Registration of 3D Point Sets Using Exponential-based Similarity Matrix

Ashutosh Singandhupe, Sanket Lokhande, Hung Manh La

arXiv:2505.04540v1 [cs.CV] 7 May 2025

Abstract—**Point cloud registration is a fundamental problem in computer vision and robotics, involving the alignment of 3D point sets captured from varying viewpoints using depth sensors such as LiDAR or structured light. In modern robotic systems, especially those focused on mapping, it is essential to merge multiple views of the same environment accurately. However, state-of-the-art registration techniques often struggle when large rotational differences exist between point sets or when the data is significantly corrupted by sensor noise. These challenges can lead to misalignments and, consequently, to inaccurate or distorted 3D reconstructions. In this work, we address both these limitations by proposing a robust modification to the classic Iterative Closest Point (ICP) algorithm. Our method, termed Exponential Similarity Matrix ICP (ESM-ICP), integrates a Gaussian-inspired exponential weighting scheme to construct a similarity matrix that dynamically adapts across iterations. This matrix facilitates improved estimation of both rotational and translational components during alignment. We demonstrate the robustness of ESM-ICP in two challenging scenarios: (i) large rotational discrepancies between the source and target point clouds, and (ii) data corrupted by non-Gaussian noise. Our results show that ESM-ICP outperforms traditional geometric registration techniques as well as several recent learning-based methods. To encourage reproducibility and community engagement, our full implementation is made publicly available on GitHub. https://github.com/aralab-unr/ESM_ICP**

Index Terms—**Registration, 3D point cloud, Iterative Closest Point**

I. INTRODUCTION

Aligning point clouds/point-sets is a fundamental and active area of research in the field of computer vision and robotics. Also referred to as geometric registration, it involves aligning two 3D point sets representing the same rigid structure or scene, typically captured from different viewpoints using 3D sensors such as LiDAR or structured-light scanners. Applications span diverse domains, including robotics, medical imaging, autonomous driving, and computational chemistry, where accurate alignment of 3D data is crucial. Formally, registration aims to estimate the rigid transformation—comprising rotation and translation—that best aligns a source point cloud with a target point cloud by minimizing a suitable alignment error metric. In this paper, we use the terms: source and the target to denote the point clouds being aligned throughout the discussion. Among the earliest and most widely adopted registration techniques is the Iterative Closest Point (ICP) algorithm [1], [3], which computes correspondences between point sets and refines the transformation iteratively. While ICP performs well

when the initial misalignment between the source and the target is small, it is prone to convergence to local minima in the presence of large rotational or translational offsets. In practice, the absence of a good initial estimate and the presence of noise or outliers often degrade ICP's performance. Over the past three decades, numerous variants of ICP have been proposed to address these limitations. These include point-to-plane ICP [8], non-linear ICP [40], [42], Generalized ICP, and Normal Distributions Transform (NDT) [25]. While these methods offer improvements in specific scenarios, they remain sensitive to large misalignments and noisy data. Alternative techniques such as Go-ICP [10] and others [27], [33], [36] attempt global optimization over the motion space SE(3), but often incur high computational costs and can still fail under large rotations.

With the advent of deep learning, several learning-based registration methods have emerged, including Deep Closest Point (DCP) [47], PointNetLK [51], PRNet [53], and DeepGMR [55]. While these methods demonstrate promising results on benchmark datasets, they often exhibit limited generalizability and robustness under extreme transformations or noise, particularly when initial alignment is poor or the data is corrupted.

In this paper, we propose a novel registration framework called Exponential Similarity Matrix ICP (ESM-ICP), designed to address the shortcomings of both traditional and learning-based approaches. Building upon the work [5], [6], [7], [4], [2], ESM-ICP introduces a similarity matrix that captures weighted correspondences between the source and target point clouds using a Gaussian-based exponential kernel. This matrix is sparse and updated iteratively during registration, enhancing robustness to large transformations and outliers. Key features of our method include: (1) The use of a similarity matrix that encodes the confidence of point correspondences, updated at every iteration. (2) Robust performance under large rotation and translation ranges, even when the point sets are heavily misaligned or corrupted by noise. (3) Superior alignment quality compared to state-of-the-art geometric and learning-based methods. We evaluate ESM-ICP on both clean and noisy datasets, including randomized rigid transformations and non-Gaussian noise injections into one of the 3D point sets. Qualitative results are visualized in Fig. 1, where the red colored point set denotes the source and the green colored point set denotes the target. In Section V, we perform a quantitative evaluation of our approach against several baselines and advanced methods.

Our contributions are summarized as follows: We introduce

Corresponding author: Hung La, E-mail: hla@unr.edu

The authors are with the Advanced Robotics and Automation (ARA) Laboratory, Department of Computer Science and Engineering, University of Nevada, Reno, NV 89557, USA
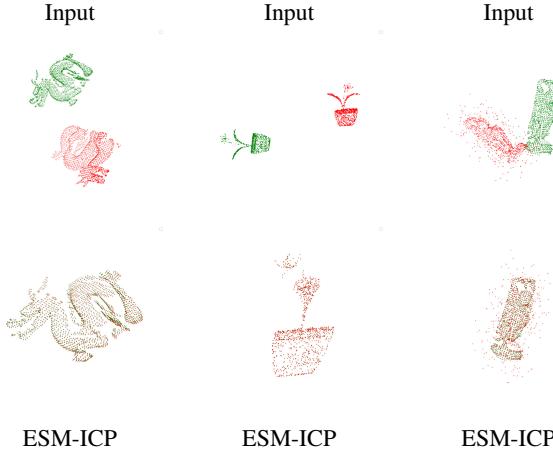
Fig. 1: Registration results using ESM-ICP. Top row: input source and target pairs. Bottom row: aligned outputs after applying ESM-ICP.

ESM-ICP, a robust registration method that incorporates a Gaussian-inspired similarity matrix within the ICP framework. We evaluate ESM-ICP under large transformation scenarios, using random rotations sampled from $[-\pi, \pi]$ and translations across a wide range. We demonstrate improved accuracy over competing methods across multiple datasets and scenarios. We evaluate performance in the presence of outliers and non-Gaussian noise, where our method consistently outperforms traditional and deep learning-based approaches. We publicly release our implementation source codes for reproducibility and community benchmarking.

II. RELATED WORK

The earliest solution to the point cloud registration problem is the classic Iterative Closest Point (ICP) algorithm [1], [3], [58], [45]. ICP iteratively matches corresponding points between two 3D point sets and solves a least-squares problem to refine the transformation. The rotation is computed using Singular Value Decomposition (SVD), while the translation is determined by the difference between the centroids of the point sets. However, traditional ICP struggles with large initial misalignments, is highly sensitive to the initial pose, and is prone to getting trapped in local minima [36].

To overcome these limitations, numerous variants of ICP have been developed [58], [54]. For instance, Point-to-Plane ICP [8] incorporates surface normal information from the target point cloud, which accelerates convergence and improves accuracy on smooth surfaces by minimizing the error along the target normal direction. While more efficient than classic ICP, it remains sensitive to large rotations and noise. Generalized ICP (GICP) [38] addresses this by modeling each point as a Gaussian distribution and using local surface covariances to minimize the Mahalanobis distance between point pairs. This enables alignment based on matching ellipsoidal uncertainty regions rather than simple points or planes.

ICP has also been formulated as an optimization problem, solvable via algorithms like the Levenberg–Marquardt method [37]. Additional robust variants employ outlier trimming techniques to enhance performance in the presence of noise. Beyond ICP-style algorithms, branch-and-bound (BnB) methods like GO-ICP [10] explore the full SE(3) motion space for global optimization. Although more computationally intensive, these methods offer improved robustness and a higher likelihood of finding the global optimum.

Beyond ICP, probabilistic methods such as the Normalized Distribution Transform (NDT) [25] have been proposed. NDT models local point distributions within voxel grids, eliminating the need for explicit point correspondences and often achieving faster and more accurate registration - though it, too, struggles with large rotations. Another category of approaches reduces registration complexity using four-point congruent sets, which significantly decrease the spatial search space and improve robustness to noise and outliers [15], [23], [22].

Feature-based approaches like RANSAC are also widely adopted, often inspired by 2D image registration techniques like SIFT [16], [21]. In 3D, descriptors such as Fast Point Feature Histograms (FPFH) [20] are used to extract keypoints and estimate transformations. Fast Global Registration (FGR) [18] is one such example. Similarly, Sample Consensus Initial Alignment (SAC-IA) [17] uses feature correspondences for coarse alignment, followed by point-to-plane ICP refinement. These methods rely heavily on the quality of the feature descriptors used.

With the rise of deep learning, numerous AI-driven registration techniques have emerged. PointNet [49] was the first neural network to directly process unordered point clouds. While widely used in classification, detection, and segmentation tasks, it also influences registration. PointNet captures global shape features but lacks local geometric context. This issue is addressed by PointNet++ [48], which aggregates local features using K-nearest neighbor clusters. Another variant, PointNetLK [51], integrates the Lucas–Kanade algorithm into PointNet's learned feature space for alignment. More advanced models like DGCNN (Dynamic Graph CNN) [50] dynamically construct graphs in both spatial and feature space to extract richer representations.

Deep Closest Point (DCP) [47] builds upon PointNet by extracting meaningful point-wise feature descriptors and uses attention mechanisms to establish soft correspondences between source and target. Once correspondences are formed, it estimates the transformation using a differentiable SVD layer. While DCP [47] is robust to noise and outliers, it often requires supervised data and pre-trained models.

A related approach, RPM-Net (Robust Point Matching using Learned Features) [46], also employs soft correspondence matching through a differentiable Sinkhorn layer with annealing. This improves resilience to noise and partial visibility. It further enhances performance by using a secondary network to predict the optimal annealing parameters during registration.

Despite the variety of available registration algorithms, point cloud alignment can still fail, particularly under conditions of large rotational transformations between source and target

point sets. In light of this challenge, we propose a novel variant of the ICP algorithm that performs weighted matching using our proposed similarity matrix. We compare our method to both classical numerical approaches and deep learning-based techniques. Through experimental results, we demonstrate how our approach can effectively resolve these registration challenges.

The remainder of this paper is structured as follows: Section III introduces the background of ICP using SVD and its associated problems. Section IV presents our proposed method and its implementation. Section V provides experimental results and evaluation. Section VI concludes the paper and outlines future directions.

III. BACKGROUND

In this section, we discuss the basic ICP problem and highlight the key issues associated with it. As mentioned earlier, for rigid body alignment, we have two 3D point sets: one model (***Source***) and one template (***Target***). We denote the ***Source*** as $\mathbf{S} = \{\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3, \ldots, \mathbf{s}_N\} \subset \mathbb{R}^3$, and the ***Target*** as $\mathbf{T} = \{\mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3, \ldots, \mathbf{t}_N\} \subset \mathbb{R}^3$, separated by a random rigid transformation consisting of a rotation matrix $\mathbf{R} \in \mathrm{SO}(3)$ and a translation vector $\boldsymbol{\tau} \in \mathbb{R}^3$. For simplicity, we assume that the ***Source*** and ***Target*** point sets contain the same number of 3D points. The vectors $\mathbf{s}_i$ and $\mathbf{t}_i$ represent the 3D coordinates of the Source and Target point clouds, respectively, and are indexed by $i = 1, 2, \ldots, N$. For the remainder of this paper, we denote matrices by bold capital letters, vectors by bold lowercase letters, and scalars by regular (non-bold) fonts.

For the rigid body alignment problem, we intend to minimize the mean-squared error $E(\mathbf{R}, \boldsymbol{\tau})$, which can be written as:

$$E(\mathbf{R}, \boldsymbol{\tau}) = \frac{1}{N} \sum_{i}^{N} \|\mathbf{R}\mathbf{s}_i + \boldsymbol{\tau} - \mathbf{t}_i\|^2. \tag{1}$$

To achieve this minimization, we first compute the centroid of both the source and the target point sets as:

$$\bar{\mathbf{s}} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{s}_i \quad \text{and} \quad \bar{\mathbf{t}} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{t}_i. \tag{2}$$

Then we compute the cross-covariance matrix $\mathbf{H}$ given as:

$$\mathbf{H} = \sum_{i=1}^{N} (\mathbf{s}_i - \bar{\mathbf{s}})(\mathbf{t}_i - \bar{\mathbf{t}}). \tag{3}$$

Performing SVD on this covariance matrix $\mathbf{H}$ can yield the rotation component. So, if $\mathbf{H} = \mathbf{U}\mathbf{D}\mathbf{V}^T$, then the rotation and the translation component are given by,

$$\mathbf{R} = \mathbf{V}\mathbf{U}^T \quad \text{and} \quad \boldsymbol{\tau} = -\mathbf{R}\bar{\mathbf{s}} + \bar{\mathbf{t}}. \tag{4}$$

It can be interpreted that $\mathbf{V}, \mathbf{U} \in \mathrm{SO}(3)$. $\mathbf{D}$ is diagonal matrix. Here, the Procrustes problem assumes that all point sets are matched from Source to Target point sets, that is, $\mathbf{s}_i$ to $\mathbf{t}_i$ after performing the final alignment for all i. If the correspondences

are unknown, then the objective function from Equ. 1 can be rewritten as:

$$E(\mathbf{R}, \boldsymbol{\tau}) = \frac{1}{N} \sum_{i}^{N} \|\mathbf{R}\mathbf{s}_i + \boldsymbol{\tau} - \mathbf{t}_{m(s_i)}\|^2, \tag{5}$$

where m denotes the mapping from each point in $\mathbf{S}$ to its corresponding points in $\mathbf{T}$. However, it can be easily seen that if we know the optimal rigid transformation, then the mapping m can be recovered, and conversely, given an optimal mapping m transformation can be computed using Equ. 4. It's a well known *chicken-egg* problem. The algorithm terminates when a certain criterion or threshold is reached, however, is extremely prone to local optima. If the initial alignment is far, it may get stuck, yielding a poor estimate m. We try to address this problem in a fairly intuitive way, where we pick only a few points based on a given criterion in each iteration, and we see how it can achieve global registration as the iteration increases.

IV. WEIGHTED CROSS-COVARIANCE MATRIX

As it is clear from the standard ICP framework that the cross-covariance matrix, referenced in Equation 3 - is computed using pairs of points in the source and target point sets after centering both with respect to their respective centroids. This conventional approach assumes a hard correspondence: each source point is matched to a single closest target point, and all point pairs contribute equally to the alignment.

In contrast, our method introduces a significant generalization by computing a weighted cross-covariance matrix, which is iteratively refined as the algorithm progresses. Instead of assuming binary correspondences, we adopt a soft correspondence model that evaluates the similarity between each point in the source set and its paired point in the target set using a Gaussian kernel. This results in a similarity matrix that modulates the influence of each correspondence based on geometric proximity.

Specifically, we define a Gaussian kernel function as:

$$f(x) = \exp\left(-\frac{x^2}{2\sigma^2}\right), \tag{6}$$

where x is the Euclidean distance between a pair of corresponding points, and $\sigma > 0$ is a user-defined scale parameter that controls the sensitivity of the similarity measure. This kernel ensures that correspondences with smaller distances are weighted more heavily, while those with larger discrepancies are exponentially suppressed. As a result, outliers and poor matches contribute minimally to the alignment process, increasing the robustness of the algorithm in the presence of noise, occlusions, or non-rigid deformations.

This weighted formulation not only generalizes the classical ICP's assumption of equal-weighted correspondences but also provides a smooth and differentiable measure of alignment quality, which is particularly advantageous when dealing with ambiguous or noisy data. The weighted cross-covariance matrix thus encapsulates both geometric structure and probabilistic confidence in the pairwise alignments, leading to more accurate and stable transformation estimates.

From the Gaussian Kernel function in Equ. 6, we define our alignment problem as a weighted Procrustes problem, where the weights for each corresponding pair from the source to the target point sets are defined as:

$$w_i = \exp\left(-\frac{\|\mathbf{s}_i - \mathbf{t}_{c(i)}\|^2}{2\sigma^2}\right), \tag{7}$$

where $c(i)$ is the index of the corresponding points from each point in the source point set to the target point set. These correspondences, as mentioned, are the closest points based on Euclidean distance. Later, we construct a symmetric similarity matrix $\mathbf{M}$ of size $N \times N$ such that:

$$\mathbf{M}(i, c(i)) = \mathbf{M}(c(i), i) = w_i. \tag{8}$$

One may note that our similarity measurement matrix $\mathbf{M}$ is symmetric, i.e., $\mathbf{M} = \mathbf{M}^\top$, where $\top$ denotes the matrix transpose. Based on the above definition, we compute the $\mathbf{H}$ matrix as:

$$\mathbf{H} = \sum_{i=1}^{N}(\mathbf{s}_i - \bar{\mathbf{s}})\mathbf{M}(\mathbf{t}_i - \bar{\mathbf{t}}). \tag{9}$$

Performing SVD as equivalent to the traditional ICP on this covariance matrix $\mathbf{H}$ can yield the rotation component. So, if $\mathbf{H} = \mathbf{U}\mathbf{D}\mathbf{V}^T$, then the rotation and the translation component is given by,

$$\mathbf{R} = \mathbf{V}\mathbf{U}^T \quad \text{and} \quad \boldsymbol{\tau} = -\mathbf{R}\bar{\mathbf{s}} + \bar{\mathbf{t}}. \tag{10}$$

Based on the above formulation, we arrive at minimizing the function:

$$\min_{\mathbf{R} \in \mathrm{SO}(3)} \sum_{i=1}^{N} w_i \|\mathbf{R}\mathbf{s}_i' - \mathbf{t}_i'\|^2, \tag{11}$$

and the weighted alignment error at iteration k is:

$$E_k = \sum_{i=1}^{N} w_i^{(k)} \|\mathbf{R}_k\mathbf{s}_i + \boldsymbol{\tau}_k - \mathbf{t}_i\|^2. \tag{12}$$

Note that the similarity weights w_i are bounded ($0 < w_i < 1$). After every iteration, the computed transformation $\mathbf{A}_k$ is the minimizer of E_k so $E_{k+1} \leq E_k$. Here $\mathbf{A}_k$ can be written as :

$$\mathbf{A}_k = \begin{bmatrix} \mathbf{R}_k & \boldsymbol{\tau}_k \\ \mathbf{0}^\top & 1 \end{bmatrix}. \tag{13}$$

A. Convergence Analysis of ESM-ICP

Theorem 1. *Let $\mathbf{A}_k$ be the sequence of rigid transformations computed by ESM-ICP over iterations. If the weights w_i are bounded and continuous (it's guaranteed since $0 < w_i < 1$), and the similarity matrix remains well-conditioned, then the sequence converges $\mathbf{A}_k$ to a fixed transformation $\mathbf{A}^*$.*

Proof. At each iteration, the objective function minimized is:

$$E_k = \sum_{i=1}^{N} w_i^{(k)} \|\mathbf{R}_k\mathbf{s}_i + \boldsymbol{\tau}_k - \mathbf{t}_i\|^2. \tag{14}$$

This is a weighted least-squares error where weights $w_i^{(k)} \in (0, 1]$ are derived from the Gaussian kernel. As the transformation is refined, the distances $\|\mathbf{s}_i - \mathbf{t}_{c(i)}\|$ decrease, stabilizing the weights. The updated transformation $\mathbf{A}_{k+1}$ minimizes , leading to:

$$E_{k+1} < E_k. \tag{15}$$

The sequence E_k is thus monotonically decreasing and bounded below, implying convergence. Furthermore, the transformation updates satisfy $\|\mathbf{A}_{k+1} - \mathbf{A}_k\|$, yielding convergence to a fixed point $\mathbf{A}^*$. □

Corollary 1. *Let $\mathbf{M}$ be the similarity matrix used in ESM-ICP. After k iterations, if the source and target point sets are sufficiently close such that the computed weights satisfy $w_i \to 1$, the ESM-ICP formulation effectively reduces to the standard ICP algorithm.*

Proof. Suppose that after k iterations, the weights w_i approach 1 for all points, as governed by the exponential weighting function. This implies the emergence of near-perfect one-to-one correspondences between the source and target point sets. Consequently, according to Equ. 8, the similarity matrix $\mathbf{M}$ asymptotically becomes a diagonal matrix with $w_i \to 1$, effectively reducing $\mathbf{M}$ to the identity matrix.

Substituting $\mathbf{M} = \mathbf{I}$ into Equ. 9 yields the original cross-covariance formulation in Equ. 3, which corresponds to the standard ICP algorithm. □

Our proposed approach is algorithmically described in Algorithm 1.

Algorithm 1 ESM-ICP Algorithm

1: **function** READDATASETS($\mathbf{S}, \mathbf{T}$)
2:     **while** not converged **do**
3:         $c \leftarrow$ **ComputeCorrespondence**($\mathbf{S}, \mathbf{T}$)
4:         $\mathbf{M} \leftarrow \mathrm{zeros}(N, N)$.
5:         **for** $i = 1$ to N **do**
6:             $\mathbf{d} \leftarrow \|\mathbf{s}_i - \mathbf{t}_{c(i)}\|$.
7:             $w_i \leftarrow \exp\left(-\frac{\|\mathbf{s}_i - \mathbf{t}_{c(i)}\|^2}{2\sigma^2}\right)$.
8:             $\mathbf{M}(i, c(i)) = w_i, \quad \mathbf{M}(c(i), i) = w_i$.
9:         **end for**
10:        $\bar{\mathbf{s}} \leftarrow \frac{1}{N}\sum \mathbf{s}_i, \quad \bar{\mathbf{t}} \leftarrow \frac{1}{N}\sum \mathbf{t}_i$.
11:        $\mathbf{H} \leftarrow \sum(\mathbf{s}_i - \bar{\mathbf{s}})\mathbf{M}(\mathbf{t}_i - \bar{\mathbf{t}})$.
12:        SVD: $\mathbf{H} = \mathbf{U}\mathbf{D}\mathbf{V}^\top$.
13:        $\mathbf{R} \leftarrow \mathbf{V}\mathbf{U}^\top$.
14:        $\boldsymbol{\tau} \leftarrow -\mathbf{R}\bar{\mathbf{s}} + \bar{\mathbf{t}}$.
15:     **end while**
16: **end function**

V. EXPERIMENTS

The effectiveness of the proposed ESM-ICP algorithm is assessed using two widely recognized benchmark datasets: the Stanford Bunny [9] and ModelNet40 [57]. While both datasets are utilized, the majority of experiments are conducted on ModelNet40 due to its diversity and scale. Within our experimental pipeline, point cloud data is loaded in *pcd*

format. A random rigid transformation is applied to one of the point clouds, which is subsequently designated as the *Source*, while the unaltered point cloud serves as the *Target*. The objective is to accurately estimate the transformation that aligns the Source to the Target using our ESM-ICP method and to quantitatively evaluate the quality of this alignment. We benchmark the performance of ESM-ICP against several baseline and state-of-the-art registration algorithms, including traditional geometry-based methods such as standard ICP [1], [3], ICP Point-to-Plane [8], Go-ICP [10], and ICP Non-Linear [40], [42], as well as learning-based methods like Deep Closest Point (DCP) [47], PointNetLK [51], RpmNet [31], and DeepGMR [55]. To assess robustness, we also introduce outliers into the Source point cloud and analyze the algorithm's performance under such perturbations. Evaluation is primarily conducted using three error metrics—Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Mean Square Error (MSE)—calculated between the predicted and ground-truth transformations. Successful rigid alignment is characterized by RMSE values approaching zero. All experiments are performed on a system equipped with an Intel i7 CPU (2.70 GHz) and 32 GB of RAM. For deep learning-based approaches, we leverage an NVIDIA GTX 1070 GPU with 8 GB of VRAM.

To explore the internal behavior of our algorithm, we initially focus on the Stanford Bunny dataset. In this setting, we analyze the evolution of the similarity matrix $\mathbf{M}$ over successive iterations to illustrate how $\mathbf{M}$ progressively approaches an identity matrix as the algorithm converges.

A. Evolution of $\mathbf{M}$

As an initial evaluation, we utilize the well-known Stanford Bunny dataset to analyze the behavior of our algorithm in a controlled setting. To simplify visualization and enhance interpretability, we downsample the point cloud using a voxel grid filter with a leaf size of 0.01, implemented via the Point Cloud Library (PCL v1.15) [37]. The reduced number of points allows for clearer observation of the alignment process between the source and target point sets.

A random rigid transformation with a large rotational component is applied to the source point cloud. Specifically, the rotation is defined by Euler angles $(r, p, y) = (0, 0, 4.53)$, while the translation vector is sampled uniformly in the range $[-0.1, 0.1]$. It is important to note that, due to the centroid-based nature of our transformation computation, the absolute translation offset has minimal influence on convergence. The focus remains on recovering accurate rotational alignment despite significant initial misalignment. Figure 2(a) depicts the initial separation between the source and target point sets, while Figures 2(b) and (c) demonstrate the convergence of the algorithm over 14 and 15 iterations, respectively.

For all experiments, we range the value of $\sigma = 0.05, 1.0$ in our exponential similarity weighting function depending on the point set density. For example, if the point set is dense (captured from a high fidelity sensor), we can set the value of sigma to be small, and large otherwise. This parameter governs

the influence of point-to-point distances on the similarity matrix and is crucial in moderating the robustness of the algorithm to noise and outliers. To analyze the internal mechanics
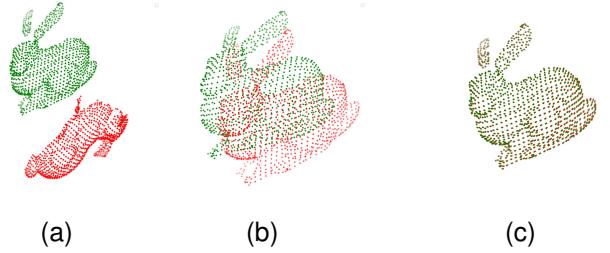

Fig. 2: (a) shows the 2 bunny rabbit point cloud (Red: Source and Green: Target) down-sampled using voxel grid filtering of leaf size 0.01. The transformation between them is $(r, p, y) = (0, 0, 4.53)$ and $(\tau_x, \tau_y, \tau_z) = (0, 0, 0.3)$ (b) shows the alignment after 14 iterations. (c) shows the alignment after 15 iterations.

of our approach, particularly the evolution of the similarity matrix $\mathbf{M}$, we track its values across iterations. As outlined earlier, $\mathbf{M}$ is reinitialized to a zero matrix at each iteration, and its entries are updated based on the exponential weighting of current point correspondences. Figure 3 visualizes how the structure of $\mathbf{M}$ evolves, illustrating a gradual increase in one-to-one correspondences as the alignment progresses.

To make this matrix more interpretable, we conduct the same experiment using a heavily downsampled version of the Bunny dataset with a voxel leaf size of 0.06. This results in just 21 points, generating a 21×21 similarity matrix that is easier to visualize and analyze. From the heatmap in Fig. 3, it is evident that only correspondences with substantial weights (highlighted in yellow) significantly contribute to the transformation computation through matrix $\mathbf{H}$. The remaining elements (shown in blue) have near-zero values and are effectively ignored.

As iterations proceed, the structure of $\mathbf{M}$ begins to resemble the identity matrix, indicating a strong one-to-one correspondence between source and target points. Once convergence is close enough, the algorithm transitions into a behavior similar to classical ICP [1], [3]. Notably, the final RMSE achieved in this test was 0.000837512 after 15 iterations, highlighting the high precision of the recovered alignment.

B. Comparison to other methods

The comparative analysis is extended to include a range of geometric registration techniques, such as standard Iterative Closest Point (ICP [1], [3]), ICP Point-to-Plane [8], ICP Non-Linear (ICP-NL) [40], [42], and the globally optimal Go-ICP [10] algorithm. All of these methods are categorized as geometry-based alignment approaches. For evaluation, we use the widely adopted ModelNet40 dataset, applying random rigid transformations to simulate diverse registration challenges. The transformation parameters include rotation
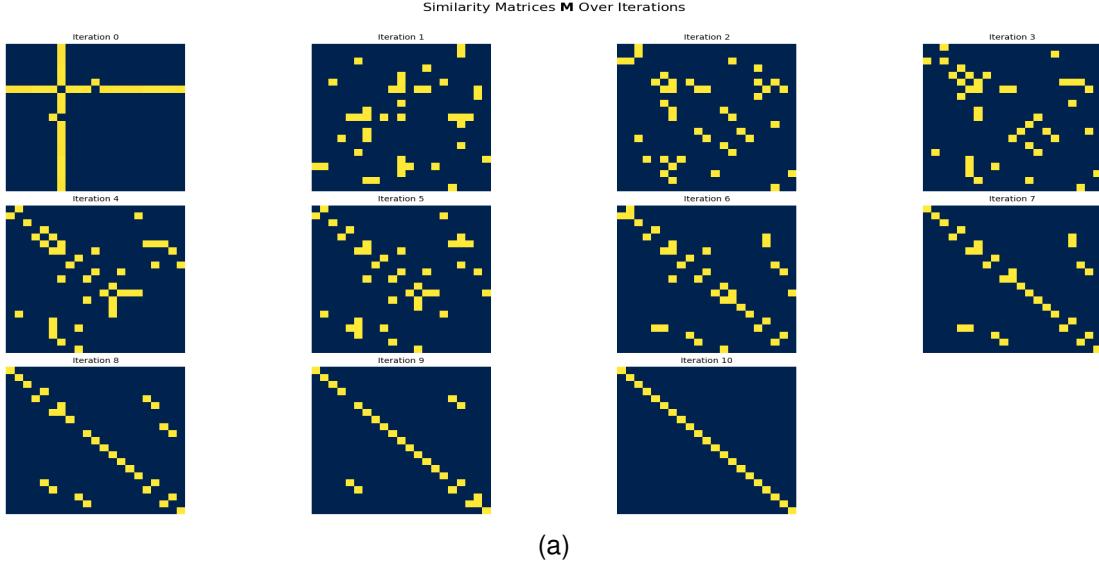
(a)

Fig. 3: (a) Evolution of the similarity matrix $\mathbf{M}$ (21×21) across iterations. (Note: the bunny rabbit point cloud dataset was sampled with a leaf size of 0.06, reducing the matrix size for better visualization purposes.) Yellow entries denote high-weight correspondences; blue entries indicate near-zero weights.

angles sampled uniformly from the range $[-\pi, \pi]$ and translations drawn from both $[-1.0, 1.0]$ and a wider interval of $[-100, 100]$ for more extreme tests.

Importantly, no additional pre-processing steps—such as voxel grid filtering, normal estimation, or downsampling—are applied to the ModelNet40 dataset. This decision was made to assess the robustness of the registration algorithms on raw, unfiltered point cloud data. Additionally, we omit runtime comparisons, as the computational performance of our proposed ESM-ICP algorithm is similar to standard ICP [1], [3], [1] and remains suitable for real-time applications.

Figure 4 provides a visual comparison of ESM-ICP against the aforementioned geometric methods. While Go-ICP [10] is known for its robustness to large rotations and occasionally demonstrates strong performance, we observed failure cases in certain complex configurations. In contrast, our method consistently succeeds across a wide range of transformation scenarios. Specifically, we evaluated our approach on over 2,000 randomly sampled rotation combinations within $[-\pi, \pi]$ and found that ESM-ICP successfully aligned all test pairs within 100 iterations.

Traditional methods, including standard ICP [1], [3], ICP-NL and ICP Point-to-Plane [8], frequently failed in our evaluation, particularly in scenarios involving large rotational misalignments. These failures align with known limitations of such techniques, which often struggle without a good initial alignment guess.

In parallel, we also compare ESM-ICP with several deep learning-based registration approaches. These include Deep Closest Point (DCP [47]), PointNetLK [51], RpmNet [31], and DeepGMR [55], all evaluated using their publicly available pre-trained models provided by the Learning3D library. Al-

though these models were trained on the ModelNet40 dataset, they typically follow a setup where the testing set differs from the training set. For example, DCP [47] applies random rigid transformations during training, drawing rotation angles from the interval $[0°, 45°]$ and translations from $[-0.5, 0.5]$. However, since our evaluation often involves rotations significantly exceeding $45°$, these models—including RpmNet [31] and DeepGMR [55]—exhibited degraded performance and often failed to produce accurate alignments.

As shown in Fig. 5, ESM-ICP outperforms all evaluated deep learning models under a wide range of transformation conditions. Quantitative results are presented in Table I for the Airplane class and in Table II for the Person class. Both illustrate the superior accuracy of ESM-ICP.

In contrast to deep learning models, Go-ICP [10] shows comparatively better performance under large rotations, however, it too can fail in scenarios with complex and coupled axis rotations. Moreover, Go-ICP [10], like most methods, struggles when the source point cloud is contaminated by noise.

To further assess robustness, we also evaluate registration performance in the presence of noise. Table IV and Table III show results when outliers are introduced in the source point cloud. Noise is injected post-transformation using a mixture of Gaussian distributions: $\mathcal{N}(0, 0.01)$ clipped to $[-0.05, 0.05]$, $\mathcal{N}(0, 0.04)$ clipped to $[-1.0, 1.0]$, and $\mathcal{N}(0, 0.1)$ clipped to $[-10, 10]$, effectively modeling non-Gaussian behavior. In these challenging scenarios, most traditional and learning-based methods fail to recover accurate transformations.

In contrast, our proposed ESM-ICP method demonstrates strong robustness against such distortions. This is attributed to our similarity matrix $\mathbf{M}$, which leverages an exponential Gaussian weighting strategy. This mechanism reduces the influ-
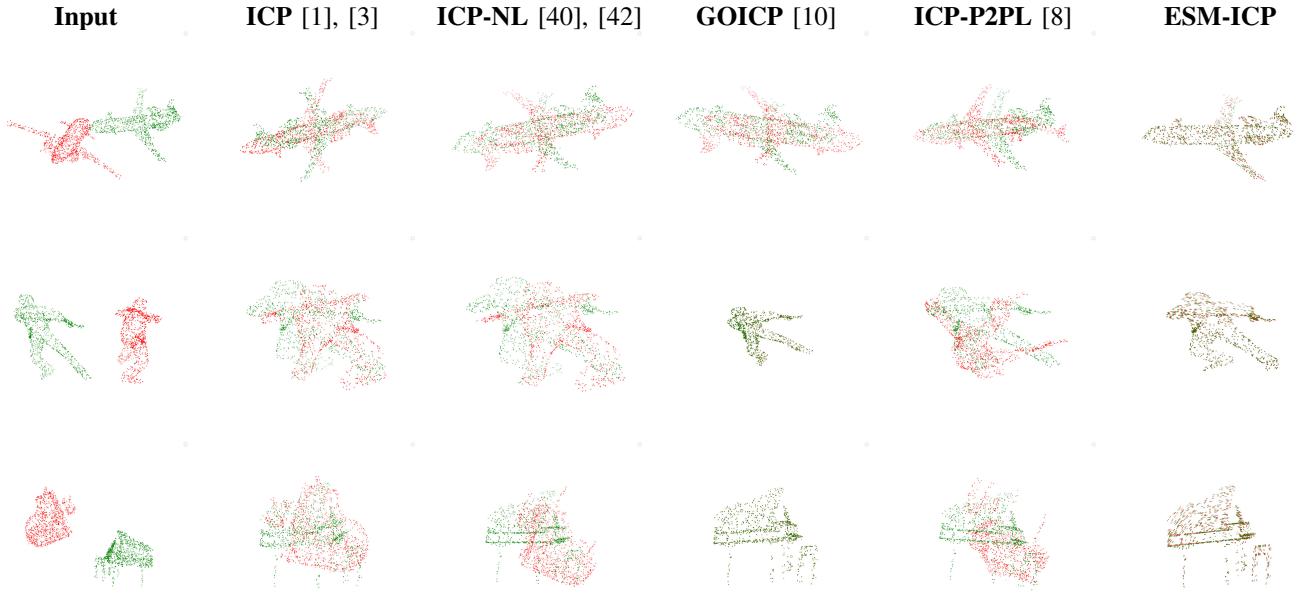
| **Input** | **ICP** [1], [3] | **ICP-NL** [40], [42] | **GOICP** [10] | **ICP-P2PL** [8] | **ESM-ICP** |

Fig. 4: Comparison of ESM-ICP alignment to other geometric-based approaches.

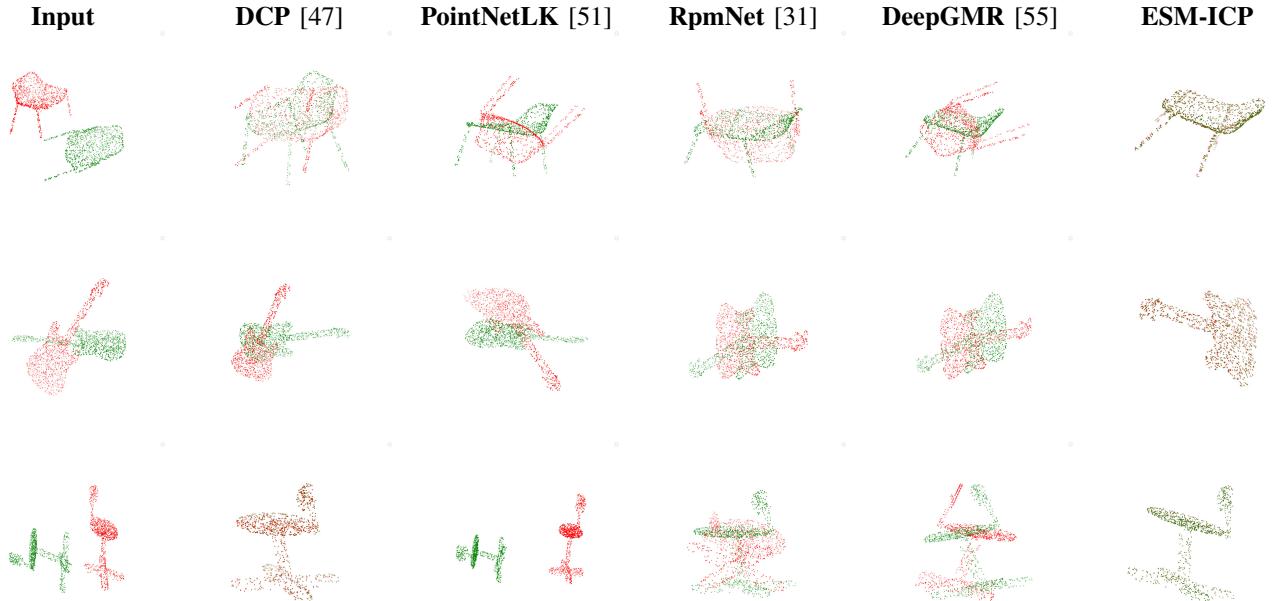| **Input** | **DCP** [47] | **PointNetLK** [51] | **RpmNet** [31] | **DeepGMR** [55] | **ESM-ICP** |

Fig. 5: Comparison to deep learning methods.

ence of outlier correspondences during optimization, thereby enhancing alignment accuracy. The consistent performance of ESM-ICP across both clean and noisy data strongly highlights its effectiveness for real-world registration applications.

C. Effect of σ.

As highlighted in the previous section, our proposed method consistently demonstrates strong performance across various ModelNet40 datasets when the parameter σ is set to values between 0.05 and 1.0. This observation suggests that the effectiveness of σ is inherently linked to the density of the point

TABLE I: ModelNet40: Evaluation on Airplane dataset with random transformation matrix $(r, p, y, \tau_x, \tau_y, \tau_z) = (2.58175, -2.16765, 0.711617, 11.0585, 16.2285, -17.0047)$ after 40 iterations.

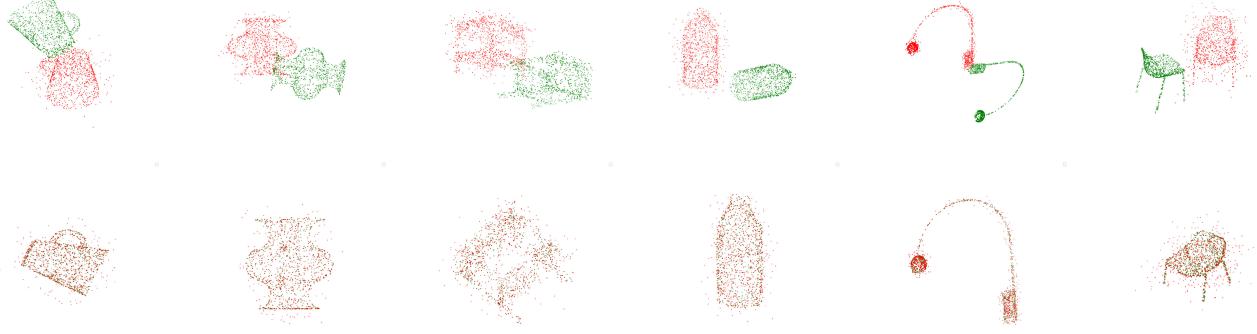| Model:Airplane	**MSE**(R)	**RMSE**(R)	**MAE**(R)	**MSE**(t)	**RMSE**(t)	**MAE**(t)
ICP [1], [3]	0.888774	0.942748	0.722478	444942	667.04	0.722478
ICP P2PL [8]	0.887952	0.942312	0.627701	396629	629.785	535.349
ICP NL [40], [42]	0.861814	0.928339	0.725019	227.584	15.0859	14.7941
GO-ICP [10]	0.711808	0.843687	0.701879	227.839	15.0943	14.743
RpmNET [31]	0.033537	0.183130	0.151623	14822.6	121.748	107.798
PointNetLK [51]	0.883079	0.939723	0.721705	321.678	17.9354	17.0547
DCP [47]	0.537342	0.733036	0.629785	330673	575.042	469.834
ESM-ICP (ours)	**4.749e-16**	**2.179e-08**	**1.076e-08**	**7.549e-11**	**8.688e-06**	**7.947e-06**

Fig. 6: Performance in the presence of outliers. The top row denotes the input, and the bottom row denotes the output from ESM-ICP.

TABLE II: ModelNet40: Evaluation on Person dataset with random transformation matrix $(r, p, y, \tau_x, \tau_y, \tau_z) = (-2.56814, 1.15753, 1.92252, 0.0635863, 0.0883894, 0.0158466)$ after 40 iterations.

Model:	MSE(R)	RMSE(R)	MAE(R)	MSE(t)	RMSE(t)	MAE(t)
ICP [1], [3]	0.888147	0.942415	0.811008	1823.66	42.7044	25.7527
ICP P2PL [8]	0.708472	0.841708	0.721835	1431.99	37.8417	29.8622
ICP NL [40], [42]	0.812057	0.901142	0.808517	0.0107494	0.103679	0.808517
GO-ICP [10]	8.95827e-14	2.99304e-07	2.16895e-07	5.11454e-10	2.26154e-05	1.4035e-05
RpmNET [31]	0.866954	0.931104	0.783011	969.146	31.1311	28.6128
PointNetLK [51]	0.885005	0.940747	0.805605	65.5633	8.09712	7.25109
DCP [47]	0.784646	0.885803	0.666492	3292.4	57.3795	51.1154
ESM-ICP (ours)	**2.46716e-16**	**1.57072e-08**	**9.93411e-09**	**1.19354e-12**	**1.09249e-06**	**9.10833e-07**

TABLE III: ModelNet40: Evaluation on Vase dataset (with 30% of the data affected with outliers) with random transformation matrix $(r, p, y, \tau_x, \tau_y, \tau_z) = (-2.51358, 0.916596, 1.10051, 105.383, -53.3576, -16.489)$ after 40 iterations.

Model:	MSE(R)	RMSE(R)	MAE(R)	MSE(t)	RMSE(t)	MAE(t)
ICP [1], [3]	0.864082	0.92956	0.794351	23136.8	152.108	125.499
ICP P2PL [8]	0.438943	0.662528	0.50979	24721.3	157.23	135.133
ICP NL [40], [42]	0.888659	0.942687	0.813719	4741.5	68.8585	58.4099
GO-ICP [10]	0.0755686	0.274897	0.240074	4173.73	64.6044	56.6411
RpmNET [31]	0.887304	0.941968	0.8128	16041.9	126.657	112.902
PointNetLK [51]	0.882742	0.939544	0.738511	4548.29	67.441	60.0684
DCP [47]	0.888659	0.942687	0.813719	84140.6	290.07	245.192
ESM-ICP (ours)	**2.26497e-07**	**0.000475917**	**0.000384784**	**0.13024**	**0.360888**	**0.356485**

TABLE IV: ModelNet40: Evaluation on Bottle dataset (with 60% of the data affected with outliers) with random transformation matrix $(r, p, y, \tau_x, \tau_y, \tau_z) = (2.82862, -0.905772, 1.88224, -62.3639, -31.1694, -56.1162)$ after 40 iterations.

Model:	MSE(R)	RMSE(R)	MAE(R)	MSE(t)	RMSE(t)	MAE(t)
ICP [1], [3]	0.888844	0.942785	0.839232	15541	124.663	110.264
ICP P2PL [8]	0.888172	0.942429	0.842185	12393.2	111.325	80.3513
ICP NL [40], [42]	0.826324	0.909024	0.909024	2648.4	51.4626	49.6877
GO-ICP [10]	0.130485	0.361228	0.309347	2777.48	52.7018	51.1383
RpmNET [31]	0.679944	0.824587	0.681382	5850.54	76.4888	60.0398
PointNetLK [51]	0.885858	0.9412	0.735128	2695.98	51.9228	50.5983
DCP [47]	0.825616	0.908634	0.792235	11491	107.196	81.2461
ESM-ICP (ours)	**1.90873e-05**	**0.00436891**	**0.00354401**	**0.220783**	**0.469875**	**0.424405**

cloud data. For point clouds with higher spatial resolution, a smaller σ tends to be sufficient, whereas lower-density data may benefit from a larger σ to preserve alignment stability. Identifying an adaptive or data-driven strategy to select the optimal σ for different sensing modalities and application domains remains a promising direction for future research.

D. Comparison with KISS-Matcher

During the final stages of this work, we became aware of another algorithm for point set registration known as KISS-Matcher [61]. KISS-Matcher [61] introduces a novel feature detector, Faster-FPFH, and incorporates a k-core based graph-theoretic pruning strategy to efficiently reject outlier correspondences and reduce computational complexity. Recognizing its potential, we evaluated KISS-Matcher [61] under random transformations and observed that it exhibited strong robustness to large rotations, comparable to our own results. To further strengthen our validation, we conducted a comparative analysis between our method and KISS-Matcher [61] across the ModelNet40 datasets (airplane, person, and stool).

We performed regression testing by loading point clouds as *Source* and applying approximately 1000 random transformations — with rotations sampled from $[-\pi, \pi]$ and translations from $[-1000, 1000]$ — to generate the corresponding *Target*, consistent with our prior experiments. For our method, we capped the iteration count at 100. Across these various transformations, both KISS-Matcher [61] and our proposed ESM-ICP method demonstrated successful convergence. However, when comparing the mean error metrics for both rotation and translation components, ESM-ICP consistently outperformed KISS-Matcher [61]. Table V presents the quantitative comparison of the two methods. Furthermore, ESM-ICP achieved significantly superior performance in scenarios with outliers, as detailed in Table VI.

TABLE V: Comparative analysis of ESM-ICP and KISS-Matcher [61] across 1000 different transformations ranging from $[-\pi, \pi]$ rotation wise and $[-1000, 1000]$ translation wise. The metrics are computed as mean of 1000 different transformations.

Model:	MSE(R)	RMSE(R)	MAE(R)	MSE(t)	RMSE(t)	MAE(t)
Airplane (KISS-Matcher) [61]	3.77e-09	0.000007	0.000005	0.001126	0.003973	0.003349
Person (KISS-Matcher) [61]	0.000231	0.000507	0.000428	0.930943	0.032311	0.023097
Stool (KISS-Matcher) [61]	0.000004	0.001152	0.000936	0.000003	0.000975	0.000847
Airplane (**ESM-ICP (ours)**)	2.28e-09	0.000002	0.000001	0.000690	0.000890	0.000821
Person (**ESM-ICP (ours)**)	1.832808e-11	2.424087e-06	1.970518e-06	3.173788e-08	1.074056e-04	9.120257e-05
Stool (**ESM-ICP (ours)**)	1.407338e-07	2.416280e-04	1.955315e-04	8.803378e-08	1.924926e-04	1.623945e-04

TABLE VI: Comparative analysis of ESM-ICP and KISS-Matcher [61] across 1000 different transformations ranging from $[-\pi, \pi]$ rotation wise and $[-1000, 1000]$ translation wise. Here we use the stool dataset and in addition to these random transformation we corrupt 30% of the source dataset with random noise. The target dataset remains unchanged.

Model:	MSE(R)	RMSE(R)	MAE(R)	MSE(t)	RMSE(t)	MAE(t)
Stool (KISS-Matcher) [61]	0.073004	0.127731	0.102708	27542.029047	45.946623	41.252042
Stool (**ESM-ICP (ours)**)	0.073224	0.220009	0.179977	0.040046	0.164954	0.142406

VI. CONCLUSIONS AND FUTURE WORK

One of the core objectives addressed in this work is the accurate estimation of the transformation between the Source and Target point clouds under varying rotations and translations. We evaluated our proposed approach across multiple datasets, where the Source was generated by applying randomly sampled transformation matrices to the Target. The results consistently demonstrate that our method outperforms existing state-of-the-art techniques regarding alignment accuracy.

Furthermore, we tested the robustness of our approach under noise conditions, where the Source was corrupted by randomly injected noise of varying intensity and at arbitrary points. Even in such challenging scenarios, our method maintained superior performance compared to leading alternatives, successfully estimating transformations between the noisy Source and the clean Target.

We strongly encourage readers to explore our implementation, which is publicly available on our GitHub repository. Looking ahead, we intend to extend our method to SLAM applications, leveraging the estimated poses for robot localization and environment mapping for aerial and ground platforms.

ACKNOWLEDGEMENT

REFERENCES

[1] Arun, K. S., Huang, T. S., Blostein, S. D.. "Least-Squares Fitting of Two 3-D Point Sets," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 9, pp. 698–700, 1987.

[2] A. Singandhupe, H. M. La. "A Review of SLAM Techniques and Security in Autonomous Driving," *2019 Third IEEE International Conference on Robotic Computing (IRC)*, pp. 602-607, 2019.

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

[4] "Sehgal, Adarsh. ""Lidar-Monocular Visual Odometry with Genetic Algorithm for Parameter Optimization"," *"Advances in Visual Computing"*, pp. "358–370", "2019".

[5] A. Singandhupe, H. M. La . "MCC-EKF for Autonomous Car Security," *Proceedings of the 4th IEEE International Conference on Robotic Computing (IRC)*, 2020.

[6] A. Singandhupe, H. M. La, D. Feil-Seifer. "Reliable Security Algorithm for Drones Using Individual Characteristics From an EEG Signal," *IEEE Access*, vol. 6, pp. 2–12, 2018.

[7] A. Singandhupe, H. M. La, D. Feil-Seifer, P. Huang, L. Guo, M. Li. "Securing a UAV Using Individual Characteristics From an EEG Signal," *IEEE Intern. Conf. on Systems, Man, and Cybernetics (SMC)*, 2018.

[8] Low, Kok-Lim. "Linear Least-Squares Optimization for Point-to-Plane ICP Surface Registration," , 2004.

[9] "Stanford". ""The Stanford 3D Scanning Repository"," , "2020".

[10] Yang, Jiaolong, Li, Hongdong, Jia, Yunde. "Go-ICP: A globally optimal solution to 3D ICP point-set registration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, pp. 2241–2254, 2015.

[11] "Zongze Wu, Hongchen Chen, Shaoyi Du, Minyue Fu, Nan Zhou, Nanning Zheng". ""Correntropy based scale ICP algorithm for robust point set registration"," *"Pattern Recognition"*, vol. "93", pp. "14 - 24", "2019".

[12] Zhang, Xuetao, Jian, Libo, Xu, Meifeng. "Robust 3D point cloud registration based on bidirectional Maximum Correntropy Criterion," *PloS one*, vol. 13, pp. e0197542, 2018.

[13] Xu, Guanglin, Du, Shaoyi, Cui, Dixiao, Zhang, Sirui, Chen, Badong, Zhang, Xuetao, Xue, Jianru, Gao, Yue. "Precise point set registration using point-to-plane distance and correntropy for LiDAR based localization," *2018 IEEE Intelligent Vehicles Symposium (IV)*, pp. 734–739, 2018.

[14] Segal, Aleksandr, Hähnel, Dirk. "Generalized-ICP," *Proc. of Robotics: Science and Systems*, 2009.

[15] Kuglin, Charles D, Hines, Daniel C. "The phase correlation image alignment method," *Proceedings of the IEEE 1975 International Conference on Cybernetics and Society*, pp. 163–165, 1975.

[16] Lowe, David G. "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, pp. 91–110, 2004.

[17] Rusu, Radu Bogdan, Blodow, Nico, Beetz, Michael. "Fast point feature histograms (FPFH) for 3D registration," *2009 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3212–3217, 2009.

[18] Zhou, Qian-Yi, Park, Jaesik, Koltun, Vladlen. "Fast global registration," *European Conference on Computer Vision (ECCV)*, pp. 766–782, 2016.

[19] Bay, Herbert, Tuytelaars, Tinne, Van Gool, Luc. "SURF: Speeded Up Robust Features," *Computer Vision–ECCV 2006*, pp. 404–417, 2006.

[20] Rusu, Radu Bogdan, Blodow, Norbert, Beetz, Michael. "Fast Point Feature Histograms (FPFH) for 3D registration," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3212–3217, 2009.

[21] Bay, Herbert, Tuytelaars, Tinne, Van Gool, Luc. "SURF: Speeded Up Robust Features," *Computer Vision–ECCV 2006*, pp. 404–417, 2006.

[22] Guizar-Sicairos, Manuel, Thurman, Samuel T, Fienup, James R. "Efficient subpixel image registration algorithms," *Optics Letters*, vol. 33, pp. 156–158, 2008.

[23] Zitova, Barbara, Flusser, Jan. "Image registration methods: a survey," *Image and Vision Computing*, vol. 21, pp. 977–1000, 2003.

[24] Casasent, David, Psaltis, Demetri. "Position, rotation, and scale invariant optical correlation," *Applied Optics*, vol. 15, pp. 1795–1799, 1985.

[25] Biber, P., Strasser, W.. "The normal distributions transform: a new approach to laser scan matching," *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)*, vol. 3, pp. 2743-2748 vol.3, 2003.

[26] Principe, Jose, Iii, John. "Information-Theoretic Learning," *Advances in unsupervised adaptive filtering*, 2000.

[27] Andrew W Fitzgibbon. "Robust registration of 2D and 3D point sets," *Image and Vision Computing*, vol. 21, pp. 1145-1153, 2003.

[28] "Zhou, Qian-Yi. ""Fast Global Registration"," *"Computer Vision – ECCV 2016"*, pp. "766–782", "2016".

[29] Choy, Christopher, Dong, Wei, Koltun, Vladlen. "Deep Global Registration," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[30] Heng Yang, J. Shi, L. Carlone. "TEASER: Fast and Certifiable Point Cloud Registration," *ArXiv*, vol. abs/2001.07715, 2020.

[31] Yew, Zi Jian and Lee, Gim Hee."RPM-Net: Robust Point Matching Using Learned Features," *"Conference on Computer Vision and Pattern Recognition (CVPR)"*, "2020".

[32] Chen, Yang, Medioni, Gerard. "Object modeling by registration of multiple range images," *Proceedings. 1992 IEEE International Conference on Robotics and Automation*, pp. 2724–2729, 1992.

[33] Rusinkiewicz, Szymon, Levoy, Marc. "Efficient variants of the ICP algorithm," *Proceedings Third International Conference on 3-D Digital Imaging and Modeling*, pp. 145–152, 2001.

[34] Zhang, Jiayi, Yao, Yao, Deng, Baoquan. "Fast and Robust Iterative Closest Point," *arXiv preprint arXiv:2007.07627*, 2020.

[35] Shi, Xiaolei, Peng, Ji, Li, Jun. "The iterative closest point registration algorithm based on the normal distribution transformation," *Procedia Computer Science*, vol. 147, pp. 181–190, 2019.

[36] Si, Haiqing, Qiu, Jingxuan, Li, Yao. "A review of point cloud registration algorithms for laser scanners: Applications in large-scale aircraft measurement," *Applied Sciences*, vol. 12, pp. 10247, 2022.

[37] Rusu, Radu Bogdan, Cousins, Steve. "3D is here: Point Cloud Library (PCL)," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1–4, 2011.

[38] Segal, A., Haehnel, D., Thrun, S.. "Generalized-ICP," *Robotics: Science and Systems (RSS)*, 2009.

[39] Huang, Xiaoshui, Mei, Guofeng, Zhang, Jian, Abbas, Rana. "A comprehensive survey on point cloud registration," *arXiv preprint arXiv:2103.02690*, 2021.

[40] Granger, S., Pennec, X.. "Multi-scale EM-ICP: A fast and robust approach for surface registration," *European Conference on Computer Vision (ECCV)*, pp. 418–432, 2002.

[41] Li, Xianzhi, Dai, Yinda, Li, Hongdong, Zhang, Zixin. "Semantic-ICP: Iterative Closest Point for Non-rigid Multi-label Point Cloud Registration," *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12988–12997, 2021.

[42] Zhou, Feng, De la Torre, Fernando. "Robust affine registration method using line/surface normals and correspondence," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, pp. 1051–1064, 2015.

[43] Lin, Junyi, Wang, Cheng, Chen, Huan, Huang, Ruihong, Li, Jun. "KSS-ICP: Point Cloud Registration based on Kendall Shape Space," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 10587–10593, 2022.

[44] Gao, Chenfeng, Wang, Jie, Zheng, Yifan, Chen, Yuxiang, Wang, Fei, Zhang, Jinqiao, Tian, Qi. "DICP: Doppler Iterative Closest Point Algorithm," *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1841–1851, 2023.

[45] Zhang, Yu-Xin, Gui, Jie, Yu, Baosheng, Cong, Xiaofeng, Gong, Xin, Tao, Wenbing, Tao, Dacheng. "Deep learning-based point cloud registration: A comprehensive survey and taxonomy," *arXiv preprint arXiv:2404.13830*, 2024.

[46] Yew, Zac, Lee, Gim Hee. "RPM-Net: Robust point matching using learned features," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11824–11833, 2020.

[47] Wang, Yue, Solomon, Justin M.. "Deep Closest Point: Learning representations for point cloud registration," *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3523–3532, 2019.

[48] Qi, Charles R, Yi, Li, Su, Hao, Guibas, Leonidas J. "PointNet++: Deep hierarchical feature learning on point sets in a metric space," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, pp. 5099–5108, 2017.

[49] Qi, Charles R, Su, Hao, Mo, Kaichun, Guibas, Leonidas J. "PointNet: Deep learning on point sets for 3D classification and segmentation," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 652–660, 2017.

[50] Wang, Yue, Sun, Yongbin, Liu, Ziwei, Sarma, Sanjay E, Bronstein, Michael M, Solomon, Justin M. "Dynamic Graph CNN for Learning on Point Clouds," *Proceedings of the ACM SIGGRAPH Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10206–10215, 2019.

[51] Aoki, Yasuhiro, Goforth, Hunter, Srivatsan, Ramakrishna, Lucey, Simon. "PointNetLK: Robust and efficient point cloud registration using PointNet," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7163–7172, 2019.

[52] Biber, Peter, Straßer, Wolfgang. "The normal distributions transform: A new approach to laser scan matching," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2743–2748, 2003.

[53] Wang, Yue, Solomon, Justin M. "PRNet: Self-supervised learning for partial-to-partial registration," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 32, 2019.

[54] Huang, Xiaoshui, Mei, Guofeng, Zhang, Jian, Abbas, Rana. "A comprehensive survey on point cloud registration," *arXiv preprint arXiv:2103.02690*, 2021.

[55] Jian, Bing, Vemuri, Baba C. "A robust algorithm for point set registration using mixture of Gaussians," *Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV)*, vol. 2, pp. 1246–1251, 2005.

[56] Y. Yuan, A. Eckart, and D. Held, "DeepGMR: Learning Latent Gaussian Mixture Models for Registration," in *Proc. European Conf. Comput. Vis. (ECCV)*, 2020, pp. 733–750.

[57] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3D ShapeNets: A deep representation for volumetric shapes," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1912–1920, 2015.

[58] Wu, Zongze, Chen, Hongchen, Du, Shaoyi, Fu, Minyue, Zhou, Nan, Zheng, Nanning. "Correntropy Based Scale ICP Algorithm for Robust Point Set Registration," *Pattern Recognition*, vol. 93, 2019.

[59] Zhang, Xuetao, Jian, Libo, Xu, Meifeng. "Robust 3D point cloud registration based on bidirectional Maximum Correntropy Criterion," *PLOS ONE*, vol. 13, pp. e0197542, 2018.

[60] G. Xu, S. Du, D. Cui, S. Zhang, B. Chen, X. Zhang, J. Xue, Y. Gao. "Precise Point Set Registration Using Point-to-Plane Distance and Correntropy for LiDAR Based Localization," *2018 IEEE Intelligent Vehicles Symposium (IV)*, pp. 734-739, 2018.

[61] H. Lim, D. Kim, G. Shin, J. Shi, I. Vizzo, H. Myung, J. Park, and L. Carlone, "KISS-Matcher: Fast and Robust Point Cloud Registration Revisited," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, 2025. [Online]. Available: https://github.com/MIT-SPARK/KISS-Matcher