

ISFA2026-6497: SME Journal Track

R-SLPR: Region-based Small-to-Large Point-cloud Registration with Contrastive Learning

Yusen Wan
Department of
Mechanical
Engineering,
University of
Washington
Seattle, US

Zeyuan Chen
Department of
Mechanical
Engineering,
University of
Washington
Seattle, US

Qianshi Zou
Department of
Mechanical
Engineering,
University of
Washington
Seattle, US

Xu Chen
Department of
Mechanical
Engineering,
University of
Washington
Seattle, US

ABSTRACT

Point-cloud (PC) registration is fundamental to three-dimensional (3D) perception in robotic systems. However, classic registration algorithms falter when aligning a source PC containing limited, incomplete, or ambiguous geometric cues against a reference. This challenge of registering a small, partial PC to a significantly larger global reference is pervasive in real-world deployment yet remains insufficiently addressed by existing learning-based approaches, which typically assume comparable scales and significant overlap. To bridge this gap, we propose the Region-based Small-to-Large Point-cloud Registration framework (R-SLPR), a novel three-stage architecture that fundamentally reformulates the scale-mismatched registration problem into a sequence of region proposal, regional matching, and iterative refinement. Unlike conventional methods that fail to localize specific regions, R-SLPR explicitly identifies candidate regions prior to estimating rigid transformations, ensuring robust alignment even under severe scale mismatch. The framework introduces a Fibonacci Grid Segmentation method coupled with a contrastive learning objective to effectively generate and match local geometric patches. Building on this, a novel Cascade Anchor Selection and Refinement algorithm iteratively aligns the source with the target region to maximize precision. Extensive evaluation on ModelNet40 demonstrates that R-SLPR establishes a new state-of-the-art accuracy standard, outperforming prior approaches and significantly reducing position and rotation Mean Absolute Error (MAE) to 0.009 and 1.104, respectively.

Keywords: Point-cloud Registration, Small-to-Large Alignment, Contrastive Learning

1. INTRODUCTION

The rise of machine vision and 3D imaging has created significant opportunities for 3D machine perception with point clouds (PCs) in automation and robotics [1–3]. Robots can cap-

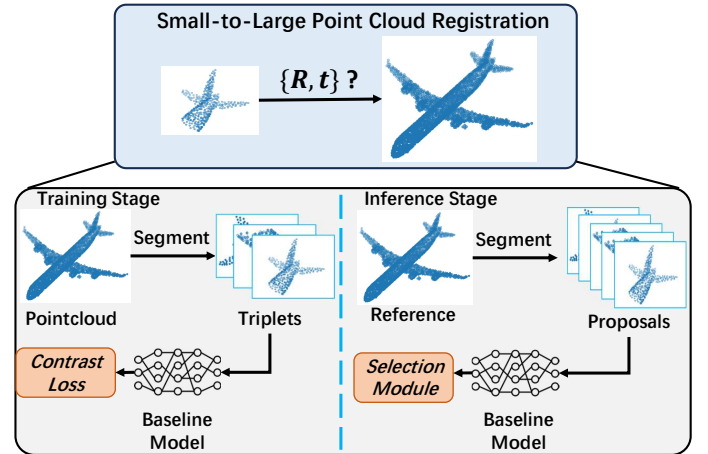


FIGURE 1: Example illustration of the small-to-large PC registration problem and an overview of the proposed method.

ture PCs of surrounding environments constantly and leverage registration methods to align and fuse these PCs to build a complete scene PC—a collection of 3D data that provides spatial information for such downstream tasks as navigation and manipulation. In this process, PC registration is crucial for aligning different spatial data, or more specifically, for providing the rigid transform between PCs and determining the fusion result of the final scene PC.

Classic PC registration has leveraged iterative local optimization and hand-crafted geometric descriptors, such as the Iterative Closest Point (ICP) algorithm [4] and Fast Point Feature Histograms (FPFH) [5]. Recent advances in PC registration have capitalized on deep neural networks (DNNs): e.g., RPMNet [6] for robust correspondence learning via Sinkhorn layers, PointNetLK [7] for iterative Lucas-Kanade alignment, and MCLNet [8] for multi-scale feature consistency. These approaches are

typically trained and evaluated using a source–reference split on multiple PC datasets, such as ModelNet40 [23] or KITTI [9]. More specifically, the reference PCs first undergo randomly generated rigid transformations that serve as ground-truth (GT) transforms. The networks then extract features and estimate the rigid transformation between the source and reference PCs, and compare the estimated transforms with the GT transforms for model training and performance evaluation. For this workflow to work, the source and reference PCs must maintain *comparable scales, substantial overlap, and congruent geometric coverage*.

However, as shown in Fig. 1, such assumptions are routinely violated in such real-world scenarios as manufacturing, where the source and reference PCs often differ drastically in size, density, and completeness, resulting in severe geometric and statistical mismatches [10, 11]. For example, in computer-assisted orthopedic surgery (CAOS), partially acquired PCs may capture as little as 20% of the target anatomy [10], while in computer-assisted manufacturing, workpiece localization typically requires aligning a sparse, partial scan with a complete PC derived from a CAD model [12]. In these settings, the overlap between the two PCs is not only limited but also often unknown *a priori*, leading to what we shall refer to as the *small-to-large PC registration problem*.

At the core of small-to-large PC registration is the challenge of identifying the correct region of a much larger reference point cloud when the source contains only a small and incomplete fragment. To address this fundamental issue, we propose a first-of-its-kind Region-based Small-to-Large Point Cloud Registration framework (R-SLPR), which reformulates registration as a region propose–match–refine process—a process that equips the system with an explicit region identification ability before estimating the rigid transformations, thereby enabling accurate registration even under severe scale mismatch. More specifically, to provide full and unbiased coverage of the reference point cloud, we first introduce a uniform segmentation strategy based on Fibonacci sphere sampling, which generates deterministic and evenly distributed directional anchors for constructing region proposals. To further strengthen the model’s ability to distinguish relevant regions from irrelevant ones, we design a patch-level contrastive learning strategy that pulls together embeddings of geometrically corresponding regions while pushing apart those of non-overlapping regions. In addition, a lightweight Cascade Anchor Selection and Refinement algorithm (CASR) iteratively evaluates qualities of the region proposals using alignment distance and gradually updates anchor directions, allowing the system to converge toward the true corresponding region even when the initial proposal is coarse. Extensive experiments on ModelNet40 demonstrate that R-SLPR substantially enhances registration robustness and accuracy under clean, noisy, unseen-category, and mixed conditions, achieving a position Root Mean Square Error (RMSE) of 0.02 and a rotation RMSE of 2.6 in benchmark evaluations.

Our main contributions are summarized as follows:

- We present R-SLPR, a first-of-its-kind region-based registration framework that enables reliable small-to-large alignment by structuring the process into proposal generation, matching, and refinement.
- We propose a deterministic Fibonacci Grid Segmentation

strategy that yields directionally uniform region proposals over the reference point cloud.

- We introduce a patch-level contrastive learning objective that enhances the discriminability of regional feature embeddings under severe scale disparity.
- We develop a Cascade Anchor Selection and Refinement algorithm that iteratively improves proposal quality and converges toward the correct region with minimal computational overhead.
- We validate the effectiveness of R-SLPR through extensive experimentation, demonstrating consistent improvements over representative registration baselines across diverse conditions.

The remainder of this article is organized as follows. Section 2 describes the background and related work. Section 3 introduces the problem formulation. Section 4 introduces the proposed R-SLPR method. Section 5 introduces the evaluation experiments. Section 6 concludes the paper.

2. RELATED WORKS

PC registration aims to estimate the rigid transformation between two PCs. A classical solution is the Iterative Closest Point (ICP) algorithm, which iteratively establishes point correspondences via nearest-neighbor search and estimates the rigid transformation using singular value decomposition (SVD) [4]. While effective in controlled settings, ICP is sensitive to initialization, noise, outliers, and limited overlap.

With the advancement of deep learning, numerous learning-based PC registration methods have been proposed. These methods can be broadly categorized into correspondence-based approaches and global feature-based approaches. Correspondence-based methods currently dominate the literature. To overcome the limitations of ICP, Wang *et al.* proposed Deep Closest Point (DCP), which replaces hand-crafted descriptors with deep feature extraction and learns point correspondences using a neural network [17]. To further improve correspondence reliability under noise and outliers, Wang *et al.* introduced the Multi-Features Guide Network (MFGNet), which employs a keypoint selection module to filter outliers and computes matching matrices using multiple learned features [18]. Similarly, Ginzburg *et al.* addressed the influence of outliers by defining a cosine-likelihood-based soft sampling map and selecting the top- K most confident points for rigid transformation estimation [19]. In this framework, features are extracted using modified Dynamic Graph CNNs (DGCNNglob and DGCNNloc) derived from DGCNN [20]. In scenarios where explicit one-to-one correspondences are ambiguous, alternative strategies have been explored. For instance, RPMNet replaces hard correspondence selection with a weighted average of potential matches by predicting a soft matching matrix and transformation parameters using a deep network [6]. In addition to correspondence-based methods, several approaches rely on global feature representations. Sarode *et al.* proposed PointNetLK, which uses PointNet to extract global features and applies the Lucas–Kanade algorithm to estimate the rigid transformation [7]. Li *et al.* further improved its generalization ability

in PointNetLK-Revisited [21]. Wu *et al.* proposed RORNet, a dual-branch overlap estimation network that filters points with low inlier probability, effectively combining similarity-based and score-based registration strategies [22]. Similarly, Tan *et al.* introduced MCLNet, an end-to-end framework that enforces point-level consistency for filtering and correspondence-level consistency for reliable matching matrix estimation [8].

Although these methods achieve high registration accuracy and computational efficiency, they primarily focus on registering PCs of comparable size with moderate to high overlap. The problem of aligning PCs with significant size disparities remains largely unaddressed. Chen *et al.* explored a full-to-partial registration framework based on reinforcement learning and evaluated robustness across different overlap ratios [10]; however, their work focuses on improving robustness rather than explicitly addressing the fundamental challenges posed by extreme size asymmetry. Overall, despite its prevalence in real-world applications, the *small-to-large PC registration problem* remains insufficiently explored in the existing literature.

3. PROBLEM FORMULATION

3.1. Small-to-Large Point Cloud Registration

Consider two PCs with a significant disparity in scale. We denote the smaller fragment as the source PC $S = \{s_i \in \mathbb{R}^3 \mid i = 1, 2, \dots, I\}$ and the complete PC as the reference PC $Q = \{q_j \in \mathbb{R}^3 \mid j = 1, 2, \dots, J\}$, where $I \ll J$. To quantify the scale mismatch, we define $\gamma \triangleq \frac{I}{J}$ as the cardinality ratio between PCs to be aligned. We aim to estimate the rigid transformation $\{R, t\}$ between these two PCs, where $R \in SO(3)$ is a rotation matrix and $t \in \mathbb{R}^3$ is a translation vector.

To show the correspondence of points, we define a matching matrix $M = \{m_{ij} \in [0, 1] \mid i = 1, 2, \dots, I; j = 1, 2, \dots, J\}$. The objective of small-to-large point cloud registration is to determine the optimal transformation that minimizes the alignment error between the transformed source fragment and the reference model. This is mathematically formulated as minimizing the mean distance between the transformed source PC S and the reference PC Q :

$$\arg \min_{R, t, M} \sum_{i=1}^I \sum_{j=1}^J m_{ij} \|Rs_i + t - q_j\|_2 \quad (1)$$

The value of m_{ij} here represents the degree of matching between s_i and q_j , and satisfies: $\forall i, \sum_{j=1}^J m_{ij} = 1$. This summation holds because every point $s_i \in S$ corresponds to exactly one point in Q .

3.2. Baseline Iterative Corresponding Point Match Method

Before we introduce the proposed solution approach, we summarize the core iterative refinement solution structure involved.

Mainstream correspondence-based methods rely on iterative point cloud matching. Each iteration consists of three primary stages: feature extraction, correspondence estimation, and transformation recovery. For each iteration, the source PC S and the reference PC Q form as the inputs, and the rigid transformation $\{R, t\}$ is the output. First, S and Q are fed into a feature extraction module to obtain the deep features F_S and F_Q by $F_S = f(S)$ and

$F_Q = f(Q)$. Then, the deep features and the original coordinates will be used for the matching-matrix calculation, mathematically abstracted as $M = g(F_S, F_Q, S, Q)$.

After obtaining the matching matrix, the corresponding coordinate $\hat{q}_i$ in the reference PC can be aligned to each point s_i in the source PC by two methods. The first method is a weighted average:

$$\hat{q}_i = \frac{1}{\sum_{j=1}^J m_{ij}} \sum_{j=1}^J (m_{ij} \cdot q_j) \quad (2)$$

The second method is the selection of the highest score by $\hat{q}_i = q_j$, where $j = \max_j m_{ij}$.

After getting the corresponding coordinate, we can calculate the rigid transforms between S and Q by SVD. Specifically, we build two coordinate matrices $\hat{S} = [s_1^T, s_2^T, \dots, s_I^T] \in \mathbb{R}^{3 \times I}$ and $\hat{Q} = [\hat{q}_1^T, \hat{q}_2^T, \dots, \hat{q}_I^T] \in \mathbb{R}^{3 \times I}$ to compute $H = \hat{S}\hat{Q}^T$. Computing the SVD of H , $H = U\Sigma V^T$, yields the rotation matrix $R = UV^T$ and the translation vector $t = \frac{1}{n} \sum_{i=1}^I \{\hat{q}_i - Rs_i\}$.

4. REGION-BASED SMALL-TO-LARGE PC REGISTRATION METHOD

4.1. Overview

Fig. 2 shows the proposed R-SLPR framework to address the small-to-large PC matching problem. At the core, R-SLPR introduces a region identification before estimating the rigid transformations, and forms a contrastive learning in a two-stage training-inference sequence. This is achieved algorithmically by integrating a baseline registration network, a contrastive learning loss function, a Fibonacci Grid Segmentation (FGS) strategy for region proposal generation, and a Cascade Anchor Selection and Refinement (CASR) module for iterative proposal refinement.

Given a small source PC fragment and a much larger reference PC, R-SLPR first decomposes the reference PC into a set of overlapping regional proposals $P = \{P_n \mid n = 1, 2, \dots, N\}$ using the FGS scheme, where each proposal is associated with a deterministic anchor direction.

In the training stage, we introduce three PCs: a source PC S , a reference PC Q from an overlapping proposal, and an irrelevance PC Ir from a distant region. The goal of the training stage is to regress accurate rigid transformations between overlapping patches and produce discriminative feature embeddings that separate relevant and irrelevant regions by minimizing the L_2 distance $\|f(S) - f(Q)\|$ and maximizing $\|f(S) - f(Ir)\|$.

In the inference stage, the trained baseline model is applied to every proposal in the current proposal set. For each proposal, the model predicts a local rigid transformation between S and the proposal $P = \{P_n \mid n = 1, 2, \dots, N\}$. The prediction results are then evaluated by an average alignment distance metric, according to which R-SLPR selects the best anchor by a CASR module and updates all anchors via an exponential moving average towards the best anchor. By repeating segmentation and selection with the updated anchors, R-SLPR gradually converges to the correct region and the final rigid transformation.

4.2. Model Training

4.2.1. Fibonacci Grid Segmentation and Triplet Generation Because input PCs are sampled on surfaces rather

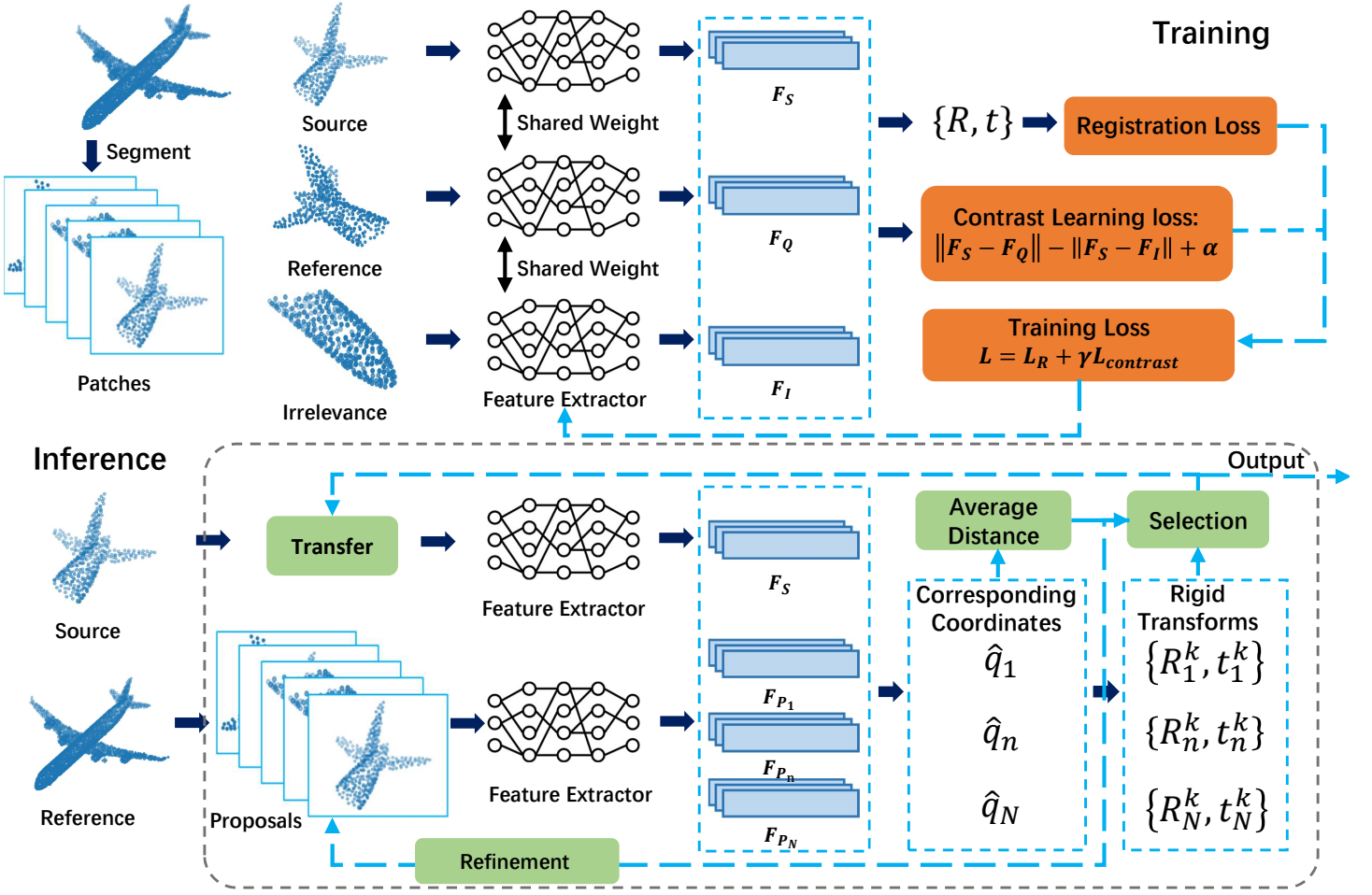


FIGURE 2: The overall structure of the proposed R-SLPR method.

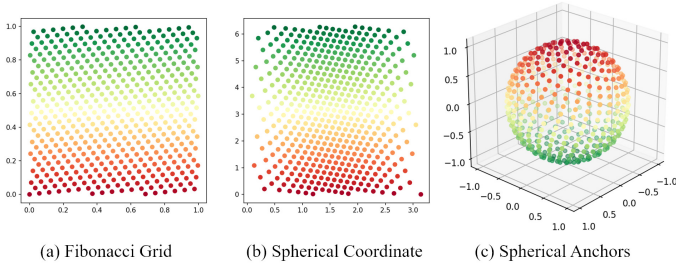


FIGURE 3: Proposed anchor generation based on a Fibonacci grid: (a) the original grid, (b) the spherical coordinates, and (c) the anchors.

than volumes, voxel-based methods cannot guarantee uniform and complete coverage of PCs. Instead, we propose a Fibonacci Grid Segmentation (FGS) method and leverage a Fibonacci grid, a deterministic lattice, to generate directionally uniform anchors on the unit sphere [24] and use them to segment the PCs into patches fully and uniformly (Fig. 3).

First, we generate a Fibonacci grid as illustrated in Fig. 3a,

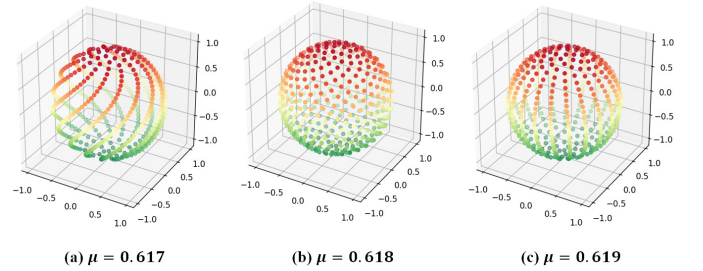


FIGURE 4: Effect of μ in the proposed anchor generation.

based on the following coordinate definition:

$$\begin{cases} u_n = \frac{n-1}{N} \\ v_n = \{\mu n\} \end{cases} \quad (3)$$

where (u_n, v_n) is the coordinate of the n th point of the grid, $n = \{1, 2, 3, \dots, N\}$ and N is the total number of sampling, $\{\cdot\}$ is the fractional part operator, e.g. $\{5.3\} = 0.3$, and μ is a grid parameter. Fig. 4 shows the effect of different μ values on the grid points. To make the grid points on the sphere as uniform as

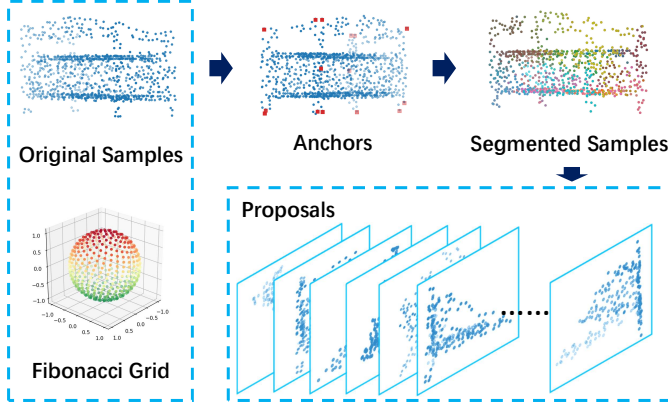


FIGURE 5: The process of segmenting the point clouds into proposals by the Fibonacci grid.

possible, the best value of μ is $\frac{\sqrt{5}-1}{2} (\approx 0.618)$, the reciprocal of the golden ratio.

After generating the Fibonacci grid, we project it onto 3D Cartesian coordinates on the unit sphere, obtaining uniformly distributed spherical grid points (Fig. 3b):

$$\begin{cases} \theta_n = \arccos(2u_n - 1) \\ \phi_n = 2\pi v_n \end{cases} \quad (4)$$

where θ_n is the polar angle and ϕ_n is the azimuthal angle.

We can now get the coordinate of grid points in the rectangular coordinate (Fig. 3c):

$$\begin{cases} x_n = r \cos \phi_n \sin \theta_n \\ y_n = r \sin \phi_n \sin \theta_n \\ z_n = r \cos \theta_n \end{cases} \quad (5)$$

where x_n , y_n , and z_n are the coordinates of the point of the grid in the rectangular coordinate system, r is a constant, chosen as $r = 1$ to anchor a unit spherical surface.

After Eq. (5), we have a set of uniform spherical anchor vectors $\vec{A} = \{\vec{a}_n = (x_n, y_n, z_n) \mid n = 1, 2, 3, \dots, N\}$, and can use these anchors to segment PCs by nearest search, as illustrated in Fig. 5. First, we translate the PC center to the origin by $\tilde{Q} = Q - \text{mean}(Q)$. Then, we compute the dot product between the centered PC $\tilde{Q}$ and the anchor vector $\vec{a}_n$ and obtain the distance score by $c_n = \tilde{Q} \cdot \vec{a}_n$. We get the indexes of the top-k highest scores by $I_n = \text{topk}(c_n)$ in Python and get the proposal $P_n = Q[I_n]$. Finally, we obtain the segmented reference PC as $P = \{P_n \mid n = 1, 2, \dots, N\}$.

After segmenting the PCs into patches, we can generate triplets from these proposals. Each triplet contains a source PC, a reference PC, and an irrelevant PC. We choose a proposal randomly as the source PC S and the nearest proposal as the reference PC Q . Next, we choose a proposal whose anchor is the farthest away from that of the source PC as the irrelevant PC Ir , completing the training triplet (S, Q, Ir) selection.

4.2.2. Model Training with Contrastive Learning

With the established training triplet, we employ a shared feature extractor network, $f(\cdot)$, which processes each point cloud to generate its respective discriminative feature embedding: $F_S = f(S)$, $F_Q = f(Q)$, and $F_{Ir} = f(Ir)$. These learned features, along with the raw point clouds S and Q , then feed into our pose estimation module, $g(\cdot)$, designed to directly regress the rigid transformation $\{R_{\text{pred}}, t_{\text{pred}}\} = g(S, Q, F_S, F_Q)$ to align S with Q .

Our comprehensive training objective integrates two critical components to optimize both the accuracy of the pose estimation and the quality of the learned feature space. First, we minimize the error between our predicted transformation $\{R_{\text{pred}}, t_{\text{pred}}\}$ and the ground truth $\{R_{\text{gt}}, t_{\text{gt}}\}$ (e.g., via an L_2 or geodesic pose loss). Second, to sculpt a robust feature space, we enforce a triplet-based feature loss: minimizing the Euclidean distance $\|F_S - F_Q\|$ for positive pairs, while simultaneously maximizing the distance $\|F_S - F_{Ir}\|$ for negative pairs. This ensures our feature embeddings are inherently discriminative for accurate point cloud registration.

For the first goal, we leverage the registration loss of the baseline model:

$$L_R = \text{loss}(R_{\text{gt}}, t_{\text{gt}}, R_{\text{pred}}, t_{\text{pred}}) \quad (6)$$

where $R_{\text{gt}}, t_{\text{gt}}$ are the GT transforms, and $R_{\text{pred}}, t_{\text{pred}}$ are the predicted transforms. The loss function quantifies the $SE(3)$ transformation error between the predictions and the ground truth, with its formulation adopted from the original literature of the baseline model. The precise formulation of the loss function is tailored to the baseline model's architecture and learning objectives. For example, in the RPMNet [6] that we will use later in the experimentation section, the loss contains an L_1 distance and a weighted loss of the matching matrix:

$$\begin{aligned} L_{\text{RPM}} &= L_{\text{reg}} + \lambda_o L_{\text{inlier}} \\ L_{\text{reg}} &= \frac{1}{J} \sum_j |(R_{\text{gt}} s_j + t_{\text{gt}}) - (R_{\text{pred}} s_j + t_{\text{pred}})| \\ L_{\text{inlier}} &= -\frac{1}{J} \sum_j \sum_i m_{ij} - \frac{1}{I} \sum_i \sum_j m_{ij} \end{aligned} \quad (7)$$

For our next goal of sculpting a robust feature space, we design a contrastive learning loss:

$$L_{\text{Contrastive}} = \frac{1}{N} \sum_{n=1}^N \max(0, \|F_S - F_Q\| - \|F_S - F_{Ir}\| + \alpha) \quad (8)$$

where n is the size of the training set; $F_S, F_Q, F_{Ir} \in \mathbb{R}^d$ are the feature vectors of the source, the reference, and the irrelevance patches, respectively. The parameter $\alpha (> 0)$ is a margin that enforces a minimum separation between the feature embeddings of dissimilar samples, thereby preventing representational collapse where all embeddings converge to a single point.

The final loss function used for training the model is given by the following equation:

$$L = L_R + \lambda L_{\text{Contrastive}} \quad (9)$$

where λ is a balancing coefficient to trade off the contributions of these two losses.

Algorithm 1 Inference with Cascade Anchor Selection and Refinement (CASR)

Require: Source point cloud S ; reference point cloud Q ; baseline model $g(\cdot)$ and extractor $f(\cdot)$; number of anchors N ; patch size k ; EMA step η ; iterations K .

Ensure: Final rigid transform (R^*, t^*) .

```

1: Initialize Fibonacci anchors  $\{\tilde{a}_n^0\}_{n=1}^N$  on unit sphere.
2:  $(R^*, t^*) \leftarrow (\mathbf{I}, \mathbf{0})$ .
3: for  $k = 0$  to  $K - 1$  do
4:   (Segment) Center:  $\tilde{Q} \leftarrow Q - \text{mean}(Q)$ .
5:   for  $n = 1$  to  $N$  do
6:     Scores  $c_n \leftarrow \tilde{Q} \cdot \tilde{a}_n^k$ .
7:      $I_n \leftarrow \text{topk}(c_n)$ ,  $P_n^k \leftarrow Q[I_n]$ .
8:   end for
9:   (Register & Evaluate)
10:  for  $n = 1$  to  $N$  do
11:     $(R_n^k, t_n^k, \{\hat{q}_i\}) \leftarrow g(S, P_n^k, f(S), f(P_n^k))$ .
12:    Average distance  $d_n^k \leftarrow \frac{1}{|S|} \sum_i \|R_n^k s_i + t_n^k - \hat{q}_i\|^2$  (Eq. 10)
13:  end for
14:   $b \leftarrow \arg \min_n d_n^k$ ;  $(R^*, t^*) \leftarrow (R_b^k, t_b^k)$ .
15:  for  $n = 1$  to  $N$  do
16:     $\tilde{a}_n^{k+1} \leftarrow (1 - \eta)\tilde{a}_n^k + \eta\tilde{a}_b^k$  (Eq. 11)
17:    Normalize  $\tilde{a}_n^{k+1} \leftarrow \tilde{a}_n^{k+1} / \|\tilde{a}_n^{k+1}\|$ .
18:  end for
19: end for
20: return  $(R^*, t^*)$ .
```

4.3. Inference with Cascade Anchor Selection and Refinement

In the inference stage, we segment the reference PC into proposals by the same method as in the training stage and utilize the trained baseline model to iteratively predict the transformations between the source PC and each individual proposal (Algorithm 1). In each iteration, we will select the best transform to update the anchors. Recall Eq. (1) and observe that a smaller value of cost indicates a better registration result. We employ the average distance as the evaluation metric. For each transform $\{R_n^k, t_n^k\}$, we calculate the average distance between the transformed source PC and the corresponding coordinates:

$$d_n^k = \frac{1}{I} \sum_{i=1}^I \|R_n^k s_i + t_n^k - \hat{q}_i\|_2 \quad (10)$$

where s_i is a point in S and $\hat{q}_i$ is the corresponding coordinate used in aligning S and proposal P_n to calculate each transform $\{R_n^k, t_n^k\}$. The transform with the lowest distance is then selected as the best result $\{R_b^k, t_b^k\}$.

After that, the anchor $\tilde{a}_b^k$ related to the best result $\{R_b^k, t_b^k\}$ is used to update other anchors in the next iteration from an exponential moving average as shown in Eq. 11:

$$\tilde{a}_n^{k+1} = (1 - \lambda)\tilde{a}_n^k + \gamma\tilde{a}_b^k \quad (11)$$

where $\tilde{a}_n^k$ is the anchor of the current iteration, $\tilde{a}_n^{k+1}$ is the anchor of the next iteration, and γ is a hyperparameter chosen manually.

5. EXPERIMENTAL EVALUATION

To validate the proposed method, we conduct comprehensive experiments on the general object dataset ModelNet40 [23] to evaluate its overall performance. We first introduce the ModelNet40 dataset and explain how it is processed for SLPR problem (Sec. 5.1). Then, we describe the experimental formulation, including the evaluation protocols, baseline methods, and quantitative metrics (Sec. 5.2). Finally, we present the experimental results and analyze the performance of the proposed method under different settings (Sec. 5.3). Furthermore, we perform ablation studies on the key components of the proposed method to analyze their contributions to the overall performance and demonstrate their necessity (Sec. 5.4). In addition, we verify its applicability to manufacturing scenarios on the industrial dataset ISOPR [12] (Sec. 5.5).

5.1. Dataset Introduction and Data Preparation

ModelNet40 is a widely used benchmark in the fields of 3D computer vision and PC analysis. It consists of 12311 pre-aligned 3D CAD models. For each model, we extract a PC with 1024 points, whose coordinates are normalized into $[-1, 1]$. These CAD models are categorized into 40 common object classes, such as airplanes, chairs, and tables, which are formally divided into 9843 training samples and 2468 test samples, providing a standardized foundation for evaluating model generalizability.

The ISOPR dataset is an industrial scene object point-cloud registration dataset constructed in NVIDIA Isaac Sim to evaluate point-cloud registration methods under manufacturing-oriented conditions. It contains 2,000 partial point clouds of 67 common workpiece models captured from simulated depth-camera observations, together with their ground-truth positions and orientations.

For the small-to-large registration, we extract a local point cloud from each sample as the source point cloud S , and use the original full sample as the reference point cloud Q . Then, we apply random transforms as ground truth transforms on the source point clouds and let the models predict the transforms.

5.2. Experiment Formulation

For the proposed R-SLPR, we choose RPMNet as the baseline model because of its excellent registration accuracy and robustness. All implementations are developed in Python. Experiments are conducted on a workstation equipped with an NVIDIA RTX 4090 GPU and an Intel Core i9-14900K CPU (6.0 GHz) running Ubuntu 22.04. We employ the Adam optimizer with a learning rate of 0.001. The baseline model is trained by the method mentioned in Section 4.1.

To rigorously evaluate the proposed method, we conduct comprehensive comparisons against both handcrafted (e.g., ICP, Small-GICP) and representative learning-based (e.g., PointNetLK [7], RPM-Net[6], MFGNet[18]) point cloud registration approaches. All learning-based baselines are faithfully reproduced using their official implementations and the training protocols described in their respective publications. To evaluate the registration results, we compute the Root Mean Square Error (RMSE-r) and the Mean Absolute Error (MAE-r) of the Euler angles as metrics for rotation, and we compute the RMSE and

TABLE 1: Experimental results of R-SLPR and existing methods on ModelNet40 Dataset with the experimental setting "clean". The red fonts highlight the best result in each metric.

Experiment Set	Method	Sampling Rate 0.2				Sampling Rate 0.3			
		RMSE-r (°)	MAE-r (°)	RMSE-t	MAE-t	RMSE-r (°)	MAE-r (°)	RMSE-t	MAE-t
Clean	ICP [4]	65.657	56.827	0.642	0.541	62.317	53.806	0.591	0.495
	Small_GICP [16]	64.329	54.128	0.587	0.501	61.334	52.152	0.547	0.422
	PointNetLK [7]	45.825	23.654	0.673	0.423	23.340	15.151	0.497	0.308
	PointNetLK-revisited [21]	26.916	17.365	0.567	0.349	25.230	16.238	0.544	0.294
	RPMNet [6]	4.102	2.223	0.065	0.020	3.841	2.011	0.039	0.026
	MFGNet [18]	29.158	18.802	0.450	0.304	13.272	10.366	0.180	0.135
	RORNet [22]	18.635	15.314	0.322	0.240	16.469	12.391	0.340	0.271
	MCLNet [8]	3.556	1.132	0.224	0.156	3.429	1.186	0.168	0.092
	R-SLPR (ours)	2.632	1.104	0.020	0.009	3.304	1.268	0.027	0.012

TABLE 2: Experimental results of R-SLPR and existing methods on the ModelNet40 Dataset with the experimental setting "Unseen". The red fonts highlight the best result in each metric.

Experiment Set	Method	Sampling Rate 0.2				Sampling Rate 0.3			
		RMSE-r (°)	MAE-r (°)	RMSE-t	MAE-t	RMSE-r (°)	MAE-r (°)	RMSE-t	MAE-t
Unseen	ICP [4]	-	-	-	-	-	-	-	-
	Small_GICP [16]	-	-	-	-	-	-	-	-
	PointNetLK [7]	44.203	27.126	0.591	0.365	21.200	16.312	0.400	0.269
	PointNetLK-revisited [21]	27.320	16.602	0.452	0.300	25.540	16.231	0.411	0.278
	RPMNet [6]	4.368	2.371	0.071	0.042	4.129	2.413	0.056	0.035
	MFGNet [18]	30.556	20.222	0.237	0.173	11.521	7.072	0.090	0.060
	RORNet [22]	19.913	14.260	0.366	0.204	16.395	12.856	0.410	0.306
	MCLNet[8]	3.638	1.328	0.150	0.098	3.521	1.200	0.140	0.101
	R-SLPR (ours)	2.833	1.234	0.026	0.012	2.977	1.183	0.023	0.010

TABLE 3: Experimental results of R-SLPR and existing methods on the ModelNet40 Dataset with the experimental setting "Noise". The red fonts highlight the best result in each metric. "-" means the value is not obtained for some reason.

Experiment Set	Method	Sampling Rate 0.2				Sampling Rate 0.3			
		RMSE-r (°)	MAE-r (°)	RMSE-t	MAE-t	RMSE-r (°)	MAE-r (°)	RMSE-t	MAE-t
Noisy	ICP [4]	66.782	58.241	0.636	0.514	63.587	55.214	0.562	0.495
	Small_GICP [16]	65.327	56.831	0.638	0.499	64.521	55.368	0.582	0.433
	PointNetLK [7]	45.825	23.654	0.673	0.423	23.340	15.151	0.497	0.308
	PointNetLK-revisited [21]	26.394	17.575	0.507	0.388	25.894	14.238	0.519	0.352
	RPMNet [6]	4.562	2.740	0.072	0.025	3.985	2.219	0.047	0.019
	MFGNet [18]	33.340	21.842	0.404	0.320	14.395	9.567	0.168	0.124
	RORNet [22]	10.254	7.352	0.224	0.156	7.856	5.421	0.208	0.149
	MCLNet [8]	-	-	-	-	-	-	-	-
	R-SLPR (ours)	2.613	1.142	0.027	0.013	2.974	1.229	0.021	0.010

TABLE 4: Experimental results of R-SLPR and existing methods on the ModelNet40 Dataset with the experimental setting "Noise & Unseen". The red fonts highlight the best result in each metric. "-" means the value is not obtained for some reason.

Experiment Set	Method	Sampling Rate 0.2				Sampling Rate 0.3			
		RMSE-r (°)	MAE-r (°)	RMSE-t	MAE-t	RMSE-r (°)	MAE-r (°)	RMSE-t	MAE-t
Noisy & Unseen	ICP [4]	-	-	-	-	-	-	-	-
	Small_GICP [16]	-	-	-	-	-	-	-	-
	PointNetLK [7]	46.014	25.236	0.594	0.317	23.721	14.106	0.443	0.357
	PointNetLK-revisited [21]	26.256	17.167	0.512	0.343	25.689	15.597	0.483	0.281
	RPMNet [6]	4.714	2.828	0.041	0.022	4.114	2.263	0.033	0.019
	MFGNet [18]	80.267	61.713	0.424	0.348	14.952	10.697	0.127	0.084
	RORNet [22]	11.026	7.864	0.271	0.197	7.882	5.663	0.325	0.188
	MCLNet [8]	-	-	-	-	-	-	-	-
	R-SLPR (ours)	2.851	1.101	0.031	0.014	2.673	1.174	0.028	0.011

TABLE 5: Ablation experiment results of R-SLPR on the ModelNet40 Dataset. The red fonts highlight the best result in each metric.

FG	CASR	CL	RMSE-r (°)	MAE-r (°)	RMSE-t	MAE-t
			4.102	2.223	0.065	0.020
✓			3.268	1.016	0.035	0.012
✓	✓		2.938	1.202	0.024	0.010
✓	✓	✓	2.632	1.104	0.020	0.009

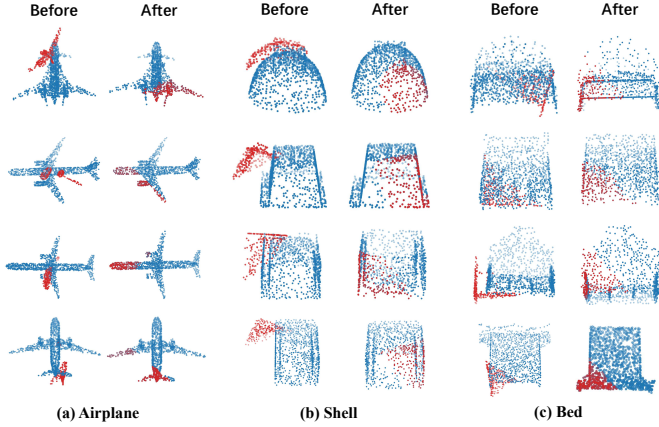


FIGURE 6: Example matching results of R-SLPR. The left columns display the point clouds before registration, while the right columns present the aligned point clouds.

MAE of the translation vector, labeled as RMSE-t and MAE-t, for translation [12].

To demonstrate the generalizability of the proposed method, we conducted comprehensive experiments under varying configurations by following previous works: 1) clean, 2) unseen, 3) noise, and 4) noise and unseen. In the “unseen” setting, models are trained on the first 20 categories and tested on the last 20 categories, while, in the “noise” setting, Gaussian noise is added into both the training and test samples. Furthermore, to systematically evaluate the robustness of the proposed method to varying source PC densities, we conducted experiments at multiple cardinality ratios, specifically $\gamma = 0.2$ and $\gamma = 0.3$.

5.3. ModelNet40 Experiment Result

Fig. 6 shows a few examples of the registration experiments. Tables 1, 2, 3, and 4 summarize the experimental results under all configurations. Overall, R-SLPR consistently outperforms prior approaches across all conditions. In particular, it achieves robust performance with the RMSE-r below 3.0 degrees and the MAE-r below 1.3, while the RMSE-t is below 0.025. Among prior methods, RPMNet and MCLNet performed best, with approximately RMSE-r 4.0 and 3.5, respectively. Other methods, such as MFGNet, PointNetLK, etc., exhibit poor fitting behavior when faced with the small-to-large registration problem. These results demonstrate that R-SLPR not only surpasses existing methods but also fulfills accuracy requirements for real-world applications.

The comparative results demonstrate that the proposed method achieves superior performance in nearly all evaluation

metrics. In the experimental settings “Clean” and “Unseen”, MCLNet delivers the strongest performance among the baseline methods, attaining an RMSE-r of approximately 3.5, as shown in Tables 1 and 2. However, as shown in Tables 3 and 4, it exhibits notable limitations in noise robustness, failing to converge in both “Noise” and “Noise & Unseen” configurations, which preclude reporting meaningful results for these cases. It is noteworthy that a noticeable discrepancy exists between the RMSE and MAE values of the proposed method. This phenomenon primarily stems from the higher sensitivity of RMSE to outlier predictions. In the context of the small-to-large registration problem, small PCs, randomly segmented from the large PCs, may be geometrically uninformative regions that lead to occasional registration failures, which inflate the value of RMSE compared to MAE.

In terms of translation accuracy, the proposed method significantly outperforms all competing approaches, achieving RMSE-t from 0.02 to 0.03 among the experimental settings, while the RMSE-t of the best performing previous method, RPMNet, is from 0.4 to 0.7. This advantage stems from a fundamental limitation of existing approaches that they rely on the method mentioned in Section 3.1 to compute translation by aligning PC centroids, which is inappropriate for the small-to-large registration where the centroids of the two point clouds are inherently misaligned. In contrast, R-SLPR first identifies corresponding regions in the large reference PCs and then estimates the rigid transforms, resulting in substantially higher precision in translation.

When tested on unseen object categories, R-SLPR demonstrates robust overall performance and does not exhibit significant degradation in accuracy with (shown in Tables 2 and 4) or without (shown in Tables 1 and 3) unseen objects. This resilience stems from its core design principle: rather than relying on features of PCs themselves, the method focuses on learning the relationship between the source PCs S and the reference PCs Q . This fundamental emphasis on cross-instance relationships enables strong generalization capability, allowing the framework to maintain stable performance even when applied to object categories not encountered during training.

Similarly, R-SLPR demonstrates robust performance under Gaussian noise corruption. This resilience is attributed to the CASR method, which leverages the average distance between transformed source PCs and corresponding coordinates for proposal selection and can effectively mitigate noise interference. In contrast, several previous methods (e.g., MCLNet) exhibit significant performance degradation under noisy conditions, with some even failing to converge properly due to their sensitivity to perturbed point distributions.

5.4. Ablation Study

To quantify the effectiveness of the major components in R-SLPR, we conduct a comprehensive ablation study under the same small-to-large point-cloud registration protocol used in the main experiments. Because the proposed framework integrates region proposal generation, contrastive regional representation learning, and cascade anchor selection and refinement, some components are structurally necessary for maintaining a valid registration pipeline and cannot be simply removed. Accordingly, we evaluate the contribution of these designs through both

TABLE 6: Experimental results of R-SLPR and existing methods on the ISOPR Dataset with the experimental setting "Noise & Unseen". The red fonts highlight the best result in each metric. "-" means the value is not obtained for some reason.

Experiment Set	Method	Sampling Rate 0.2				Sampling Rate 0.3			
		RMSE-r (°)	MAE-r (°)	RMSE-t	MAE-t	RMSE-r (°)	MAE-r (°)	RMSE-t	MAE-t
Clean	RPMNet [6]	0.833	0.476	0.011	0.006	0.608	0.339	0.008	0.004
	MFGNet [18]	15.118	10.424	0.208	0.143	11.032	7.134	0.141	0.095
	RORNet [22]	21.970	17.545	0.273	0.220	20.756	16.847	0.260	0.213
	MCLNet [8]	2.080	0.310	0.022	0.003	2.234	0.299	0.020	0.002
	R-SLPR (ours)	0.490	0.178	0.006	0.002	0.463	0.171	0.006	0.002
Noisy	RPMNet [6]	0.905	0.434	0.012	0.006	0.771	0.326	0.010	0.003
	MFGNet [18]	14.686	9.786	0.195	0.133	12.449	8.118	0.157	0.104
	RORNet [22]	21.972	17.757	0.292	0.235	20.504	16.500	0.264	0.215
	MCLNet [8]	-	-	-	-	-	-	-	-
	R-SLPR (ours)	0.506	0.225	0.006	0.002	0.498	0.211	0.006	0.003

component removal and controlled replacement. In particular, we compare the full R-SLPR framework with variants that remove the contrastive learning loss, disable the CASR refinement module, or replace the Fibonacci Grid Segmentation strategy with alternative anchor-generation schemes such as random sampling or farthest point sampling. In addition, we investigate the influence of the number of CASR iterations to assess the role of iterative proposal refinement. All ablated variants are evaluated using the same training and testing splits, source-to-reference ratios, and performance metrics as the main benchmark experiments. This ablation study aims to determine whether the observed performance gains arise from the proposed region-based formulation and its individual algorithmic components rather than from the registration backbone alone.

5.5. ISOPR Experiment Result

To further demonstrate the applicability of R-SLPR in manufacturing scenarios, we evaluate it additionally on the ISOPR dataset proposed in the literature[12]. Different from ModelNet40, ISOPR contains point clouds of industrial parts with more manufacturing-specific geometric characteristics, such as local planar surfaces, sharp edges, repetitive structures, and partial geometric ambiguity.

Since the ISOPR experiment is intended as an application-oriented validation rather than a full-scale benchmark, we compare R-SLPR with representative baselines selected from the ModelNet40 experiments, including MFGNet [18], RPMNet [6], RORNet [22], and MCLNet [8]. The experimental formulation is consistent with the protocols used in previous studies [12] and follows the configurations described in Sec. 5.2.

The results are reported in Table 6. Overall, R-SLPR achieves the best performance on the ISOPR dataset, indicating that the proposed region-based formulation generalizes well to manufacturing-style point clouds. Compared with RPMNet, R-SLPR reduces the rotation RMSE from 0.833° to 0.490° and the translation RMSE from 0.011 to 0.006 under the clean setting with a sampling rate of 0.2. Compared with MCLNet, R-SLPR further improves the rotation MAE from 0.310° to 0.178° and the translation MAE from 0.003 to 0.002. These improvements suggest that directly applying conventional registration methods to small-to-large manufacturing data remains challenging, espe-

cially when the source point cloud covers only a local region of the reference model.

The superior performance of R-SLPR mainly comes from its explicit region identification mechanism. By first localizing the most plausible corresponding region and then performing refined registration with CASR, R-SLPR effectively reduces the influence of scale mismatch and centroid inconsistency in small-to-large registration. The ISOPR results further show that the proposed method can generalize beyond synthetic ModelNet40 benchmarks and is applicable to manufacturing-oriented tasks such as CAD-to-scan alignment, workpiece localization, and robotic inspection.

6. CONCLUSION AND FUTURE WORK

This paper presented Region-based Small-to-Large Point-cloud Registration (R-SLPR), a framework designed to overcome significant scale discrepancies in point-cloud alignment through a multi-stage regional approach. By integrating Fibonacci Grid Segmentation (FGS) with a contrastive learning-based model and a Cascade Anchor Selection and Refinement (CASR) algorithm, the method effectively bridges the gap between small-scale source patches and large-scale reference environments. Experimental validation on the ModelNet40 dataset confirms the precision of this approach, yielding a position Mean Absolute Error (MAE) of 0.009 and a rotation MAE of 1.104°. These results demonstrate that the proposed region-based formulation significantly enhances registration robustness and estimation accuracy in “small-to-large” settings.

This research opens up several promising directions for future work. First, an end-to-end framework could be developed to unify proposal generation, selection, and registration within a single learnable pipeline, potentially enhancing both efficiency and coordination between modules. Also, the proposal screening and generation mechanism could benefit from further optimization such as the introduction of a hierarchical scoring network or more geometrically-aware sampling strategies.

REFERENCES

- [1] H. Chen, Y. Huang, Q. Xie, et al, “Multiscale feature line extraction from raw point clouds based on local surface variation and anisotropic contraction,” IEEE Trans. Autom.

- Sci. Eng., vol. 19, no. 2, pp. 1003-1016, February 2021, doi: 10.1109/TASE.2021.3053006
- [2] Y. Zhang, Y. Geng, X. Tian, et al, "A feature extraction approach over workpiece point clouds for robotic welding," *IEEE Trans. Autom. Sci. Eng.*, vol. 22, pp. 75-84, January 2024, doi: 10.1109/TASE.2023.3345868
 - [3] P. Chi, Z. Wang, H. Liao, et al, "Low-latency visual-based high-quality 3-D reconstruction using point cloud optimization," *IEEE Sensors J.*, vol. 23, no. 17, pp. 20055-20065, September. 2023, doi: 10.1109/JSEN.2023.3297232
 - [4] P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239-256, Feb. 1992.
 - [5] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (FPFH) for 3D registration," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2009, pp. 3212-3217.
 - [6] Z. J. Yew, & H. L. Gim, "RPM-net: Robust point matching using learned features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June 2020, pp. 11824-11833.
 - [7] Y. Aoki, H. Goforth, R. A. Srivatsan, & S. Lucey, "Pointnetlk: Robust efficient point cloud registration using pointnet," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June 2019, pp. 7163-7172.
 - [8] B. Tan, H. Qin, X. Zhang, et al, "Using multi-level consistency learning for partial-to-partial point cloud registration," *IEEE Trans. Vis. Comput. Graph.*, vol. 30, no. 8, pp. 4881-4894, August 2023, doi: 10.1109/TVCG.2023.3280171
 - [9] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231-1237, Sep. 2013.
 - [10] F. Chen, Q. Du, J. Zhao, et al, "A generalized full-to-partial registration framework of 3d point sets for computer-aided orthopedic surgery," *IEEE Trans. on Biomed. Eng.*, vol. 71, no. 3, pp. 1010-1021, March 2023, doi: 10.1109/TBME.2023.3325355.
 - [11] Y. Feng, Q. Bi, & Z. Lin, "PGPNet: A Novel Iterative Partial-Global-Partial Point Cloud Registration Method," in *Int. Conf. Artif. Intell. Comput. Eng. (ICAICE)*, November 2024, pp. 146-152.
 - [12] Y. Wan, & X. Chen, "iLSPR: A Learning-based Scene Point-cloud Registration method for robotic spatial awareness in intelligent manufacturing," *Robot. Comput. Integr. Manuf.*, vol. 99, pp. 103204, June 2026, doi: 10.1016/j.rcim.2025.103204.
 - [13] R. Girshick, "Fast r-cnn," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, December 2015, pp. 1440-1448.
 - [14] Z. Cai, & N. Vasconcelos (2019). "Cascade R-CNN: High quality object detection and instance segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 5, pp. 1483-1498, May 2021, doi: 10.1109/TPAMI.2019.2956516.
 - [15] F. Schroff, D. Kalenichenko, & J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June 2015, pp. 815-823.
 - [16] K. Koide, "Small_GICP: Efficient and Parallel Algorithms for Point Cloud Registration," *J. Open Source Softw.*, vol. 9, no. 100, pp. 6948, 2024.
 - [17] Y. Wang, & J. M. Solomon, "Deep closest point: Learning representations for point cloud registration," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, November 2019, pp. 3523-3532
 - [18] X. Zhang, X. Fan, P. Chen, et al, "MFGNet: Multi-branch feature generation networks for few-shot remote sensing scene classification," *IEEE Trans. Geosci. Remote. Sens.*, vol. 61, pp. 1-13, May 2023, doi: 10.1109/TGRS.2023.3275754.
 - [19] D. Ginzburg, & D. Raviv, "Deep weighted consensus dense correspondence confidence maps for 3D shape registration," in *IEEE Int. Conf. Image Process. (ICIP)*, October 2022, pp. 71-75.
 - [20] Y. Wang, Y. Sun, Z. Liu, et al, "Dynamic graph cnn for learning on point clouds," *ACM Trans. Graph.*, vol. 38, no. 5, pp. 1-12, October 2019, doi:10.1145/3326362.
 - [21] X. Li, J. K. Pontes, & S. Lucey, "Pointnetlk revisited", in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June 2021, pp. 12763-12772.
 - [22] Y. Wu, Y. Zhang, W. Ma, et al, "RORNet: Partial-to-partial registration network with reliable overlapping representations," *IEEE Trans. Neural Netw. Learn. Syst.* vol. 35, no. 11, pp. 15453-15466, November 2024, doi: 10.1109/TNNLS.2023.3286943.
 - [23] Z. Wu, S. Song, A. Khosla, et al, "3D Shapenets: A Deep Representation for Volumetric Shapes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June 2015, pp. 1912-1920.
 - [24] Á. González, "Measurement of areas on a sphere using Fibonacci and latitude-longitude lattices," *Math. Geosci.*, vol. 42, no. 1, pp. 49-64, 2010.