MS-Transformer: Masked and Sparse Transformer for Point Cloud Registration

Qingyuan Jia, Guiyang Luo, Quan Yuan, Jinglin Li, Congzhang Shao and Ziyue Chen
State Key Laboratory of Networking and Switching Technology
Beijing University of Posts and Telecommunications
Beijing, China
{qyjia, luoguiyang, yuanquan, jlli, shaocongzhang, bupt_czy}@bupt.edu.cn

Abstract—In this paper, we propose a masked and sparse transformer to address the problem of point cloud registration with low overlap. The mask mechanism reduces the overall data, increasing the corresponding point ratio in the overlap region, while also reducing the computational cost to accelerate the algorithm’s execution speed. Moreover, we combine spatial position encoding and sparse self-attention to establish relationships within the source point cloud, as well as the relationships and attention scores between the source and target point clouds. This approach is specifically designed for the task of point cloud registration. Finally, we search for the maximum overlap area by matching the spatial consistency between points and calculate the 3D transformation matrix to complete the registration process. Our method achieves an improvement in the inlier ratio and performs well on the 3DMatch and 3DLoMatch datasets, demonstrating high registration efficiency.

Index Terms—Point cloud registration, Self-attention, Mask mechanism

I. INTRODUCTION

Point cloud registration involves estimating transformations between point clouds. By calculating the coordinate transformations, the process unifies the point cloud data from different viewpoints through rigid transformations such as rotation and translation to a specified coordinate system. Point cloud registration finds applications in various fields, including 3D reconstruction, parameter evaluation, localization, pose estimation, and autonomous driving [1], [2]. Autonomous driving relies on various sensors, with LiDAR [3] and RADAR [4] as important sensors to generate point cloud data. Combined with multi-view images captured by the camera [5], point cloud registration can lay an accurate foundation for subsequent segmentation, tracking [6], positioning [7], and target detection [8] tasks. Recently, the progress in point cloud registration has mainly been based on machine learning and correspondence-based methods [9]–[12]. Learning-based methods mostly extract the correspondence between two input point clouds by training a neural network and then calculating the pose transformation using a robust estimator such as RANSAC [13]. Correspondence-based methods rely mostly on keypoint detection, but this approach performs poorly in situations with low overlap because it is challenging to find repeatable keypoints in two point clouds. Learning-based methods have performed well in datasets with high overlap, and various state-of-the-art methods have achieved over 95% registration recall rate on 3DMatch [14] and KITTI [15] datasets. However, the accuracy significantly decreases in situations with low overlap, where the best-performing algorithm achieves only a 74% registration recall rate on the 3DLoMatch dataset. We have identified two factors contributing to this issue. Firstly, the presence of irrelevant data in the non-overlapping areas affects the registration process. Secondly, the number of matching points and corresponding features in the overlapping area is also reduced.

VIT [16] has successfully applied the transformer in image processing by dividing the image into small blocks and training them in the network. Similarly, in point cloud registration, after subsampling, superpoints which are groups of nearby points, can be used to form small patches. Sparse and loose superpoint matching reduces the strict point matching to overlapping blocks, thereby relaxing the requirement for
in the source point cloud, the feature similarity is calculated coordinate data as features. Secondly, for each coordinate point features from the raw coordinate data of the source point.

A. Correspondence-based Registration

Our method has demonstrated the effectiveness of our approach in terms of accuracy levels. Extensive experiments on indoor benchmarks and other comparable methods while maintaining comparable robustness and accuracy, all without relying on RANSAC. Consequently, our approach surpasses the speed of RANSAC and other comparable methods while maintaining comparable accuracy levels. Extensive experiments on indoor benchmarks have demonstrated the effectiveness of our method. Our main contributions are:

- We present a fast and accurate point cloud registration method that relies on point correspondences, resulting in a significant improvement in the inlier ratio.
- We introduce a registration transformer based on sparse attention, which effectively learns the correlation between superpoints in registration scenarios.
- Through evaluation of various datasets, we demonstrate good registration accuracy and efficiency. Our method achieves well-aligned results even with small amounts of data.

II. RELATED WORKS

A. Correspondence-based Registration

The correspondence-based matching method first extracts features from the raw coordinate data of the source point cloud and the target point cloud, or directly uses the raw coordinate data as features. Secondly, for each coordinate point in the source point cloud, the feature similarity is calculated in a point-to-point manner using the features extracted from both point clouds. Then, the feature similarities are sorted and the point pair with the highest similarity is selected as the matching point pair for each point in the source point cloud to the target point cloud. Finally, the three-dimensional rigid body transformation matrix is solved using the matching point pairs based on singular value decomposition. In order to achieve higher registration accuracy, the above solution process is usually carried out iteratively, and the three-dimensional rigid body transformation estimated from the previous iteration will be used as the initial transformation for the next iteration. The iterative closest point [22] algorithm was the most basic and classic method for 3D point cloud registration. Later works improved the robustness of the algorithm to some extent by manually designing features to describe local geometric shapes [23], [24]. The above methods are all traditional methods and have relatively average registration accuracy in registration scenes where there is overlap in content. After the widespread application of deep learning, convolutional networks are used as mapping functions to adaptively extract high-dimensional features from point cloud coordinate information, while using real labels as strong constraints in the network training process. The earliest deep learning algorithm DCP [17] replaced the hand-designed feature extractor in the traditional method with DGCNN [20] and mimics the algorithm flow of ICP [25] for deep nearest neighbor point iterative calculation, which improves the registration performance compared to the traditional method. PRNet [19] improves on this basis. DeepGMR [26] used a Gaussian mixture model for deep learning registration and implements a ”point-distribution” matching strategy. D3Feat [9] performs well in indoor scenes, utilizing the point cloud feature extraction backbone network KPConv [27] and redesigning the detector and descriptor. PREDATOR [11] combines attention mechanisms to establish connections within and between point clouds and mainly considers point cloud registration in low-overlap situations based on D3Feat. Geotransformer [28] embeds geometric information into attention, resulting in further improvement.

B. Global Features-based Registration

Registration methods based on global features estimate the 3D rigid transformation parameters by utilizing the global information of the entire point cloud region, including both overlapping and non-overlapping areas. These methods are relatively new and all based on deep learning algorithms. Their advantage lies in the ability to avoid point correspondences mismatches caused by local receptive fields, insignificant geometric features, or strong noise interference by utilizing the global information of the point cloud. PointNetLK [29] was the first to propose a deep learning-based global feature registration method, utilizing PointNet [30] as the backbone network. PCRNet [31] improved upon PointNetLK in terms of noise sensitivity. FMR [32] restored the coordinate information of the input point cloud by the decoder and performs global feature registration. REGTR [33] is an end-to-end point cloud registration model that utilizes transformers to predict the
be solved by the following formula:

\[ R^*, T^* = \arg\min_{R, T} \sum_{i=1}^{N} ||Rx_i + t - \phi(x_i, Y)||^2, \]

where \( R \) is the rotation matrix, \( T \) is the translation vector, \( x_i \) is a point in the source point cloud, \( Y \) is the target point cloud, and \( \phi \) is a distance function. The specific transformation can include rotation \( R \) and translation \( T \), which we denote as the source and target. The objective of point cloud registration is to recover the unknown rigid transformation consisting of a rotation \( R \in SO(3) \) and translation \( t \in \mathbb{R}^3 \) that aligns \( X \) to \( Y \). The specific transformation can be solved by the following formula:

\[ R^*, T^* = \arg\min_{R, T} \sum_{i=1}^{N} ||Rx_i + t - \phi(x_i, Y)||^2, \]

we need to first establish point correspondence between two point clouds, and then estimate the alignment transformation.

### A. MS-Transformer

#### Point sampling and masking.

We adopt the KPConv-FCNN [27] model as the backbone network for point cloud downsampling, which accompanies point features during the downsampling process. The reason for downsampling point clouds is that a large number of points in point cloud registration are useless and can be matched using coarser-level correspondences. Over-clustered points may not produce good results. Similar to [18], [28], we refer to the sampled points at the lowest level as superpoints. Unlike previous works, we adopt the idea from [21] and perform further masking by randomly masking 50% of the superpoints. The remaining superpoints in the source and target point clouds are referred to as \( S' \) and \( T' \), respectively, and their corresponding learning features are denoted as \( F_{in}' \) and \( F_{in}' \), respectively. Let \( x_i \) and \( f_i \) represent the i-th point and feature in the point cloud, and the convolutional kernel \( g \) in \( x \) is defined as:

\[ (F_{in} * g) \sum_{x_i \in N_e} g(x_i - x) f_i, \]

where \( N_e \) is the radius domain of point \( x \), and \( x_i \) is a support point in this domain. In addition to the coarsest separation rate of superpoints, referring to the structures of [9], [37], we also added dense points corresponding to the first layer of downsampling, which can be used to construct a local patch around each superpoint using a point-to-point node grouping strategy.

#### Sparse self-attention.

We designed a sparse self-attention structure that incorporates geometric spatial structure to learn the feature space and global correlations between superpoints in a point cloud. The attention mechanism involves huge computational costs when dealing with a large number of inputs [38]. Reducing attention computation has been a problem that researchers have been studying. Although point clouds contain a large number of points, most of them are useless in the case of low overlap. Their involvement in the calculation will instead affect the subsequent results. The extracted superpoints only need to establish local relationships with nearby points and global relationships with some distant points.

### C. Registration Datasets

In the field of 3D point cloud registration, there are various public datasets available for different scenarios. For instance, for synthetic scenes, commonly used public datasets include the ModelNet40 [34] dataset and the ShapeNet [34] dataset. For indoor scenes, commonly used public datasets include the 7Scenes dataset and the 3DMatch [14] dataset. For outdoor scenes, commonly used public datasets include the KITTI [35] dataset, and the Oxford dataset [36]. However, at present, the availability of datasets for low-overlap registration is limited, with only the 3DMatch dataset being available. Furthermore, most existing methods are primarily designed for high-overlap point cloud registration. This indicates that there is still considerable potential for improvement in the field of low-overlap point cloud registration. It would be beneficial to propose additional datasets tailored to different scenarios to address this gap.

### III. Method

#### Problem Definition: Given two point clouds:

\[ X = \{ X^i \in \mathbb{R}^3 | i = 1, \cdots, N \}, \]

\[ Y = \{ Y^i \in \mathbb{R}^3 | i = 1, \cdots, M \}, \]

which we denote as the source and target. The objective of point cloud registration is to recover the unknown rigid transformation consisting of a rotation \( R \in SO(3) \) and translation \( t \in \mathbb{R}^3 \) that aligns \( X \) to \( Y \). The specific transformation can be solved by the following formula:
The specific design is shown in the figure. We first encode a triplet with angles to replace the original position encoding for a learnable replacement [28]. We use Euclidean distance to measure the distance between two superpoints, \( p_{i,j} \). Then the position encoding for a pair of points can be calculated as:

\[
\begin{align*}
D_{i,j,2k} = \sin \frac{d_{i,j}/\sigma_d}{1000^2/d_k}, \\
D_{i,j,2k+1} = \cos \frac{d_{i,j}/\sigma_d}{1000^2/d_k},
\end{align*}
\]

where \( d_k \) is the feature size and \( \sigma_d \) is the hyperparameter controlling the distance change. After obtaining the distance between two points, the algorithm for triplets is the same. Next, we introduce sparse attention [39] by applying sparse connections on \( K \) and \( V \) through the sparse features \( K_S \) and \( V_S \), which reduces the complexity of \( QK^T \). Then, we obtain the feature results for that moment and merge all the results together to obtain the final \( \text{Atten}(X, S) \) [40]. This way, the attention has the characteristics of local dense correlation and remote sparse correlation:

\[
\text{Atten}(X, S) = \{a(x_i, S_i)\}_{i \in \{1, \ldots, n\}},
\]

\[
a(x_i, S_i) = \text{softmax} \left( \frac{(W_q x_i) K_T^T}{\sqrt{d}} \right) V_S,
\]

\[
K_S = (W_k x_i)_{i_i}, V_S = (W_v x_j)_{i_j}.
\]

**Feature cross-attention.** The cross-attention module is widely used in point cloud registration tasks [28], [33], [41], for feature exchange between the source and target point clouds. After obtaining the feature matrices \( F^X \) and \( F^Y \) through self-attention, calculate the cross attention feature matrix \( Z^X \) of \( X' \) with \( Y' \) feature:

\[
Z^X_i = \sum_{j=1}^{\lvert Y' \rvert} A_{i,j} (F^Y_j W^V),
\]

\[
a_{i,j} = \frac{(F^X_i W^Y) (F^Y_j W^Q)^T}{\sqrt{d_i}}.
\]
$C_i$. Ultimately forming a globally dense point correspondences $\mathcal{C} = \bigcup_{i=1}^{C_i} C_i$.

C. Loss Function

We use two loss functions to monitor the ground truth relationships: $\mathcal{L} = \mathcal{L}_s + \mathcal{L}_p$, which $\mathcal{L}_s$ is superpoint matching loss and $\mathcal{L}_p$ is point matching loss.

**Superpoint Matching Loss.** We use circle loss [44] for superpoint matching, measuring different ground conditions and superpoint correspondence by using the overlap rate near the superpoints [9], [11]. Select a set of superpoints from $Y$, and if there is more than 10% overlap with the selected superpoints in $X$, it is considered a positive sample. Otherwise, it is called a negative sample [45]. The set with at least one positive sample is called set $N$. For each $X$ in the set, the positive samples in $Y$ are called $C(Y)$ and negatives as $\mathcal{E}(i)$. So the superpoint matching loss can be defined as:

$$\mathcal{L}_s^X = \frac{1}{N} \sum_{g_i} \log[1 + \sum_{g_j \in \mathcal{E}(i)} e^{\lambda/\beta_s^2(d - \Delta p)} + \sum_{g_j \in \mathcal{E}(i)} e^{\beta_s^2(\Delta n - d_i)}],$$

where $d$ is the distance in the feature space, $\lambda$ indicates the overlap ratio between two patches, $\beta$ is the positive and negative weight of each sample [46]. Set margin hyper-parameters Default setting is $\Delta p = 0.1$ and $\Delta n = 0.1$. In this way, patches with high overlap become more important [47]. The reverse loss $\mathcal{L}_s^Y$ on $Y$ is computed in the same way. For total superpoint matching loss is $\mathcal{L}_s = (\mathcal{L}_s^X + \mathcal{L}_s^Y)/2$.

**Point Matching Loss.** We use negative logarithmic likelihood loss on the allocation matrix of superpoints [28], [41]. During the training process, we do not use the predicted correspondence, but rather randomly sample the $C_i$ ground truth superpoint correspondence $\{C_i\}$, $M_i$ is the set of ground-truth point correspondences. Using $P_i$ and $Q_i$, respectively to represent the set of mismatched points in two patches, we can compute the individual point matching loss as:

$$\mathcal{L}_p, i = - \sum_{(u,v) \in M_i} \log z_{i,u,v} - \sum_{u \in P_i} \log z_{i,u,m_i+1} - \sum_{v \in Q_i} \log z_{n_i+1,v}. \tag{12}$$

The total point matching loss is the average of all individual losses $\mathcal{L}_p = \frac{1}{|y|} \sum_{i=1}^{C_i} \mathcal{L}_p, i$.

IV. EXPERIMENT

A. Dataset

We evaluated the performance of our method on indoor 3DMatch [14] and 3DLoMatch datasets. The 3DMatch dataset is divided into three sets, namely training, validation, and test sets, each consisting of 46, 8, and 8 scenes, respectively. We used preprocessed data from Predator and followed its split, where the 3DLoMatch dataset includes only scenes with overlap ratios between 10-30%. Each input point cloud in the dataset contains no more than 30,000 points and was downsampled using the KPConv [27] backbone and augmented with small rigid perturbations during training.

B. Evaluation Metrics

Similar to most of the current point cloud registration works, we evaluated the performance of our methods using the standard metrics of 3DMatch, which mainly includes three metrics [9], [11]: 1) registration recall (RR), the most important metric that calculates the root mean square error of the transformed correspondences to measure the success of registration (defined as the RMSE of corresponding point pairs should be less than 0.2); 2) feature matching recall (FMR), which measures the score of point cloud pairs to judge whether a pair of point clouds is suitable for registration and evaluate the potential success of registration; 3) inlier ratio (IR), which is the ratio of the number of point pairs in the overlapping area used for matching to all the point pairs, and if the distance between two points is less than 10 cm in the true transformation, it is considered as an inlier.

C. Implementation

We conducted experiments on our method in the PyTorch environment, trained for 20 epochs, using a single NVIDIA RTX3090 GPU with 24GB memory, and an Intel 12900k CPU. The training took approximately 24 hours. For data augmentation, we added 0.005 noise perturbation and proportional rotation perturbation, with a batch size of 1. We used an initial learning rate of $10^{-4}$, weight decay of $10^{-6}$, and the learning rate was decayed by a factor of 0.05 after each epoch. The backbone network underwent two down-samplings, and the transformer alternated between four layers of self-attention and cross-attention. We used a multi-head attention mechanism with 4 heads and 32 groups of superpoints. For the superpoint matching loss, we calculated it using up to $n_p = 64$ positive pairs and selected the top $k = 3$, using the Sinkhorn algorithm for 100 iterations. When compared to RANSAC, we performed 1000 iterations.

D. Registration Results

**Comparison with recent methods.** We compared our method with some recent methods and feature-based registration methods. In Table 1, we show the matching and registration results of 3DMatch and 3DLoMatch using 5000 points/correspondence. It can be seen that the FMR [32] ratio of recent methods is high, and the registration recall rate is also high on the 3DMatch dataset. Our method is slightly lower than CoFiNet [18], but in the low overlap 3DLoMatch, the recall rate of other methods decreases significantly, and our method is in a leading position. In terms of the inlier ratio, our method outperforms other methods by a large margin, reaching more than 20 points, which is in line with our expectations and indicates that our method is better at dealing with difficult situations.

**Scenes results.** We also conducted separate experiments on different scenes within the 3DMatch and 3DLoMatch datasets, which included a total of 8 scenes. In the 3DMatch dataset, FMR was generally high, and the Hotel_1 scene had the highest average inlier ratio, while the Home_1 scene had the highest registration recall rate, with a difference of up to 12%.
TABLE I
RESULTS ON THE 3DMATCH AND 3DLLOMATCH DATASETS

<table>
<thead>
<tr>
<th>#samples</th>
<th>3DMatch 5000</th>
<th>3DLLOMATCH 5000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature Match Recall(%)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3DSN [48]</td>
<td>95.0</td>
<td>63.6</td>
</tr>
<tr>
<td>FCGF [49]</td>
<td>97.4</td>
<td>76.6</td>
</tr>
<tr>
<td>D3Feat [9]</td>
<td>95.6</td>
<td>67.3</td>
</tr>
<tr>
<td>Predeator [11]</td>
<td>96.6</td>
<td>71.7</td>
</tr>
<tr>
<td>SpinNet [50]</td>
<td>97.6</td>
<td>75.3</td>
</tr>
<tr>
<td>YOHO [51]</td>
<td>98.2</td>
<td>79.4</td>
</tr>
<tr>
<td>CoFiNet [52]</td>
<td>98.1</td>
<td>83.1</td>
</tr>
<tr>
<td>Sparse-transformer(ours)</td>
<td>98.1</td>
<td>86.3</td>
</tr>
</tbody>
</table>

"Inlier Ratio(%)"

| 3DSN [48] | 36.0 | 11.4 |
| FCGF [49] | 56.8 | 21.4 |
| D3Feat [9] | 39.0 | 13.2 |
| Predeator [11] | 49.9 | 20.0 |
| SpinNet [50] | 47.5 | 20.5 |
| YOHO [51] | 64.4 | 25.9 |
| CoFiNet [52] | 49.8 | 24.4 |
| Sparse-transformer(ours) | 72.8 | 53.3 |

"Registration Recall(%)"

| 3DSN [48] | 78.4 | 33.0 |
| FCGF [49] | 85.1 | 40.1 |
| D3Feat [9] | 81.6 | 37.2 |
| Predeator [11] | 88.3 | 54.2 |
| SpinNet [50] | 88.6 | 59.8 |
| YOHO [51] | 90.8 | 65.2 |
| CoFiNet [52] | 89.3 | 67.5 |
| Sparse-transformer(ours) | 89.1 | 71.3 |

TABLE II
RESULTS IN DIFFERENT SCENES ON THE 3DMATCH AND 3DLLOMATCH DATASETS

<table>
<thead>
<tr>
<th>SCENES</th>
<th>3DMatch FMR IR RR</th>
<th>3DLLOMATCH FMR IR RR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kitchen</td>
<td>0.994 0.812 0.924</td>
<td>0.928 0.563 0.826</td>
</tr>
<tr>
<td>Home_1</td>
<td>0.994 0.817 0.98</td>
<td>0.903 0.502 0.678</td>
</tr>
<tr>
<td>Home_2</td>
<td>0.971 0.753 0.824</td>
<td>0.874 0.558 0.721</td>
</tr>
<tr>
<td>Hotel_1</td>
<td>1.000 0.834 0.973</td>
<td>0.963 0.655 0.890</td>
</tr>
<tr>
<td>Hotel_2</td>
<td>0.990 0.802 0.91</td>
<td>0.861 0.532 0.674</td>
</tr>
<tr>
<td>Hotel_3</td>
<td>1.000 0.831 0.855</td>
<td>0.796 0.578 0.690</td>
</tr>
<tr>
<td>Study</td>
<td>1.000 0.761 0.902</td>
<td>0.771 0.420 0.523</td>
</tr>
<tr>
<td>MIT_lab</td>
<td>0.949 0.707 0.911</td>
<td>0.806 0.454 0.700</td>
</tr>
</tbody>
</table>

Rotation invariance. We evaluated the rotational invariance of different positional embeddings in Table 3. During training, we introduced rotation perturbations [11], [28]. In different experiments, the performance of the self-attention with absolute coordinate embeddings (a) was significantly reduced, indicating that it failed to deal with variance changes due to transformations. However, although the performance of self-attention with relative coordinate embeddings (b) was not as good as that with absolute position embeddings in normal situations [54], the model’s results remained unchanged after adding rotation perturbations, indicating that relative position embeddings had no effect on the model [55]. Our method (c) demonstrated good rotational invariance relative to perturbations and had an effective effect on encoding spatial structure.

Experimental runtime. We ran the methods listed in the table on a machine equipped with a single Nvidia RTX 3090 GPU and an Intel 12900K CPU. All models were tested with a batch size of 1 and without CPU parallelism. The final time reported is an average taken over 1623 point cloud pairs in the 3DMatch test set. The “Data” column reports the time it takes to prepare the data, while the “Model” column reports the time it takes to generate descriptors from the prepared data. Although all of these methods use KPConv as the backbone, our model benefits from the masking mechanism and sparse attention, which significantly reduces computation time and alleviates the computational pressure of the transformer. Our model can also be experimented with on larger point cloud datasets in the future.

V. CONCLUSION

We present the MS-Transformer, a deep model specifically designed for low-overlap point cloud registration tasks. The model’s core features a sparse attention mechanism based on a masked transformer, complemented by spatial positional encoding and sparse self-attention. These components establish internal relationships and geometric consistency within the source point cloud. Moreover, cross-attention is employed to establish connections between the source and target point clouds, reducing the computational load of the transformer while enhancing registration accuracy without compromising speed. However, the issue of losing key matching points still persists due to the reduction in the number of matching points. In the future, our goal is to devise improved methods for selecting matching points and integrate point cloud registration with the utilization of labels from other visual tasks.


