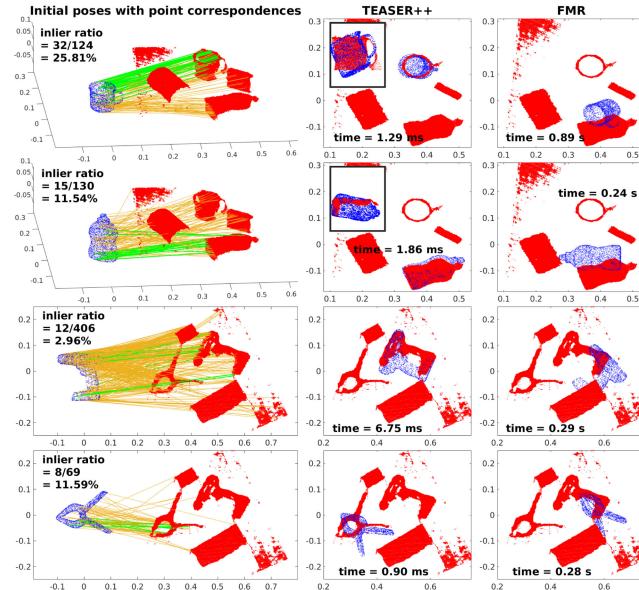


Fig. 18. Registration results of TEASER++ [8] and FMR [17] for the four cases in Scene 3 and Scene 4. Among the point correspondences derived by PFPFH features, the inliers are in green and the outliers are in orange. In the four cases, TEASER++ and FMR do not provide correct alignments.

shown in the third row of Fig. 19, due to the symmetry of the object (the bowl), FuzzyPSReg-O2S gives a pose with a rotation error compared to the ground truth. We also consider it as a correct alignment. Thus, FuzzyPSReg-O2S gives correct results in every case of this test.

Next, we conduct a multitrail test for FuzzyPSReg-O2S, using the following four cases selected from Fig. 12: Case 1. the scissors and Scene 1; Case 2. the X-large clamp and Scene 2; Case 3.

the mustard bottle and Scene 3; Case 4. the power drill and Scene 4. For each case, we perform 10 trials of FuzzyPSReg-O2S to align the point clouds, and the results are shown in Fig. 21. For Cases 1, 2, and 4, the user is involved in checking in every trial; while for Case 3, three trials do not include the user's checking. In some trials of Cases 2 and 4, FuzzyPSReg-SS is applied to correct the object pose using the global optimization, and thus, may need more time. In all the trials, FuzzyPSReg-O2S correctly aligns the point clouds.

In the above tests, compared with TEASER++ and FMR, FuzzyPSReg-O2S takes a longer time in calculation but provides accurate registration results. Also, it neither relies on correspondences nor needs a long training phase. Besides, it does not put a heavy burden on the user. In some cases, such as aligning the mustard bottle with Scene 3 and aligning the large clamp with Scene 5, FuzzyPSReg-O2S can locate the object pose without involving the user. On the other hand, FuzzyPSReg-O2S has some limitations, which also exist in many other methods using depth-only data. First, it does not deal with the symmetry of the object and may give a rotation error compared to the ground truth when the object is symmetrical in shape, such as in some cases shown in Figs. 17 and 19. Second, it may not handle an overlarge size difference between the point clouds. For example, in Scene 2 or 5 of Fig. 12, it is difficult for FuzzyPSReg-O2S to align the marker pen's model with the scene scan. Third, it may not work when the part of the scene scan corresponding to the object takes up a too-small portion of the object's surface. For example, when the object is severely occluded or the sensor does not capture enough depth information of the object, it is hard for FuzzyPSReg-O2S to locate the object pose. In future work, we plan to include the texture information of the point clouds in their registration to handle the limitations and improve the recognition capability.

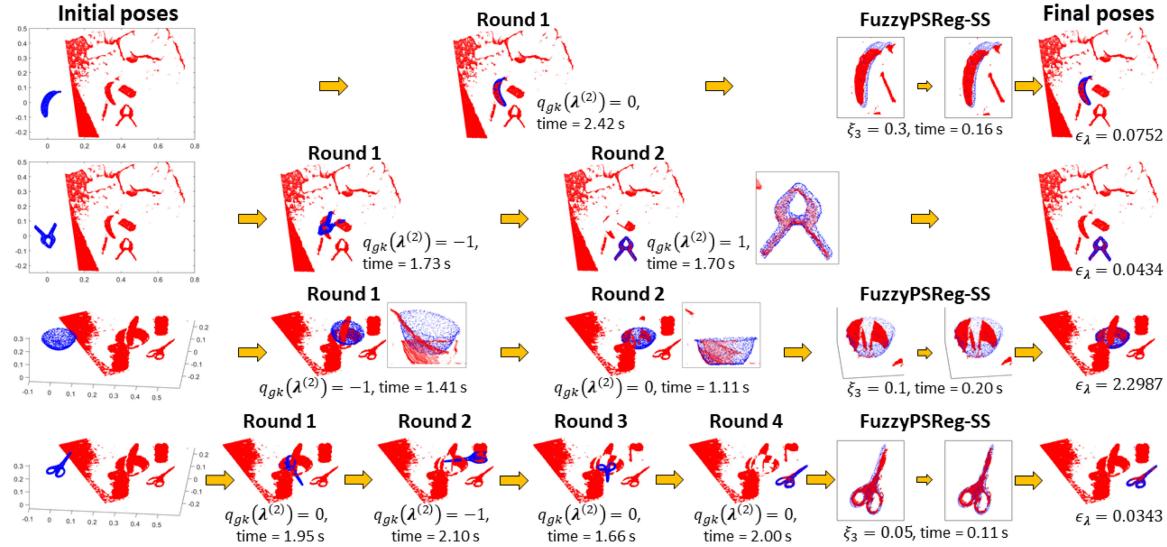


Fig. 19. Registration results of FuzzyPSReg-O2S for Scenes 5 and 6 of Fig. 12. Excluding the user's checking time, the time costs of the cases in rows 1–4 are 2.58, 3.43, 2.72, and 7.82 s, respectively. In the case shown in row 3, due to the symmetry of the object (the bowl), FuzzyPSReg-O2S gives a pose with a rotation error when compared to the ground truth. We consider it as a correct alignment.

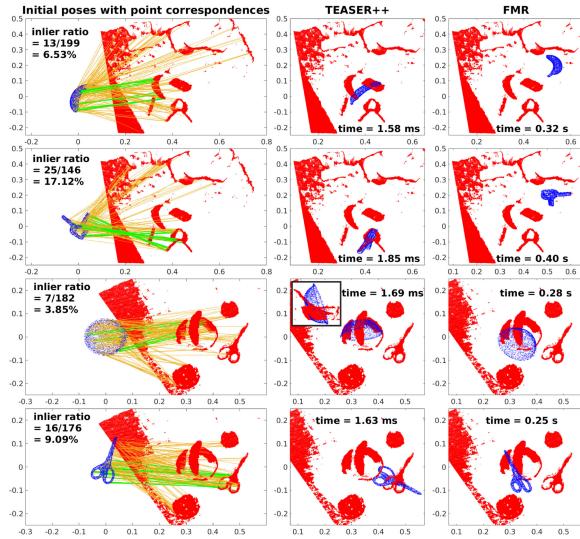


Fig. 20. Registration results of TEASER++ [8] and FMR [17] for the four cases in Scene 5 and Scene 6. Among the point correspondences derived by PFPFH features, the inliers are in green and the outliers are in orange. In the four cases, TEASER++ and FMR do not provide correct alignments.

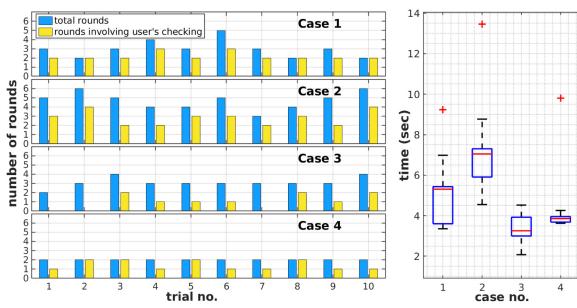
## VI. CONCLUSION

This article studied FuzzyPSReg. First, by extending the FCM-based registration metric of our previous work, we developed the GK-based metric. Then, we developed the registration quality assessment of the GK-based metric. Afterward, we presented the mathematical analysis and comparison of the two fuzzy cluster-based metrics. Next, two FuzzyPSReg strategies were proposed by effectively combining the two metrics. FuzzyPSReg-SS aligned similar-sized point clouds and makes a great improvement in computational efficiency when compared with the global registration method in our previous work. FuzzyPSReg-O2S aligned two point clouds with a relatively large size difference and could be used for object pose estimation. FuzzyPSReg-O2S involved human–computer collaboration and provided shared autonomy to complete tasks without putting a heavy burden on users. Different point clouds were used to test and compare the proposed method with state-of-the-art registration approaches. The experimental results demonstrated the advantages and effectiveness of our method.

## REFERENCES

- [1] P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 239–256, Feb. 1992.
- [2] D. Chetverikov, D. Stepanov, and P. Krsek, "Robust euclidean alignment of 3D point sets: The trimmed iterative closest point algorithm," *Image Vis. Comput.*, vol. 23, no. 3, pp. 299–309, 2005.
- [3] A. Segal, D. Haehnel, and S. Thrun, "Generalized-ICP," *Robot.: Sci. Syst.*, vol. 2, no. 4, 2009, Art. no. 435.
- [4] S. Bouaziz, A. Tagliasacchi, and M. Pauly, "Sparse iterative closest point," in *Proc. Comput. Graph. Forum*, 2013, pp. 113–123.
- [5] J. Zhang, Y. Yao, and B. Deng, "Fast and robust iterative closest point," *IEEE Trans. Pattern Anal. Mach. Intell.*, to be published, doi: 10.1109/TPAMI.2021.3054619.
- [6] C. Olsson, F. Kahl, and M. Oskarsson, "Branch-and-bound methods for euclidean registration problems," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 5, pp. 783–794, May 2009.

Fig. 21. Multitrial results of FuzzyPSReg-O2S for the four selected cases.

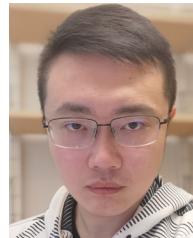


- [7] Q.-Y. Zhou, J. Park, and V. Koltun, "Fast global registration," in *Proc. Eur. Conf. Comp. Vis.*, 2016, pp. 766–782.
- [8] H. Yang, J. Shi, and L. Carlone, "Teaser: Fast and certifiable point cloud registration," *IEEE Trans. Robot.*, vol. 37, no. 2, pp. 314–333, Apr. 2021.
- [9] M. Magnusson, A. Lilienthal, and T. Duckett, "Scan registration for autonomous mining vehicles using 3D-NDT," *J. Field Robot.*, vol. 24, no. 10, pp. 803–827, 2007.
- [10] A. Myronenko and X. Song, "Point set registration: Coherent point drift," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 12, pp. 2262–2275, Dec. 2010.
- [11] B. Jian and B. C. Vemuri, "Robust point set registration using Gaussian mixture models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1633–1645, Aug. 2011.
- [12] 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," *Int. J. Robot. Res.*, vol. 31, no. 12, pp. 1377–1393, 2012.
- [13] G. D. Evangelidis and R. Horaud, "Joint alignment of multiple point sets with batch and incremental expectation-maximization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 6, pp. 1397–1410, Jun. 2018.
- [14] Y. Aoki, H. Goforth, R. A. Srivatsan, and S. Lucey, "Pointnetlk: Robust & efficient point cloud registration using pointnet," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 7163–7172.
- [15] Z. Gojcic, C. Zhou, J. D. Wegner, and A. Wieser, "The perfect match: 3D point cloud matching with smoothed densities," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 5545–5554.
- [16] 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 *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 7193–7203.
- [17] X. Huang, G. Mei, and J. Zhang, "Feature-metric registration: A fast semi-supervised approach for robust point cloud registration without correspondences," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11366–11374.
- [18] Q. Liao, D. Sun, and H. Andreasson, "Point set registration for 3D range scans using fuzzy cluster-based metric and efficient global optimization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 9, pp. 3229–3246, Sep. 2021.
- [19] L. Silva, O. R. P. Bellon, and K. L. Boyer, "Precision range image registration using a robust surface interpenetration measure and enhanced genetic algorithms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 5, pp. 762–776, May 2005.
- [20] R. Sandhu, S. Dambreville, and A. Tannenbaum, "Point set registration via particle filtering and stochastic dynamics," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 8, pp. 1459–1473, Aug. 2010.
- [21] J. Yang, H. Li, D. Campbell, and Y. Jia, "Go-icp: A globally optimal solution to 3D ICP point-set registration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 11, pp. 2241–2254, Nov. 2016.
- [22] A. W. Fitzgibbon, "Robust registration of 2D and 3D point sets," *Image Vis. Comput.*, vol. 21, no. 13/14, pp. 1145–1153, 2003.
- [23] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes," in *Proc. Robot. Sci. Syst.*, 2018, *arXiv: 1711.00199*.
- [24] D. Xu, D. Anguelov, and A. Jain, "Pointfusion: Deep sensor fusion for 3D bounding box estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 244–253.
- [25] C. Wang *et al.*, "Densefusion: 6D object pose estimation by iterative dense fusion," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3343–3352.
- [26] A. Collet, D. Berenson, S. S. Srinivasa, and D. Ferguson, "Object recognition and full pose registration from a single image for robotic manipulation," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2009, pp. 48–55.
- [27] M. Blum, J. T. Springenberg, J. Wülfing, and M. Riedmiller, "A learned feature descriptor for object recognition in RGB-D data," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2012, pp. 1298–1303.
- [28] M. Zhu *et al.*, "Single image 3D object detection and pose estimation for grasping," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2014, pp. 3936–3943.
- [29] H. Dong, E. Asadi, G. Sun, D. K. Prasad, and I.-M. Chen, "Real-time robotic manipulation of cylindrical objects in dynamic scenarios through elliptic shape primitives," *IEEE Trans. Robot.*, vol. 35, no. 1, pp. 95–113, Feb. 2019.
- [30] Z. Wu *et al.*, "3D shapenets: A deep representation for volumetric shapes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1912–1920.
- [31] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 652–660.
- [32] O. J. Woodford, M.-T. Pham, A. Maki, F. Perbet, and B. Stenger, "Demistifying the hough transform for 3D shape recognition and registration," *Int. J. Compt. Vis.*, vol. 106, no. 3, pp. 332–341, 2014.
- [33] A. E. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3D scenes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 5, pp. 433–449, May 1999.
- [34] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (FPFH) for 3D registration," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2009, pp. 3212–3217.
- [35] C. Zach, A. Penate-Sanchez, and M.-T. Pham, "A dynamic programming approach for fast and robust object pose recognition from range images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 196–203.
- [36] J. C. Bezdek, R. Ehrlich, and W. Full, "FCM: The fuzzy c-means clustering algorithm," *Comput. Geosci.*, vol. 10, no. 2/3, pp. 191–203, 1984.
- [37] D. E. Gustafson and W. C. Kessel, "Fuzzy clustering with a fuzzy covariance matrix," in *Proc. IEEE Conf. Decis. Control*, 1979, pp. 761–766.
- [38] S. Granger and X. Pennec, "Multi-scale EM-ICP: A fast and robust approach for surface registration," in *Proc. Eur. Conf. Comp. Vis.*, 2002, pp. 418–432.
- [39] A. L. Pavlov, G. W. Ovchinnikov, D. Y. Derbyshev, D. Tsetserukou, and I. V. Oseledets, "AA-ICP: Iterative closest point with anderson acceleration," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2018, pp. 3407–3412.
- [40] C. Charalambous and J. Bandler, "Non-linear minimax optimization as a sequence of least p th optimization with finite values of p," *Int. J. Syst. Sci.*, vol. 7, no. 4, pp. 377–391, 1976.
- [41] StanfordComputerGraphicsLaboratory, "The stanford 3D scanning repository," 1996. [Online]. Available: <http://graphics.stanford.edu/data/3Dscanrep/>
- [42] F. Pomerleau, M. Liu, F. Colas, and R. Siegwart, "Challenging data sets for point cloud registration algorithms," *Int. J. Robot. Res.*, vol. 31, no. 14, pp. 1705–1711, 2012.
- [43] J. Xiao, A. Owens, and A. Torralba, "Sun3D: A database of big spaces reconstructed using SFM and object labels," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 1625–1632.



**Qianfang Liao** received the B.Eng. degree in automation from Wuhan University, Wuhan, China, in 2006, the M.Eng. degree in automation from Shanghai Jiao Tong University, Shanghai, China, in 2009, and the Ph.D. degree in electrical engineering from the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, in 2015.

Currently, she is a Researcher with Örebro University, Örebro, Sweden. Her research interests include fuzzy modeling, computer vision, and control theory.



**Da Sun** received the B.Eng. and Ph.D. degrees in mechatronics from the University of Wollongong, Wollongong, Australia, in 2012 and 2016, respectively.

From 2016 to 2017, he was a Research Fellow with the National University of Singapore, Singapore. He is currently a Tenured Researcher with Örebro University, Örebro, Sweden. His research interests include robot learning and control, and human–machine interaction.



**Henrik Andreasson** received the master's degree in mechatronics from the Royal Institute of Technology, Stockholm, Sweden, in 2001 and the Ph.D. degree in computer science from Örebro University, Örebro, Sweden, in 2008.

He is currently an Associate Professor with the Center for Applied Autonomous Sensor Systems (AASS), Örebro University. His research interests include mobile robotics, computer vision, and machine learning.