3DMNDT: 3D Multi-View Registration Method Based on the Normal Distributions Transform

Jihua Zhu, Member, IEEE, Jiaxi Mu, Chao-Bo Yan, Senior Member, IEEE, Di Wang, and Zhongyu Li

Abstract—The normal distributions transform (NDT) is an effective paradigm for point set registration. This method was initially designed for pair-wise registration and suffers from the accumulated error problem when directly applied to multi-view registration. Under the framework of point-to-cluster correspondence, this paper proposes a novel multi-view registration method named 3D multi-view registration based on the normal distributions transform (3DMNDT), which integrates the k-means clustering and Lie algebra optimizer to achieve multi-view registration. More specifically, the multi-view registration is cast into the maximum likelihood estimation problem. Firstly, k-means clustering is utilized to divide all data points into different clusters, where one normal distribution is computed to locally model the probability of measuring a data point in each cluster. Subsequently, the multi-view registration problem is formulated by the NDT-based likelihood function. To maximize this likelihood function, the Lie algebra optimizer is introduced and developed to optimize each rigid transformation sequentially. 3DMNDT implements data point clustering, NDT computing, and rigid transformation optimization alternately until the desired registration results are obtained. Experimental results tested on benchmark data sets illustrate that 3DMNDT can achieve state-of-the-art performance for multi-view registration.

Note to Practitioners—This paper is motivated by solving the problem of registering multiple point sets. The normal distributions transform (NDT) is a well-known pair-wise registration method widely applied in the robotic domain. This paper extends the original NDT and proposes a novel registration method to simultaneously align more than two point sets. The multi-view registration is cast into the maximum likelihood estimation problem. Subsequently, the k-means clustering and Lie algebra optimizer are integrated to estimate registration parameters. Experimental results demonstrate its superior performance on the accuracy, efficiency, and robustness for multi-view registration of point sets.

Index Terms—Multi-view registration, normal distributions transform, k-means clustering, Lie algebra optimizer.

I. INTRODUCTION

POINT set registration is a fundamental problem arising in main domains, such as computer vision [1], [2], [3], robotics [4], [5], and medical image analysis [6], [7]. With the development of scanning technology, many scan devices can acquire accurate scan data from a scene or object. However, it is difficult to scan the whole object or scene simultaneously due to the limited field of view. For many applications, such as 3D model reconstruction and robot mapping, it is necessary to acquire point data from different viewpoints and transform these point sets from a set-centered coordinate frame into a unified coordinate frame, which raises the point set registration problem.

In the literature, Iterative Closest Point (ICP) [9] is one of the most popular solutions to point set registration. Given initial rigid transformation, this method alternatively establishes point correspondences and optimizes rigid transformation to achieve pair-wise registration with reasonable accuracy. However, the original ICP method can not register partially overlapping point sets, and it is time-consuming due to the establishment of point-to-point correspondences. Then, the normal distributions transform (NDT) [10], [11] was proposed to solve the pair-wise registration problem. By utilizing standard division or k-means clustering [12], target points are divided into clusters (voxels), where points in each cluster are approximated by a normal distribution before registration. To achieve registration, it alternately operates the establishment of point-to-cluster correspondence and optimization of rigid transformation by the Newton optimizer. Since target points in each cluster are approximated and replaced by one NDT in advance, this method is efficient for aligning large-scale data sets. What’s more, it can align partially overlapping point sets. Therefore, it is popular in computer vision and robotics.

Although the NDT method has been taken as the prerequisite for some multi-view registration methods based on energy function [13], it has not been directly extended to solve the...
multi-view registration problem. In multi-view registration, point set poses keep changing with the update of rigid transformations, and data points cannot be divided into fixed NDT in advance. Therefore, it is difficult to formulate the multi-view registration problem directly and simultaneously optimize a lot of rigid transformations under the original NDT framework. Besides, the original NDT divides the space into axis-aligned voxels. The standard division leads to a significant variance of point number in each voxel and results in invalid voxels containing a small number of data points. Therefore, the standard division inevitably reduces the registration performance due to discarding these invalid voxels. Moreover, the original NDT utilizes the Jacobian and Hessian matrices in the Newton solver to optimize the rigid transformation. In the 3D case, these two matrices contain 8 and 15 complex terms required to be computed for each data point, which is inconvenient and tedious. This paper extends the original NDT and proposes a novel method named 3DMNDT for multi-view registration.

The contributions of this paper are delivered as follows: 1) We directly formulate the multi-view registration problem into an NDT-based likelihood function, which takes the initial alignment as inputs and requires updating point-to-cluster correspondences for optimization. This method is different from these energy functions, which take pair-wise registration results as their inputs and do not update point correspondences during the optimization of rigid transformations. 2) We integrate k-means clustering and the Lie algebra optimizer to maximize the NDT-based likelihood function until convergence. Given initial alignment, 3DMNDT alternately operates point clustering and rigid transformation optimization to achieve multi-view registration. This method differs from previous NDT variants, which cluster all target points into different clusters before registration and then optimize rigid transformation without clustering update. The Lie algebra optimizer utilizes the Lie optimization theory to maximize the likelihood function. As the substitute for the Newton optimizer, this optimizer can directly implement optimization on the rotation matrix with simple implementation and obtain promising registration results.

The remainder of this paper is organized as follows. Section 2 briefly discusses related methods of point set registration. In Section 3, we formulate the multi-view registration problem under the NDT framework and then derive 3DMNDT to solve 3D multi-view registration. We also analyze its computation complexity and present its implementation details in Section 4. Section 5 displays experimental results on benchmark data sets. Finally, we conclude this paper in Section 6.

II. RELATED WORKS

This section only discusses these methods related to the proposed method for multi-view registration. For convenience, the terms motion and rigid transformation are utilized interchangeably throughout this paper. Specifically, one relative motion denotes the rigid transformation between two set-centered frames of point sets, and one global motion indicates the rigid transformation between one set-centered frame to the global coordinate frame.

A. Pair-Wise Registration

According to the number of point sets being registered, point set registration is divided into pair-wise and multi-view registration problems. For pair-wise registration, many variants have been derived from the original ICP algorithm, and they improve its performance from different aspects [14], [15].

For partially overlapping point sets, the Trimmed ICP (TrICP) algorithm [16] introduces the overlap parameter to distinguish overlapping regions automatically and can achieve good registration. Besides, Peter and Wolfgang [10] proposed the NDT algorithm for registering 2D laser scans. Magnusson et al. [11] extended the original NDT algorithm to 3D cases, where the computation of one Jacobian and Hessian matrices contains 8 and 15 complex terms for each data point. Das and Waslander [12] replaced standard division with k-means clustering to divide all data points of the target point set into different clusters, where the target point set is always static during registration. But this method cannot be directly extended to multi-view registration, where rigid transformations of point sets keep changing, and all data points should be dynamically clustered at each iteration. Some other NDT variants were also proposed to solve the pair-wise registration problem, and their performances have been evaluated in [17].

For accurate registration, Myronenko and Song proposed the Coherent Point Drift (CPD) algorithm [18], which considers pair-wise registration as a probability density estimation problem. It takes the source point set as the Gaussian mixture model (GMM) centroids and fits them to the target point set by maximizing the likelihood. Besides, Jian et al. [19] formulated the pair-wise registration problem as aligning two GMMs, where one GMM represents each point set. Then the GMM-REG method is proposed to achieve pair-wise registration by minimizing the $L_2$ measure between two GMMs. Like CPD, Gao and Tedrake [20] proposed the FilterReg, which takes the target point set as GMM centroids and fits the source point set to GMM by maximizing the likelihood. Although these GMM-based methods are powerful for rigid and non-rigid registration, they require building all point correspondences...
between two point sets being registered, leading to high computational complexity. Since these methods are locally convergent, they require to be provided with suitable initial parameters. Therefore, Yang et al. [2] utilized the box-and-ball (BnB) algorithm and proposed the first globally optimal method named Go-ICP for pair-wise registration. Without any initialization, this method can obtain promising registration of two 3D point sets with high overlap percentages. But this method is inefficient due to the search of the entire 3D motion space. One should provide initial rigid transformation to local convergent registration algorithms for registration efficiency. To estimate initial parameters, Sipiran and Bustos [21] proposed the 3D Harris detection method, which can detect the interesting point from the 3D point set represented by meshes. Given detected interesting points, the feature descriptor and matching method [22] was proposed to provide initial alignment for pair-wise registration.

Some deep learning methods have been proposed to achieve global pair-wise registration. Aoki [23] proposed the PointNetLK method, which integrated PointNet and Lucas & Kanade (LK) algorithm into a single trainable recurrent deep neural network to achieve pair-wise registration. Wang and Solomon [24] proposed the DCP method, which combines the embedding network and the transformer module with a differentiable SVD layer to achieve pair-wise registration. Given a set of 3D point correspondences, Pais et al. [25] proposed 3DRegNet method, which can classify point correspondences into inliers or outliers and regress parameters for pair-wise registration. Choy [26] proposed a deep global registration method, which utilizes a 6D convolutional network, a differentiable Weighted Procrustes algorithm, and a gradient-based optimizer to achieve correspondence confidence prediction, closed-form pose estimation, and pose refinement, respectively. For point-set pairs with low overlap, Huang et al. [27] proposed PREDATOR, which uses an overlap-attention module to exchange information between point sets to infer the salient points that lie in their overlap region. Although these methods are effective, they are supervised learning methods requiring many data sets to train the registration model. Besides, extending them to solve the multi-view registration problem is complex.

**B. Multi-View Registration**

Usually, many pair-wise registration methods are the basis of multi-view registration methods. Compared with the problem of pair-wise registration, the multi-view registration problem receives less attention due to its complexity. For multi-view registration, the primary method [28] is to incrementally register and integrate point set pairs until all point sets are integrated into one model. Although this method is easy to operate, it suffers from the accumulation error problem with the increase of integrated point sets. Then, Bergevin et al. [29] proposed the first method for multi-view registration. It sequentially optimizes the rigid transformation over the point correspondences, which are built between one point set and each other point set. This method overcomes the problem of cumulative error, but it neglects non-overlapping regions between each point set pair, which reduces the registration accuracy. To address this issue, Zhu et al. [30] proposed the coarse-to-fine registration method, which sequentially optimizes the trimmed errors of point correspondences built between one point set and other aligned point sets. With the increase of point sets, the coarse-to-fine registration method tends to be trapped into the local minimum. Accordingly, Tang et al. [31] proposed a hierarchical multi-view registration method, which continuously implements multi-view registration on a small number of point sets and integrates them as the one in the subsequent implementation. A similar approach is also proposed in [32] to register multi-view forest point clouds.

As many rigid transformations are involved in multi-view registration, most methods utilize an alternating optimization strategy to optimize each rigid transformation sequentially. However, some methods use batch optimization strategy. Given point correspondences established between point set pairs, Krishnan et al. [33] proposed a multi-view registration method via optimization-on-a-manifold, which can simultaneously optimize all rigid transformations. However, it isn’t easy to obtain point correspondences, which should be provided in advance. Toldo et al. [34] proposed the multi-view registration method by embedding the Generalized Procrustes Analysis into an ICP framework, which requires finding a mutually nearest neighbor. Although the mutually nearest neighbor is well defined, finding mutually nearest neighbors between each pair of point sets is time-consuming. Further, Mateo et al. [35] treated pairwise correspondences as missing data and formulated multi-view registration as the maximum likelihood estimation, where all rigid transformations are optimized by the Expectation–Maximization (EM) algorithm in batch mode. Since there are massive point correspondences, this method requires estimating huge hidden variables and is time-consuming.

Since multi-view registration is more complex than pair-wise registration, one feasible method is to recover the global motions from relative motions, where the term motion denotes rigid transformation. Accordingly, Peter and Wolfgang [13] proposed the nScan-matching method based on the energy function built upon existing pair-wise registration results. The energy function is optimized to recover multi-view registration results without updating point correspondences. Further, Govindu et al. [36] proposed the motion averaging and applied it to multi-view registration. This method utilizes the Lie group structure of motions [37] to implement the averaging of all available relative motions. Theoretically, accurate global motions can be obtained from several relative motions. However, this method is sensitive to outliers, where even one outlier will lead to the failure of multi-view registration. Further, Arrigoni et al. [38], [39] cast multi-view registration to a low-rank and sparse (LRS) matrix decomposition problem, where registration results are recovered from the low-rank matrix. This method requires concatenating all available relative motions and some zero matrices into one large matrix, which is then decomposed into a low-rank matrix and a sparse matrix. It is robust to outliers but tends to failure when the
equally treat all relative motions, which may possess different reliabilities. Accordingly, Guo et al. [40] proposed weighted motion averaging algorithm, and Jin et al. [41] proposed the weighted LRS for multi-view registration. Besides, Zhu et al [42] proposed the robust motion averaging algorithm under maximum correntropy criterion for robust registration of multi-view scans. Since these methods pay more attention to these relative motions with high reliability, they are more likely to obtain desired registration results than their original methods.

Georgios et al. [8] cast multi-view registration to the clustering problem and proposed JRMPc into solve this particular problem. This method assumes that all data points are generated from a central GMM and utilizes the EM algorithm to estimate the GMM components and rigid transformations. Under the same framework, Ravikumar [7] replaced the GMM with the student’s t-mixture model (TMM), which leads to more robust registration. Then Min et al. [43], [44] extended this method to align high-dimensional point sets by considering isotropic and anisotropic uncertainties, respectively. These two methods can jointly register multiple generalized point sets, where high-dimensional points consist of both positional and normal vectors. Besides, Peng et al. [45] extent GMMREG method for multi-view registration, which sequential estimates the rigid transformation by minimizing the L2 distance between probability distributions of the integrated model and each scan. Since these methods require estimating massive model parameters related to all data points, it is time-consuming and likely to be trapped into the local minimum.

To address these issues, Zhu et al. [46] extended the k-means clustering to solve the multi-view registration problem. Compared with JRMPc, this method is efficient and more likely to obtain desired registration results. However, it utilizes one cluster centroid to approximate all data points in the same cluster, which inevitably leads to much information loss and reduces registration accuracy. Further, Guo et al. [47] utilized the hierarchical k-means clustering algorithm to achieve multi-view clustering, improving registration accuracy. Besides, Zhu et al. [48] proposed an efficient GMM-based method under the perspective of EM. This method assumes that each data point is drawn from its corresponding GMM. Its nearest neighbors in other point sets are taken as Gaussian models and each scan. Since these methods require estimating quite a few rigid transformations and one covariance, it is efficient and time-economical. To address the robustness issue, Ma et al. [49] utilized the TMM to replace GMM, which renders the EM-based registration algorithm inherently robust to outliers and heavy-tail noise.

Recently, Gojcic et al. [50] proposed the first end-to-end algorithm to solve the multi-view registration problem by joint learning initial pair-wise alignment and globally consistent refinement. Yew and Lee [51] utilized the graph neural networks (GNNs) to learn transformation synchronization, which can recover multi-view registration results from a set of pair-wise registration results. Given enough training data, these methods may achieve global registration of multiple point sets without initialization. However, there may be a lack of training data in practical applications. Therefore, this paper proposes an unsupervised method for multi-view registration. It starts with the definition of the NDT-based likelihood function, which is then maximized by integrating the k-means clustering and the Lie algebra optimizer.

III. PROBLEM FORMULATION

Let $V_i = \{v_{i,1}, \ldots, v_{i,j}, \ldots, v_{i,N_i}\}$ denotes $N_i$ data points in the $i$-th point set and $V = \{V_i\}_{i=1}^M$ indicates $M$ point sets being registered. Similar to previous methods [8], [46], we regard multi-view registration as the clustering problem. Given the reference frame, the goal of multi-view registration is to simultaneously divide all data points into $K$ clusters and estimate $M$ rigid transformations. Specifically, the rigid transformation consists of 3D rotation $R_i \in SO(3)$ and translation vector $t_i = [t_{ix}, t_{iy}, t_{iz}] \in \mathbb{R}^3$, where the symbol SO(3) indicates Special Orthogonal group. The elements of $R$ are defined as:

$$R = \begin{bmatrix}
  c_3c_2 & -c_3s_2 & s_3 \\
  c_2s_1 - s_2c_3c_1 & c_1c_3 + s_2s_3c_1 & -c_1s_3c_2 + s_2c_3 \\
  s_2c_1 + c_2s_3c_1 & -c_1s_3s_2 + s_2c_3 & c_1c_3c_2 + s_2s_3 \\
\end{bmatrix},$$

(1)

where $c_i = \cos \theta_i$ and $s_i = \sin \theta_i$. For convenience, the rigid transformation is concatenated into Euclidean motion as:

$$T_i = \begin{bmatrix} R_i & t_i \\ 0 & 1 \end{bmatrix},$$

(2)

where $T_i \in SE(3)$ and the symbol SE(3) denotes the Special Euclidean group. Besides, we define the group action of SE(3) on a 3D vector $v_{i,j}$ as $T_i v_{i,j} = R_i v_{i,j} + t_i$. For each cluster, it is convenient to compute the NDT $(\mu, \Sigma)$, where $\mu$ and $\Sigma$ denote the cluster centroid and information (inverse of covariance) matrix, respectively.

Suppose the aligned data point $T_i v_{i,j}$ belongs to the $c_{i(j)}$-th cluster, then it is reasonable to assume that the aligned data point $T_i v_{i,j}$ is drawn from the normal distribution $T_i v_{i,j} \sim N(\mu_{c_{i(j)}}, \Omega_{c_{i(j)}}^{-1})$, where $c_{i(j)}$ denotes the index of a cluster. For the data point $v_{i,j}$, the probability function is defined as:

$$p(v_{i,j}) = N(T_i v_{i,j}; \mu_{c_{i(j)}}, \Omega_{c_{i(j)}}^{-1}),$$

(3)

Further, all data points are assumed to be independent of each other, and their joint probability distribution is formulated as follows:

$$p(V) = \prod_{i=1}^M \prod_{j=1}^{N_i} N(T_i v_{i,j}; \mu_{c_{i(j)}}, \Omega_{c_{i(j)}}^{-1}),$$

(4)

where the normal distribution on $\mathbb{R}^3$ is defined as follows:

$$N(v; \mu, \Sigma^{-1}) = \frac{1}{\sqrt{(2\pi)^3}} \exp(-\frac{1}{2}(v - \mu)^\top \Sigma^{-1} (v - \mu)).$$

This joint probability distribution is utilized to define the logarithm of the likelihood function as:

$$L(\Theta) = \log p(V) = \sum_{i=1}^M \sum_{j=1}^{N_i} \log N(T_i v_{c_{i(j)}}, \mu_{c_{i(j)}}, \Omega_{c_{i(j)}}^{-1}),$$

(5)
aligned data points into different clusters and then computes the NDT consisting of one cluster centroid \( \mu_k \) and information matrix \( \Omega_k \). Further, it utilizes the Lie algebra optimizer to sequentially optimize each rigid transformation, including the 3D rotation \( R_i \) and translation vector \( t_i \). Desired registration results are obtained by alternately implementing data point clustering, NDT computation, and transformation optimization on all point sets.

(a) Initial models, where different point sets are denoted in different colors. (b) Update cluster centroids \( (\mu_k^i, \Omega_k^i) \) for clusters containing more than 5 points, and other clusters are regarded as invalid. (c) Compute NDTs \( \{ (\mu_k^i, \Omega_k^i) \}_{k=1}^K \), where an ellipse indicates the NDT of one cluster. (d) Optimize rigid transformations \( \{(R_i^i, t_i^i)\}_{i=1}^M \), where a solid circle denotes one point set. (e) Refined model.

where \( \Theta \triangleq \left\{ (\mathcal{T}_i^M, (\mu_k^i)^K) \right\} \) denotes model parameters. It should be noted that \( (\Omega_k^i)_{k=1}^K \) is the by-product of clustering, and they do not require to be optimized.

Accordingly, the multi-view registration problem is transformed into the maximum likelihood estimation as follows:

\[
\max_{\Theta} L(\Theta) \\
\text{s.t. } \mathcal{T}_i \in \text{SE}(3), i = 1, 2, \cdots, M. \tag{6}
\]

We can achieve multi-view registration with data point clustering by optimizing Eq. (6).

IV. 3DMNDT METHOD

For the optimization of Eq. (6), we propose the 3D multi-view version of the NDT method (3DMNDT) displayed in Fig. 2. As shown in Fig. 2, this method optimizes Eq. (6) by iterations. Given initial rigid transformations \( \{ (\mathcal{T}_i^M, (\mu_k^i)^K) \}_{i=1}^M \) and initial cluster centroids \( \{ (\mu_k^i)^K \}_{k=1}^K \), three steps are alternately operated in each iteration.

1. Build point-to-cluster correspondence.

\[
c^i(j) = \arg \min_{k=1,2,\cdots,K} \| \mathcal{T}_i^i \mathbf{v}_{i,j} - \mu_k^{i-1} \|_2. \tag{7}
\]

Eq. (7) denotes the nearest neighbor search problem, which is usually solved by the k-d tree-based method. Specifically, the k-d tree is built on all cluster centroids \( \{ (\mu_k^{i-1})_k \}_{k=1}^K \). For each aligned data point \( \mathcal{T}_i^i \mathbf{v}_{i,j} \), the nearest neighbor \( \mu_k^{i-1} \) can be efficiently searched by k-d tree search algorithm [52].

2. Compute the NDT for each cluster.

\[
\mu_k^i = \frac{\sum_{i=1}^{M} \sum_{j=1}^{N_i} c^i(j) = k} {\sum_{i=1}^{M} \sum_{j=1}^{N_i} c^i(j) = k} \mathcal{T}^{-1}_i \mathbf{v}_{i,j}, \tag{8}
\]

\[
\Omega_k^i = (\Sigma_k^i + \varepsilon \mathbf{I})^{-1}, \tag{9}
\]

where \( \varepsilon = 10^{-6} \) prevents the occurrence of zero denominator and \( \Sigma_k^i \) indicates the covariance matrix calculated by:

\[
\Sigma_k^i = \frac{\sum_{i=1}^{M} \sum_{j=1}^{N_i} c^i(j) = k} {\sum_{i=1}^{M} \sum_{j=1}^{N_i} c^i(j) = k} (\Delta \mathbf{v}_{i,j}^i (\Delta \mathbf{v}_{i,j}^i)^\top), \tag{10}
\]

where \( \Delta \mathbf{v}_{i,j}^i = \mathcal{T}_i^{-1} \mathbf{v}_{i,j} - \mu_k^i \).

Step (2) is a trivial calculation problem. It should be noted that some clusters may only contain a small number of data points. In 3D cases, points in one cluster may be perfectly co-planar, and the covariance matrix will always be singular and cannot be inverted. For this reason, NDTs only computed for clusters containing more than 5 points, and other clusters are regarded as invalid.

3. Rigid transformation optimization.

If one data point belongs to an invalid cluster, it should be neglected in the optimization of rigid transformation. Accordingly, all rigid transformations can be optimized by maximum likelihood estimation:

\[
\max_{\Theta} \sum_{i=1}^{M} \sum_{j=1}^{N_i} \log \mathcal{N}(\mathbf{v}_{i,j}; \mu_{c^i(j),i}, \Omega_{c^i(j),i}^{-1}), \tag{11}
\]

where \( N_i \leq N_i \) denotes the number of valid data points in the \( i \)-th point set.

To obtain desired results, 3DMNDT iteratively operates these three steps until the likelihood \( L(\Theta) \) has no significant change or the iteration number \( s \) reaches the maximum value \( S \). Previously, the Newton optimizer is usually utilized to optimize rigid transformation in Step (3). It requires computing the Jacobian and Hessian matrices, which contain 23 complex terms for each data point. Accordingly, a more efficient solution is needed.

A. The Lie Algebra Optimizer

Both the Special Orthogonal group and the Special Euclidean group belong to the Lie group. According to [36], the Lie group SO(3) (or SE(3)) has a smooth differentiable structure and can be fully described by the tangent space. The tangent space to the manifold is called Lie algebra, which is denoted by so(3) (or se(3)). Based on Lie optimization theory [53], we develop the Lie algebra optimizer to optimize the rotation matrix in the likelihood function Eq.(11) directly with simple implementation.
It seems difficult to simultaneously optimize all rigid transformations involved in Eq. (6). Given clustering results, each rigid transformation can be separately optimized by the following objective function:

$$\max_{T \in SE(3)} \sum_{j=1}^{N_j} \log \mathcal{N}(T v_{i,j}; \mu_{c(i,j)}, \Omega_{c(i,j)})^{-1},$$

where \(i = 1, 2, \cdots, M\). For simplicity, we discard the iteration number \(s\), the point set number \(i\), and cluster number \(c(i,j)\) in all variables. Accordingly, Eq. (12) can be simplified as a non-linear least-squares problem:

$$\max_{T \in SE(3)} L(T) = \max_{T \in SE(3)} \sum_{j=1}^{N_j} \log \mathcal{N}(T v_{j}; \mu_j, \Omega_j^{-1})^{-1} = \min_{T \in SE(3)} \sum_{j=1}^{N_j} r_j^\top \Omega_j r_j,$$

where \(r_j = T v_j - \mu_j\) indicates the residual error between data point \(T v_j\) and its cluster centroid \(\mu_j\). For the optimization of Eq. (13), we turn to utilize retraction technique [54].

In the retraction technique, the vector space of \(se(3)\) is decomposed into rotational and translational components, i.e., \(\xi = [\xi_R \in \mathbb{R}^3, \xi_t \in \mathbb{R}^3]\), and then the retraction technique \(\mathcal{R}(\cdot) : \mathbb{R}^6 \rightarrow SE(3)\) lifts \(\xi\) to \(SE(3)\) as follows:

$$\mathcal{R}(\xi) = \begin{bmatrix} \exp(\xi_R^\top) \xi_t \\ 0 \end{bmatrix},$$

where exponential map \(\exp(\cdot) : so(3) \rightarrow SO(3)\) lifts \(so(3)\) to \(SO(3)\) and the hat operator \((\cdot)^\wedge : \mathbb{R}^3 \rightarrow \mathbb{R}^{3 \times 3}\) converts vector space \(\mathbb{R}^3\) to Lie algebra \(so(3)\) as follows:

$$\xi^\wedge = \begin{bmatrix} \xi_1 \xi_2 \xi_3 \\ -\xi_3 \xi_1 \xi_2 \\ \xi_2 \xi_1 \xi_3 \end{bmatrix}\in so(3).$$

It is worth noting that the hat operator of \(so(3)\) corresponds to a skew-symmetric matrix and has an important property:

$$\xi_R^\top v = -v^\wedge \xi_R, \forall v \in \mathbb{R}^3,$$

which will be useful later on.

Since the group action of \(SE(3)\) is matrix multiplication, we denote the current rigid transformation as \(T^0\) and assume the perturbation of rigid transformation is \(\xi \in \mathbb{R}^6\). Consequently, the optimization is with respect to \(\xi\), and the rigid transformation is updated as follows:

$$T^* = \mathcal{R}(\xi^*)T^0,$$

where \(\xi^*\) denotes the optimized perturbation. Specifically,

$$T^* = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \exp(\xi_R^\top) \xi_t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R^0 & t^0 \\ 0 & 1 \end{bmatrix}.$$

Based on the assumption of small perturbation, the exponential operator of \(so(3)\) can be approximated as:

$$\exp(\xi_R^\top) \approx I + \xi_R^\top.$$

We can substitute the Eq. (19) into the Eq. (18), which leads to the rigid transformation defined as:

$$R = \exp(\xi_R)v_R^0 \approx R^0 + \xi_R^\top v_R^0,$$

$$t = \exp(\xi_R^\top) t^0 + \xi_t \approx (I + \xi_R^\top)t^0 + \xi_t,$$

Then, the residual error is formulated as follows:

$$r_j = T v_j - \mu_j \approx (R^0 + \xi_R^\top v_R^0) v_j + t^0 + \xi_R^\top t^0 + \xi_t - \mu_j = \xi_R^\top (R^0 v_j) + \xi_R^\top t^0 + \xi_t + r_j^0,$$

Given the property depicted in Eq. (16), the residual error is converted into:

$$r_j = -(R^0 v_j + t^0)^\top \xi_j + r_j^0 = (H_j \xi_j + r_j^0),$$

where \(H_j = \begin{bmatrix} \sum_{j=1}^{N_j} (R^0 v_j + t^0)^\top \end{bmatrix} \in \mathbb{R}^{3 \times 6}\). By replacing \(r_j\) with \((H_j \xi_j + r_j^0)\), Eq. (13) is converted into a quadratic program problem depicted as:

$$\xi^* = \arg \min_{\xi \in \mathbb{R}^6} \xi^\top H \xi + 2b^\top \xi + c,$$

where the Hessian matrix \(H = \sum_{j=1}^{N_j} H_j\), \(H_j \in \mathbb{R}^{1 \times 6}\), the gradient vector \(b = \sum_{j=1}^{N_j} \sum_{i=1}^{N_j} (R^0 v_i + t^0)^\top \Omega_i r_i^0 \in \mathbb{R}^6\), the constant term \(c = \sum_{j=1}^{N_j} \sum_{i=1}^{N_j} (R^0 v_i + t^0)^\top \Omega_i r_i^0 \in \mathbb{R}^1\) and \(r_j^0 = (T^0 v_j - \mu_j)\).

The objective function Eq. (23) is a local approximation to the original objective function depicted in Eq. (13). Take the derivative of term \((\xi^\top H \xi + 2b^\top \xi + c)\) with respect to \(\xi\) and set it to zero, the optimal perturbation is calculated as follows:

$$\xi^* = -H^\top b,$$

where \((\cdot)^\top\) denotes the pseudo-inverse operator. Finally, the rigid transformation can be updated by Eq. (18), which is the optimal solution of Eq. (13). The Lie algebra optimizer can be sequentially utilized for each point set to optimize its rigid transformation.

B. Implementation

As a local convergent method, 3DMNDT requires to be provided with initial alignment. Given data points with arbitrary orientation, we utilize the coarse registration method [54] to estimate initial rigid transformations for multi-view registration. In 3DMNDT, there is a free parameter \(K\), which requires to be predefined. Generally, it is difficult to determine the optimal value of \(K\). However, we can set the average number \(N_c\) of data points in each cluster, which can indirectly determine the cluster number. Then, we uniformly sample \(K\) data points from all initial aligned point sets and take them as the initial centroids to start the iteration. To obtain desired results, the proposed method should alternately operate data point clustering, NDT computation, and rigid transformation optimization until all rigid transformations have no significant change or the iteration number \(s\) reaches the maximum value \(S\). Accordingly, 3DMNDT is summarized in Algorithm 1, where we set \(S = 300\) in the next section.
This section analyzes the computation complexity of 3DMNDT. Before analysis, we restate that there are \( M \) point sets being registered, where the \( i \)-th point set contains \( N_i \) data points and the total data points is denoted as \( N = \sum_{i=1}^{M} N_i \). Besides, these data points are divided into \( K \) clusters, and the number of valid clusters is \( K' < K \). For analysis convenience, we assume the iteration number of this method is \( S \). In each iteration, three operations are included to optimize one rigid transformation.

1) **Update Cluster:** Before updating the cluster, it requires building \( k \)-d tree for \( K \) cluster centroids, which leads to the total complexity \( O(SK \lg K) \) for \( S \) iterations. Then, each point should be assigned to one cluster by the nearest neighbor search, which leads to the total complexity \( O(SN \lg K) \) for \( S \) iterations.

2) **Compute NDT:** After updating the cluster, the NDT should be computed for each cluster. Since each data point is only assigned to one cluster and each cluster utilizes all its data points to compute one NDT, the complexity is \( O(N) \) in each iteration. For \( S \) iterations, the total complexity is \( O(SN) \).

3) **Optimize Rigid Transformation:** The proposed method utilizes \( N'_i \approx N_i \) point-cluster pairs to optimize the \( i \)-th rigid transformation. For the estimation of \( M \) rigid transformations, the total complexity is \( O(SN) \) for \( S \) iterations.

Table I displays the total computation complexity of the proposed method. As shown in Table I, the most complex operation is linear proportion to \( N \), \( S \), and \( \lg K \). Usually, the cluster number \( K \) is much smaller than the point number \( N \), and \( \lg K \) is much smaller than the cluster number \( K \). Therefore, 3DMNDT is an efficient registration method.

### Algorithm 1 3DMNDT Method

**Input:** Point sets \( \{V_i\}_{i=1}^{M} \) with initialization \( T^0, N_c \).

1. Determine \( K = \left| \sum_{i=1}^{M} N_i / N_c \right| \).
2. Uniformly initialize \( \mu_{i0}^s \), set \( s = 0 \);
3. repeat
   4. \( s = s + 1 \);
   5. Update correspondence \( c^s(i, j) \) by Eq. (7);
   6. Compute NDT \( (\mu_s^i, \Sigma_s^i) \) by Eqs. (8) and (9);
   7. for \( (i = 1 : M) \) do
      8. Optimize the perturbation \( \xi_s^i \) by Eq. (24);
      9. Update the transformation \( T_s^i \) by Eq. (18);
   end for
10. Compute likelihood \( L(\Theta^s) \) by Eq. (5);
11. until \( |L(\Theta^s) - L(\Theta^{s-1})| < \epsilon \) or \( (s > S) \)

### Computation Complexity of 3DMNDT in Each Operation

| Operation                  | Complexity |
|----------------------------|------------|
| Build k-d tree             | \( O(SK \lg K) \) |
| Build correspondence       | \( O(SN \lg K) \) |
| Compute NDT                | \( O(SN) \) |
| Optimize transformation    | \( O(SN) \) |

### C. Computational Complexity

This section analyzes the computation complexity of 3DMNDT. Before analysis, we restate that there are \( M \) point sets being registered, where the \( i \)-th point set contains \( N_i \) data points and the total data points is denoted as \( N = \sum_{i=1}^{M} N_i \). Besides, these data points are divided into \( K \) clusters, and the number of valid clusters is \( K' < K \). For analysis convenience, we assume the iteration number of this method is \( S \). In each iteration, three operations are included to optimize one rigid transformation.

1) **Update Cluster:** Before updating the cluster, it requires building \( k \)-d tree for \( K \) cluster centroids, which leads to the total complexity \( O(SK \lg K) \) for \( S \) iterations. Then, each point should be assigned to one cluster by the nearest neighbor search, which leads to the total complexity \( O(SN \lg K) \) for \( S \) iterations.

2) **Compute NDT:** After updating the cluster, the NDT should be computed for each cluster. Since each data point is only assigned to one cluster and each cluster utilizes all its data points to compute one NDT, the complexity is \( O(N) \) in each iteration. For \( S \) iterations, the total complexity is \( O(SN) \).

3) **Optimize Rigid Transformation:** The proposed method utilizes \( N'_i \approx N_i \) point-cluster pairs to optimize the \( i \)-th rigid transformation. For the estimation of \( M \) rigid transformations, the total complexity is \( O(SN) \) for \( S \) iterations.

Table I displays the total computation complexity of the proposed method. As shown in Table I, the most complex operation is linear proportion to \( N \), \( S \), and \( \lg K \). Usually, the cluster number \( K \) is much smaller than the point number \( N \), and \( \lg K \) is much smaller than the cluster number \( K \). Therefore, 3DMNDT is an efficient registration method.

### V. EXPERIMENT RESULTS

In this section, we evaluate the proposed 3DMNDT on various benchmark data sets, which include: (1) **Object data sets.** Four data sets are downloaded from the Stanford 3D Scanning Repository [55] and one data set is provided by Torsello [56]. Each was acquired from one object model in multiple viewpoints with the ground truth of rigid transformations. (2) **Environment data sets.** Three environment data sets are provided by ETH Autonomous Systems Lab [57]. They are all acquired by a mobile robot equipped with a laser range finder. During robot movement, range scans were acquired with the ground truth of the robot’s location. Gazebo data sets were acquired from the outdoor environment in summer and winter, where the robot mainly moved on the 2D plane ground and tracked the path to form a closed loop (4×5×0.09m). For simplicity, they are abbreviated as GazeboS and GazeboW, respectively. The apartment data set was acquired in structured indoor environments, where dynamic elements were moved in between scans. Table II displays some details of these benchmark data sets. For efficiency, each data set is uniformly down-sampled to around 2000 points per scan.

To illustrate its performance, we compare 3DMNDT with eight baseline methods:

1) **MATrICP [36].** It utilizes motion averaging algorithm to recover global motions from a set of relative motions, which are estimated by the TriICP algorithm.

2) **JRMP [8].** It supposes that all data points are drawn from a central GMM and utilizes the EM algorithm to estimate GMM components and rigid transformations.

3) **TMM [7].** It supposes that all data points are drawn from a central TMM and utilizes the EM algorithm to estimate the TMM components and rigid transformations.

4) **UnTMM [49].** It assumes that each data point is generated by one unique TMM, where each component centroid is defined by its nearest neighbor in other scans.

5) **LRS [39].** It concatenates available relative motions into one matrix, which is decomposed by LRS decomposition algorithm to recover global motions for multi-view registration.

6) **Kmeans [46].** It casts multi-view registration into the clustering problem and extends the k-means clustering algorithm to achieve multi-view registration.

7) **HKmeans [47].** This method casts multi-view registration into the clustering problem and extends the hierarchical k-means clustering algorithm to achieve multi-view registration.

8) **EMPMR [48].** This method assumes that each data point is generated by one unique GMM, where each Gaussian centroid is defined by its nearest neighbor in other scans.

Since all competing methods are locally convergent, they are provided with the same initial alignment, which is estimated...
Fig. 3. Registration performance of 3DMNDT under varied average point number \( N_c \) in each cluster. (a) Root mean square error (RMSE). (b) Runtime.

by the method [54]. For comparison, the ground truth of rigid transformations is utilized to evaluate the performance of all competing methods. Experimental results are reported in the form of Root Mean Square Error (RMSE) defined as:

\[
RMSE = \sqrt{\frac{1}{N_c} \sum_{i=1}^{M} \sum_{j=1}^{N_i} \| T_{m,i} v_{i,j} - T_{g,i} v_{i,j} \|^2_2},
\]

(25)

where \( T_{g,i} \) and \( T_{m,i} \) denote ground truth and estimated one of the \( i \)-th rigid transformation, respectively. Therefore, RMSE’s unit is similar to the data set unit. All competing methods are implemented on Matlab without any extra library and utilize the k-d tree method to search the nearest neighbor. Experiments are carried out on a quad-core 3.6 GHz computer with 8 GB memory.

A. Number of Clusters

Since 3DMNDT integrates the k-means clustering with Lie algebra optimizer to achieve multi-view registration, its performance is inevitably affected by the cluster number \( K \), which should be carefully determined. For different data sets, we can derive it by choosing the optimal point number \( N_c \) in each cluster. Accordingly, we test 3DMNDT on eight benchmark data sets with different values of \( N_c \) and compare its registration performance in RMSE and run time. Fig. 3 displays the registration performance of 3DMNDT for different values of \( N_c \).

As shown in Fig. 3, the performance of 3DMNDT is relatively stable as long as \( N_c \) is chosen in a suitable range, i.e., from 15 to 25 for most data sets. It becomes more efficient with the increase of \( N_c \), which is consistent with computation complexity analysis. When the value of \( N_c \) increases, the cluster number \( K \) will be decreased, which leads to low computation complexity. With the increase of \( N_c \), the accuracy of 3DMNDT results will obviously improve at first and then gradually worsen. For large data sets, the increase of \( N_c \) tends to obtain more accurate results for 3DMNDT. It is better to set a large value at the beginning for efficiency. After several iterations, we can set \( N_c = M \), when \( 15 \leq M \leq 25 \). To ensure accuracy, we force \( N_c = 15 \) when \( M < 15 \) and force \( N_c = 25 \) when \( M > 25 \).

B. Ablation Study

Since the original NDT utilizes the standard space division and the Newton optimizer to achieve pair-wise registration, we compare the proposed 3DMNDT with two NDT variants to illustrate its validation. 1) NDTO. This variant replaces the k-means clustering and the Lie algebra optimizer in 3DMNDT by the standard space division and the Newton optimizer, respectively. 2) NDTL. This variant only replaces the Lie algebra optimizer in 3DMNDT with the Newton optimizer. All these three NDT variants are tested on eight benchmark data sets. Experimental results are reported in the form of RMSE and runtime, which are displayed in Table III. As shown in Table III, 3DMNDT can obtain more accurate registration results among three NDT variants. Their efficiency is very close for eight data sets.

Fig. 4 displays NDT representations of Gazebo Summer for the standard space division and the k-means clustering under different alignments, where each ellipsoid indicates one NDT and red points denote these data points in invalid NDTs containing no more than 5 data points. (a) Standard space partition under initial alignment. (b) Standard space partition under the ground truth alignment. (c) K-means clustering under initial alignment. (d) K-means clustering under the ground truth alignment.

| Dataset     | Initial | NDTO | NDTL | 3DMNDT |
|-------------|---------|------|------|--------|
| Armadillo   | 1.4166  | 2.4815 | 0.6417 | 1.6315  | 0.3899 | 1.0217 | 0.3408 |
| Bunny       | 3.5363  | 1.2315 | 0.2864 | 0.8006  | 0.2362 | 0.2362 | 0.2912 |
| Buddha      | 3.5827  | 2.2864 | 0.9572 | 1.6214  | 0.4411 | 1.0386 | 0.4992 |
| Dragon      | 3.3861  | 4.8821 | 0.8629 | 2.2056  | 0.4655 | 0.9611 | 0.4717 |
| Hand        | 1.4132  | 0.3645 | 1.6592 | 1.8913  | 1.7617 | 0.0259 | 0.7166 |
| Apartment   | 0.1335  | 0.1247 | 1.9139 | 0.0982  | 1.3817 | 0.0518 | 1.1883 |
| GazeboS     | 0.2121  | 0.0580 | 0.4198 | 0.2506  | 1.1454 | 0.0421 | 0.5737 |
| GazeboW     | 0.1623  | 1.1209 | 1.6947 | 0.4695  | 0.4342 | 0.0552 | 0.6238 |

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
clustering can utilize most sparse data points far from the
sensor by sparse NDTs, but the standard space division fails
to use them due to restricting the minimum point number
in NDT. As shown in Fig. 4, different division strategies
lead to different NDT representation results. In NDTO and
NDTL, the occupied space of all point sets is divided into
standard voxels, which inevitably leads to unbalanced division.
Specifically, some voxels may only contain a small number of
data points, i.e., smaller than 6. To avoid singular covariance,
data points in these voxels are neglected to compute NDT
for the optimization of rigid transformations, which leads to
information loss and reduces the performance of multi-view
registration. Since the poses of all point sets constantly change
during multi-view registration, the standard space division
reduces the continuity of NDT representation between two
iterations. Accordingly, NDTO and NDTL may be challenging
to achieve promising results for multi-view registration.

While 3DMNDT computes NDT by the k-means clustering,
which can avoid unbalanced division and reduce the number
of invalid NDTs. Since the k-means clustering updates the
current clustering according to the previous clustering, it keeps
the continuity of NDT representation between two iterations.
Compared with NDTO and NDTL, 3DMNDT tends to obtain
more accurate registration results due to more valid NDTs
with increased continuity. Further, the Lie algebra optimizer
directly optimizes rotation matrices in the likelihood function.
Compared with the Newton optimizer, this optimizer can
avoid computing the Jacobian and Hessian matrices. Moreover,
it does not require swapping rotation angles and rotation
matrices.

C. Convergence

In theory, it isn’t easy to prove the convergence of
3DMNDT. Accordingly, we explore the convergence properties
of 3DMNDT on eight data sets. Fig. 5 demonstrates the convergence of 3DMNDT for iterations.

As shown in Fig. 5, the log-likelihood value tends to rise
with the increase of iterations, but it is not monotonically increasing. Because the k-means clustering may introduce invalid NDTs, the Lie algebra optimizer is not a closed-form
solution. These two reasons may decrease the log-likelihood value at some iterations. It should be noted that the log-
likelihood value finally converges to a stable value within
acceptable iterations. Therefore, the 3DMNDT has good con-
vergence performance.

D. Object Data Sets

To illustrate its registration performance, 3DMNDT is tested
on object data sets and compared with several baseline meth-
ods on accuracy, efficiency, and robustness.

1) Accuracy: For the comparison of registration accuracy,
all competing methods are tested on five object data sets with
experimental results reported in the form of RMSE. Table IV
illustrates the registration RMSE of all competing methods
tested on five object data sets. For a more intuitive comparison,
Fig. 6 displays these registration results in the cross-section.
As shown in Table IV and Fig. 6, 3DMNDT can obtain the
most promising registration results for all five data sets.

For the multi-view registration, both MATrICP and LRS
recover global motions from a set of available relative motions
estimated by the TrICP algorithm. For accurate relative
motions, these two methods may obtain promising registration
results. Since the TrICP algorithm is not error-free, relative
motions inevitably contain errors, which reduce the accuracy
of multi-view registration. Besides, MATrICP is sensitive to
unreliable relative motions, and even one relative motion
will lead to its registration failure, e.g., worse registration
results than initialization. Meanwhile, LRS depends on the
ratio of available relative motions, and a low ratio will lead
to registration failure, e.g., worse registration results than
initialization.

As probabilistic registration methods, both JRMPC and
TMM are expected to obtain accurate results. However, this
is not the case because both of them require estimating many

| Method | Armadillo | Bunny | Buddha | Dragon | Hand |
|--------|----------|-------|--------|--------|------|
| Initial | RMSE     | Time(min.) |
| MATrICP | 8.5450   | 1.2750 |
| JRMPC   | 5.0816   | 1.0551 |
| TMM     | 6.0787   | 1.0217 |
| UnTMM   | 0.3046   | 0.0276 |
| Kmeans  | 2.5730   | 0.2883 |
| HKmeans | 1.5092   | 0.7651 |
| EMPMR   | 1.3164   | 0.4909 |
| 3DMNDT  | 1.0217   | 0.3408 |

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
mixture model parameters and rigid transformations, making them be easily trapped into a local minimum. While EMPMR and UnTMM only need to estimate rigid transformations and one variance, it is more likely to obtain accurate results.

Although Kmeans, HKmeans, and 3DMNDT utilize k-means clustering to achieve multi-view registration, their performance is different. In Kmeans, all data points in each cluster are approximated by one centroid, which inevitably leads to information loss. As Kmeans directly aligns each point set to cluster centroids for registration, it is challenging to obtain promising results. To reduce information loss, HKmeans gradually increases cluster number to register multiple point sets and finally utilizes the coarse-to-fine method [30] to achieve accurate registration results. Like the coarse-to-fine method, HKmeans tends to be trapped into local minimum with increasing point sets. In contrast, 3DMNDT computes NDT for each cluster and achieves multi-view registration by maximizing the likelihood function. Since NDT contains statistical information on each cluster, 3DMNDT is more likely to obtain accurate registration results. Thanks to k-means clustering and the Lie algebra optimizer, 3DMNDT can obtain promising performance for multi-view registration.

2) Efficiency: All competing methods are tested on five object data sets for efficiency comparison. Experimental results are reported in averaging run time over 10 independent tests and are shown in Table IV. As shown in Table IV, Kmeans is the most efficient among all competing methods. 3DMNDT also belongs to an efficient algorithm. But TMM and JRPMC are more time-consuming than other registration methods.

As aforementioned, MATrICP and LRS recover global motions from available relative motions. Therefore, they require building point correspondences between most scan pairs. Since the convergence speed of MATrICP is slower than that of LRS, it is less efficient than LRS. Besides, both EMPMR and UnTMM require building point correspondences between each scan pair. Accordingly, their efficiency is close to MATrICP but slightly less efficient than LRS.

Like Kmeans, HKmeans and 3DMNDT also utilize k-means clustering algorithm to achieve multi-view registration. To optimize rigid transformations, Kmeans computes a centroid and minimizes the sum function of point-centroid errors. HKmeans implements Kmeans several times by increasing the number of clusters, so it is less efficient than Kmeans. While 3DMNDT only requires computing NDT and minimizing the NDT-based likelihood function. Therefore, it is a little less efficient than Kmeans but more efficient than HKmeans, especially for Hand data sets. Considering registration accuracy, 3DMNDT is an efficient method.

3) Robust to Initialization: To illustrate its robustness to initialization, 3DMNDT and other baseline methods are also tested on object data sets under different initial rigid transformations, where rotation angles are added with three levels of uniformly distributed noises. Each experiment group is carried out by 20 independent tests to eliminate randomness. Experimental results are reported in the form of RMSE’s mean and standard deviation. Table V demonstrates comparison
results for all competing methods. Under small and moderate noise levels, 3DMNDT is generally the most robust one to initialization among all competing methods. Under significant noise levels, HKmeans is the robust one, and 3DMNDT may obtain promising results for some data sets.

MATrICP and LRS recover global motions from a set of relative motions estimated by the pair-wise registration algorithm. Given a set of reliable relative motions, MATrICP can obtain promising registration results. However, this method is sensitive to unreliable relative motions. Without good initialization, the probability of getting unreliable relative motions will increase, leading to the failure of MATrICP. Although LRS is robust to unreliable relative motions, it is sensitive to the ratio of reliable relative motions. Under poor initialization, the number of reliable relative motions will be reduced, leading to the failure of LRS. Therefore, LRS is also sensitive to initialization. Since JRMPC and TMM require estimating many model parameters, they are more likely trapped into a local minimum, especially when the initialization is poor. Therefore, both of them are also sensitive to initialization.

Benefiting from data clustering, Kmeans, HKmeans, and 3DMNDT have broad convergent domains. However, Kmeans is challenging to obtain promising registration results due to information loss. By reducing information loss, HKmeans can get more promising registration results. Since Kmeans with a small cluster number can provide good initialization for the coarse-to-fine registration method in HKmeans, HKmeans is the most robust among all competing methods under significant noise levels.

As 3DMNDT utilizes one NDT to represent data points of each cluster, it is also likely to obtain promising results due to less information loss. Under small and moderate noise levels, 3DMNDT is the most robust one to initialization among all competing methods. Under significant noise levels, good initialization should be provided by other methods. Otherwise, it is challenging to obtain promising registration results. Other competing methods also share this limitation, except for HKmeans, which utilizes Kmeans to provide good initialization for the coarse-to-fine process.

4) Robustness to Data Noise: To illustrate its robustness to noises, 3DMNDT and other baseline methods are tested on object data sets added with random Gaussian noises. Specifically, each data point is shifted with randomly generated Gaussian noise. Each experiment group is carried out by 20 independent tests to eliminate randomness. Experimental results are reported in the form of RMSE's mean and standard deviation. Table VI demonstrates comparison results under two noise levels. As shown in Tables VI, the accuracy of all methods decreases with the increase in noise level. In general, 3DMNDT is more robust to data noise than other methods. Theoretically, JRMPC and TMM are probabilistic methods where data noises are considered in the mixture model. However, they are more likely trapped into local minimum due to a massive number of model parameters required to be optimized. Therefore, they seem to be sensitive to data noise.

Compared with JRMPC and TMM, EMPMR and UnTMM need to optimize fewer model parameters, and they are more likely to obtain promising registration results. Since EMPMR and UnTMM directly formulate data noise in the objective function, they are robust to data noise. Although both MATrICP and LRS do not consider data noise, they seem robust to data noise for some data sets. Because these two methods recover

---

**TABLE V**

| Method   | Initialization | Armadillo | Bunny | Buddha | Dragon | Hand |
|----------|----------------|-----------|-------|--------|--------|------|
| MATrICP  | [0.01,0.01]    | 5.2854±0.1067 | 0.9019±0.0170 | 7.8077±7.3019 | 23.1167±10.3668 | 21.956±10.0009 |
|          | [0.03,0.03]    | 18.7410±2.1579 | 0.9071±0.0114 | 23.1167±10.1111 | 29.2487±10.7262 | 21.954±10.0004 |
| JRMPC    | [0.01,0.01]    | 1.9063±0.2880 | 2.1685±0.5556 | 2.6141±0.4274 | 3.0703±0.6607 | 0.7128±0.0271 |
|          | [0.03,0.03]    | 3.1661±1.1472 | 5.3964±0.9213 | 6.0592±5.3163 | 7.4510±7.0525 | 1.1634±1.0486 |
| TMM      | [0.01,0.01]    | 7.3424±1.5520 | 8.6662±7.9892 | 9.5398±8.2583 | 11.2038±11.8084 | 2.1246±1.2275 |
|          | [0.03,0.03]    | 4.9782±1.645 | 3.5077±0.8859 | 3.8900±0.9231 | 3.2862±0.4914 | 1.2116±0.2176 |
| HKmeans  | [0.01,0.01]    | 1.0690±0.1156 | 0.3453±0.0481 | 1.1490±0.0926 | 1.1901±0.0818 | 0.3535±0.0405 |
|          | [0.02,0.02]    | 1.5002±1.4955 | 0.3638±0.0567 | 1.2984±0.6604 | 1.2554±1.8876 | 0.9978±0.8962 |
|          | [0.03,0.03]    | 5.0383±3.0360 | 0.3893±0.0653 | 3.1719±2.5601 | 6.8979±3.2282 | 2.6998±1.2352 |
| Kmeans   | [0.01,0.01]    | 8.8786±5.2714 | 1.2877±3.6070 | 2.5116±0.5537 | 1.6877±2.8353 | 1.9932±0.0013 |
|          | [0.02,0.02]    | 16.3962±7.5955 | 3.3597±1.4908 | 24.2722±0.5127 | 7.9758±7.8579 | 1.9934±0.0022 |
|          | [0.03,0.03]    | 16.6474±4.1015 | 7.6745±5.8488 | 24.7031±3.0443 | 23.1166±8.1132 | 1.9935±0.0024 |
| LRS      | [0.01,0.01]    | 2.8694±0.2866 | 2.4612±0.1898 | 2.9905±0.1744 | 2.1835±0.4049 | 0.4924±0.0494 |
|          | [0.02,0.02]    | 3.2403±0.6221 | 3.2988±0.6182 | 3.2981±0.4193 | 2.9399±0.4403 | 1.2751±0.2265 |
|          | [0.03,0.03]    | 4.8559±1.3240 | 3.7662±0.6520 | 4.4053±1.1357 | 6.9225±1.7746 | 2.3501±0.2171 |
| EMPMR    | [0.01,0.01]    | 1.1134±0.0808 | 0.3146±0.0281 | 1.1162±0.0508 | 1.1195±0.0485 | 0.0535±0.0054 |
|          | [0.02,0.02]    | 3.4653±1.9403 | 1.9380±1.4624 | 2.0685±1.5769 | 3.3768±2.0585 | 1.0821±0.9266 |
|          | [0.03,0.03]    | 8.6751±2.8260 | 6.0722±2.0524 | 6.3262±2.8246 | 8.8145±2.3832 | 2.0558±1.1157 |
| 3DMNDT   | [0.01,0.01]    | 1.0219±0.0537 | 0.2349±0.0272 | 1.0142±0.0304 | 1.0356±0.0347 | 0.3277±0.0402 |
|          | [0.02,0.02]    | 1.1309±1.0902 | 0.2593±0.0440 | 1.0780±0.1338 | 2.2332±1.3410 | 0.1280±0.1226 |
|          | [0.03,0.03]    | 3.1602±2.9285 | 0.7340±1.1434 | 2.7145±1.7058 | 6.1479±2.4547 | 1.3333±0.7031 |

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
TABLE VI

| Method | Noise  | Armadillo | Bunny | Buddha | Dragon | Hand |
|--------|--------|-----------|-------|--------|--------|------|
| MATrICP   | 15dB   | 7.6460±1.4120 | 0.9407±0.0535 | 1.5841±0.0257 | 1.9059±0.0311 | 1.9914±0.0066 |
|           | 25dB   | 7.3109±1.5760 | 0.8923±0.0248 | 1.4974±0.0147 | 1.9090±0.0413 | 1.9686±0.0022 |
| JRMPC    | 15dB   | 1.8415±0.0372 | 1.8639±0.1007 | 2.6992±0.0763 | 2.6269±0.0576 | 0.8411±0.0398 |
|           | 25dB   | 1.7970±0.0164 | 1.8618±0.1145 | 2.6839±0.0552 | 2.6075±0.0386 | 0.7443±0.0052 |
| TMM      | 15dB   | 1.6606±0.0381 | 1.3355±0.0292 | 1.7827±0.0308 | 2.2797±0.0523 | 0.7411±0.0387 |
|           | 25dB   | 1.6492±0.0492 | 1.3234±0.0305 | 1.7661±0.0367 | 2.2638±0.0447 | 0.7322±0.0086 |
| UnTMM    | 15dB   | 1.1158±0.0292 | 0.3376±0.0449 | 1.2770±0.0391 | 1.0555±0.0295 | 0.4005±0.0373 |
|           | 25dB   | 1.0803±0.0115 | 0.3003±0.0130 | 1.2287±0.0168 | 1.0387±0.0129 | 0.1185±0.0174 |
| LRS      | 15dB   | 14.4703±0.3025 | 1.3444±0.1622 | 1.6359±0.0436 | 2.2229±0.1583 | 0.9284±0.0525 |
|           | 25dB   | 14.4865±0.3054 | 1.2793±0.1174 | 1.6225±0.0227 | 2.1696±0.1718 | 0.8951±0.0307 |
| Kmeans   | 15dB   | 2.7468±0.0563 | 2.4367±0.0626 | 2.1823±0.0392 | 2.8130±0.0430 | 0.4474±0.0085 |
|           | 25dB   | 2.7076±0.0421 | 2.4319±0.0442 | 2.1877±0.0204 | 2.8356±0.0224 | 0.4048±0.0049 |
| HKmeans  | 15dB   | 1.5616±0.4376 | 0.6083±0.2905 | 1.1328±0.1066 | 1.9527±0.8616 | 0.3742±0.0561 |
|           | 25dB   | 2.2111±0.4131 | 0.5560±0.1364 | 1.0362±0.0823 | 1.3399±0.0883 | 0.1521±0.0435 |
| EMPMR    | 15dB   | 1.1858±0.0413 | 0.3090±0.0449 | 1.2461±0.0339 | 1.1111±0.0736 | 0.3731±0.0291 |
|           | 25dB   | 1.1372±0.0151 | 0.2996±0.0240 | 1.2029±0.0187 | 1.1431±0.1135 | 0.0996±0.0122 |
| 3DMNDT   | 15dB   | 1.0414±0.0167 | 0.2534±0.0164 | 1.0668±0.0186 | 0.9765±0.0198 | 0.3122±0.0052 |
|           | 25dB   | 1.0172±0.0080 | 0.2441±0.0084 | 1.0378±0.0097 | 0.9503±0.0118 | 0.0915±0.0011 |

global motions from a group of relative motions, the influence of data noise on multi-view registration may be eliminated by the motion averaging or LRS matrix decomposition algorithm. However, high-level noises may result in unreliable relative motions, leading to multi-view registration failure. Meanwhile, both Kmeans and HKmeans also do not consider data noises, so they are less robust than EMPMR and UnTMM.

Unlike EMPRM and UnTMM, 3DMNDT does not directly formulate data noises in the objective function. However, it utilizes the NDT-based likelihood function, which allows data containing noises. Therefore, it is also robust to noise and can achieve promising registration results under two noise levels.

E. Environment Data Sets

Usually, multi-view registration methods are potentially applied to scene reconstruction. To illustrate its performance, 3DMNDT and other baseline methods are tested on three environment data sets for scene reconstruction. Before registration, we use the height value to filter ground surface data for GazeboS and GazeBoW data sets. The proposed 3DMNDT and six baseline methods register one point set to all different point sets aligned by the current estimation of rigid transformations. Therefore, they do not require that all point sets mutually overlap in each data set. In contrast, MATrICP and LRS recover global motions from a set of relative motions. Only one unreliable relative motion can cause the failure of motion averaging in MATrICP and lead to unexpected registration results. Although LRS is robust to unreliable relative motions, it requires that most relative motions are available. Otherwise, it is challenging to obtain promising registration results. However, we can neither guarantee that all relative motions are reliable nor get most of the relative motions due to low or non-overlap percentages between two point sets in environment data sets.

It seems that JRMPC has a solid theory, but it is also challenging to obtain promising registration results due to the tight coupling between rigid transformations and Gaussian components. In JRMPC, each Gaussian component is estimated by all range points aligned by rigid transformations. Estimating the $i^{th}$ rigid transformation requires computing the soft assignment between one point in the $i^{th}$ point set and all other points in the point set, which makes the computation expensive.

In terms of efficiency, only Kmeans is more efficient than 3DMNDT, but it is less accurate than 3DMNDT due to the information loss in clustering.

It should be noted that MATrICP and LRS may obtain unexpected registration results with accuracy worse than the initial alignment. Both MATrICP and LRS recover global motions from a set of relative motions. Only one unreliable relative motion can cause the failure of motion averaging in MATrICP and lead to unexpected registration results. Although LRS is robust to unreliable relative motions, it requires that most relative motions are available. Therefore, it is challenging to obtain promising registration results. However, we can neither guarantee that all relative motions are reliable nor get most of the relative motions due to low or non-overlap percentages between two point sets in environment data sets.

It seems that JRMPC has a solid theory, but it is also challenging to obtain promising registration results due to the tight coupling between rigid transformations and Gaussian components. In JRMPC, each Gaussian component is estimated by all range points aligned by rigid transformations. Estimating the $i^{th}$ rigid transformation requires computing the soft assignment between one point in the $i^{th}$ point set...
to all Gaussian components. This interdependence increases the difficulty of obtaining promising results. Besides, many Gaussian component parameters and rigid transformations must be optimized, making JRMPC to be easily trapped into a local minimum. Since TMM is developed from JRMPC, they share all the same disadvantages.

Kmeans can obtain promising registration results for three environment data sets efficiently. By reducing information loss, HKmeans may get more favorable results but costs more time. But sometimes, this method may easily trap into a local minimum, e.g., GazeboS data set. UnTMM and EMPRM can obtain promising results. They are more efficient than HKmeans, but cost more time than Kmeans.

In 3DMNDT, each NDT seems equal to one Gaussian component of GMM; its covariance is the by-product of k-means clustering and does not require to be optimized. Besides, the cluster center of each NDT is only estimated by these points belonging to one cluster, and each point only requires building one point-to-cluster correspondence. Therefore, each rigid transformation is not associated with all NDTs. Since rigid transformations and NDTs need to be alternately optimized, the loose association contributes to obtaining the most accurate registration results. What’s more, it is more efficient than most baseline methods.

VI. CONCLUSION

This paper proposes a novel NDT-based method named 3DMNDT for multi-view registration. We formulate the multi-view registration as the NDT-based likelihood function, which is optimized by integrating k-means clustering and the Lie algebra optimizer under the framework of point-to-cluster correspondence. 3DMNDT utilizes the k-means clustering algorithm to cluster data points and then computes NDT for each cluster. Compared with standard space division, the K-means clustering can avoid unbalanced division of data points and increases the continuity of NDT representation between two iterations. As the substitute for the Newton optimizer, the Lie algebra optimizer is simple and can optimize the NDT-based likelihood function with promising results. Experimental results tested on benchmark data sets illustrate that 3DMNDT can achieve state-of-the-art performance for multi-view registration.

REFERENCES

[1] R. Sandhu, S. Dambreveil, 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.
[2] 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.
[3] J. Ma, J. Zhao, J. Jiang, H. Zhou, and X. Guo, “Locality preserving matching,” Int. J. Comput. Vis., vol. 127, no. 5, pp. 512–531, 2019.
[4] W. Tabib, C. O’Meachra, and N. Michael, “On-manifold GMM registration,” IEEE Robot. Autom. Lett., vol. 3, no. 4, pp. 3805–3812, Oct. 2018.
[5] Z. Jiang, J. Zhu, Z. Lin, Z. Li, and R. Guo, “3D mapping of outdoor environments by scan matching and motion averaging,” Neurocomputing, vol. 372, pp. 17–32, Jan. 2020.
[6] A. Rasoulian, R. Rohling, and P. Abolmaesumi, “Group-wise registration of point sets for statistical shape models,” IEEE Trans. Med. Imag., vol. 31, no. 11, pp. 2025–2034, Nov. 2012.
[7] N. Ravikumar, A. Gooya, S. Çimen, A. F. Frangi, and Z. A. Taylor, “Group-wise similarity registration of point sets using student’s t-mixture model for statistical shape models,” Med. Image Anal., vol. 44, pp. 156–176, Feb. 2018.
[8] 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.
[9] P. J. Besl and N. D. McKay, “Method for registration of 3-D shapes,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 14, no. 2, pp. 239–256, Feb. 1992.
[10] P. Biber and W. Strasser, “The normal distributions transform: A new approach to laser scan matching,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS), vol. 3, Oct. 2003, pp. 2743–2748.
[11] M. Magnusson, “The three-dimensional normal-distributions transform: An efficient representation for registration, surface analysis, and loop detection,” Ph.D. dissertation, Dept. Technol., Örebro Universitet, Örebro, Sweden, 2009.
[12] A. Das and S. L. Waslander, “Scan registration with multi-scale k-means normal distributions transform,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., Oct. 2012, pp. 2705–2710.
[13] P. Biber and W. Strasser, “NScan-matching: Simultaneous matching of multiple scans and application to SLAM,” in Proc. IEEE Int. Conf. Robot. Autom. (ICRA), May 2006, pp. 2270–2276.
[14] S. Rusinkiewicz and M. Levoy, “Efficient variants of the ICP algorithm,” in Proc. Int. Conf. 3-D Digit. Imag. Modeling (3DIM), vol. 1, 2001, pp. 145–152.
[15] F. Pomereux, F. Colas, R. Siegwart, and S. Magnenat, “Comparing ICP variants on real-world data sets,” Auto. Robots, vol. 34, no. 3, pp. 133–148, 2013.

[16] D. Chetverikov, D. Stepanov, and P. Krske, “Robust Euclidean alignment of 3D point sets: The trimmed iterative closest point algorithm,” Image Vis. Comput., vol. 23, no. 3, pp. 299–309, 2005.

[17] M. Magnusson, N. Vaskevicius, T. Stoyanov, K. Pathak, and A. Birk, “Beyond points: Evaluating recent 3D scan-matching algorithms,” in Proc. IEEE Int. Conf. Robot. Autom. (ICRA), May 2015, pp. 3631–3637.

[18] 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.

[19] 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.

[20] W. Gao and R. Tedrake, “FilterReg: Robust and efficient probabilistic point-set registration using Gaussian filter and twist parameterization,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 11095–11104.

[21] I. Sipiran and B. Bhanu, “Harris 3D: A robust extension of the Harris operator for interest point detection on 3D meshes,” Vis. Comput., vol. 27, no. 11, pp. 963–976, 2011.

[22] H. Lei, G. Jiang, and L. Quan, “Fast descriptors and correspondence propagation for robust global point cloud registration,” IEEE Trans. Image Process., vol. 26, no. 8, pp. 3614–3623, Aug. 2017.

[23] Y. Aoki, H. Goforth, R. A. Srivatsan, and S. Lucey, “PointNetLK: Robust & efficient point cloud registration using PointNet,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 7163–7172.

[24] Y. Wang and J. Solomon, “Deep closest point: Learning representations for point cloud registration,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 3523–3532.

[25] 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/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 1–10.

[26] Y. Choy, W. Dong, and V. Koltun, “Deep global registration,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 2514–2523.

[27] S. Huang, Z. Gojicic, M. Usvyatsov, A. Wieser, and K. Schindler, “PRDATOR: Registration of 3D point clouds with low overlap,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 7193–7203.

[28] C. Choy, W. Dong, and V. Koltun, “Deep global registration,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 2467–2476.

[29] S. Huang, Z. Gojicic, M. Usvyatsov, A. Wieser, and K. Schindler, “PRDATOR: Registration of 3D point clouds with low overlap,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 7193–7203.

[30] Y. Chen and G. Medioni, “Object modelling by registration of multiple 3D views,” Comput. Graph. Forum, vol. 34, no. 5, pp. 77–87, 2015.

[31] X. Ge, Q. Zhu, L. Huang, S. Li, and S. Li, “Globalization of multiview unordered forest point clouds guided by common subgraphs,” IEEE Trans. Geosci. Remote Sens., vol. 60, pp. 1–14, 2021.

[32] S. Krishnan, P. Y. Lee, J. B. Moore, and S. Venkatasubramanian, “Global registration of multiple 3D point sets via optimization-on-a-manifold,” in Proc. Symp. Geometry Process., 2005, pp. 187–196.

[33] R. Toldo, A. Beinat, and F. Cossola, “Global registration of multiple point clouds embedding the generalized Procrustes analysis into an ICP framework,” in Proc. Int. Symp. 3D Data Process., Vis. Transmiss., vol. 2, 2010, pp. 1–8.

[34] X. Mateo, X. Orells, and X. Binefa, “Bayesian perspective for the registration of multiple 3D views,” Comput. Vis. Image Understand., vol. 118, pp. 84–96, Jan. 2014.

[35] Y. Ma, J. Zhu, Z. Tian, and Z. Li, “Effective multiview registration of multiple views,” IEEE Trans. Autom. Sci. Eng., vol. 17, no. 4, pp. 1337–1351, Oct. 2020.

[36] V. M. Govindu and A. Pooja, “On averaging multiview relations for 3D scan registration,” IEEE Trans. Image Process., vol. 23, no. 3, pp. 1289–1302, Mar. 2014.

[37] J. Solà, J. Deray, and D. Atchuthan, “A micro lie theory for state estimation in robotics,” 2018, arXiv:1812.01537.

[38] F. Arrigoni, B. Rossi, and A. Fusiello, “Global registration of 3D point sets via LRS decomposition,” in Proc. Eur. Conf. Comput. Vis. Amsterdam, The Netherlands: Springer, 2016, pp. 489–504.

[39] F. Arrigoni, B. Rossi, P. Fragnetto, and A. Fusiello, “Robust synchronization in SO(3) and SE(3) via low-rank and sparse matrix decomposition,” Comput. Vis. Image Understand., vol. 174, pp. 95–113, Sep. 2018.

[40] R. Guo, J. Zhu, Y. Li, D. Chen, Z. Li, and Y. Zhang, “Weighted motion averaging for the registration of multi-view range scans,” Multimedia Tools Appl., vol. 77, no. 9, pp. 10651–10668, May 2018.

[41] J. Zhu, J. Hu, H. Lu, B. Chen, Z. Li, and Y. Li, “Robust motion averaging under maximum correntropy criterion,” in Proc. IEEE Int. Conf. Robot. Autom. (ICRA), May 2021, pp. 5283–5288.

[42] Z. Min, J. Wang, and M. Q.-H. Meng, “Robust generalized point cloud registration with orientational data based on expectation maximization,” IEEE Trans. Autom. Sci. Eng., vol. 17, no. 1, pp. 207–221, Jan. 2020.

[43] Z. Min, J. Wang, and M. Q.-H. Meng, “Joint rigid registration of multiple generalized point sets with anisotropic positional uncertainties in image-guided surgery,” IEEE Trans. Autom. Sci. Eng., vol. 19, no. 4, pp. 3612–3627, Oct. 2022.

[44] R. Guo, J. Zhu, and J. Solomon, “Hierarchical K-means clustering for registration of multi-view point sets,” Comput. Electr. Eng., vol. 94, Sep. 2021, Art. no. 107321.

[45] Z. Min, J. Zhu, J. Hu, B. Chen, Z. Li, and Y. Li, “Robust motion averaging under maximum correntropy criterion,” in Proc. IEEE Int. Conf. Robot. Autom. (ICRA), May 2021, pp. 5283–5288.