Attention PointResNet: End-to-end Point Cloud Registration With Attention Residual Connections

Yousef Alborzi
University of Manitoba
66 Chancellors Cir, Winnipeg, MB R3T 2N2
alborzis@myumanitoba.ca

Abstract

The ability to perform accurate and efficient point cloud registration is critical for many real-world robotics applications, including object recognition, pose estimation, and 3D reconstruction. This process involves finding the spatial transformation that optimally aligns two or more point clouds by estimating the rotation and translation parameters that bring the point clouds into correspondence with each other. The performance of traditional methods, including Iterative Closest Point (ICP) and Fast Global Registration (FGR), in the point cloud registration task, is limited in terms of achieving high levels of accuracy. This paper introduces a new deep learning-based method that incorporates attention mechanisms within residual connections to improve the accuracy and robustness of point cloud registration. The proposed approach utilizes an extension to the PointNet architecture, a powerful deep neural network that can learn features directly from unprocessed point cloud data without relying on manual feature engineering. By integrating attention mechanisms within the extended PointNet architecture, we aim to enhance the performance of point cloud registration.

1. Introduction

3D point clouds are an incredibly powerful tool for robotics applications. By representing the environment as a dense set of 3D points, point clouds enable robots to quickly and accurately perceive their surroundings and make informed decisions based on that information. This makes them especially useful for tasks such as object recognition, path planning, and navigation, where having a detailed understanding of the environment is crucial. Point clouds are also highly versatile, as they can be generated using a variety of sensors, including lidar, RGB-D cameras, and structured light systems. Overall, the use of 3D point clouds in robotics represents a significant step forward in the field and holds enormous potential for the development of more advanced and sophisticated robotic systems in the future. Because point clouds are unorganized and don't have a structure like 2d images, to process them, they are typically converted to voxel grids. However, Voxelization can introduce errors and distortions in the data. PointNet [12] is the first of the series of works that use raw point clouds for the task of classification and segmentation.

Point cloud registration is an important technique in robotic vision systems. It is a crucial part of tasks such as simultaneous localization and mapping (SLAM), visual localization, and scene reconstruction. The process of registration involves identifying a rigid transformation that maps two or more point clouds to one another, such that they match and align with each other. Point cloud registration algorithms range from classic iterative optimization-based methods like Iterative Closest Point (ICP) [2] and its adaptations as well as more recent deep learning approaches. PCRNet [16] utilizes PointNet encoding for point cloud registration in a siamese type architecture to encode the point clouds and a fully connected network to find the rigid transformation that aligns the point clouds.

In this paper, we propose an improved extension of the PointNet architecture to achieve better point cloud feature extraction. We utilize attention-based skip connections to preserve important low-level features. We train and evaluate our model on the ModelNet40 [20] dataset. Our improved model outperforms PCRNet, achieving a lower rotation and translation error, while also converging in fewer epochs. In summary, our contributions are (1) introducing a new architecture for point cloud encoding to improve registration, based on attention mechanisms and residual connections, (2) a novel chamfer distance cycle loss to improve the generalization and convergence of the model and (3) presenting an evaluation of our proposed method and a comparison to classical and other deep learning approaches.

2. Literature review

In the field of 3D point cloud registration, the task of determining a rigid transformation between two point clouds is trivial if their correct correspondences are already known. However, in the case of automatic point cloud registration, this is not a straightforward process. Consequently, much of the research on point cloud registration focuses on first identifying accurate correspondences and subsequently determining the rigid transformation. This can be accomplished through the utilization of iterative optimization-based algorithms or feature-matching methods. Additionally, some researchers have investigated end-to-end learnable techniques that do not explicitly require a predefined set of correspondences to ascertain the rigid transformations. As such, 3D point cloud registration can be classified into three categories: optimization-based, feature-matching, and end-to-end methods.

2.1. Optimisation-based

One of the most widely acknowledged techniques employed in 3D point cloud registration is the Iterative Closest Point (ICP) method, as introduced in [2]. The ICP algorithm aims to minimize the least-squares distance between two point clouds through an iterative process that involves identifying correspondences between neighboring points and determining the rigid transformation necessary to align them. Over the years, several variations of the ICP algorithm have been investigated, aiming to address limitations in the original formulation by incorporating diverse schemes for point selection, point matching, correspondence weighting, and correspondence rejection [4, 13]. While ICP guarantees convergence to a local minimum, the accuracy of the resultant model is significantly influenced by the initial alignment of the point clouds and therefore is mostly used for fine registration and not global registration. Notably, an extension of the ICP algorithm, presented in [22], proposes a method that is capable of attaining a globally optimal solution. Although this approach presents the advantage of achieving a globally optimal solution, it is important to acknowledge that it comes with a trade-off in terms of computational complexity. The proposed method exhibits a relatively high computational burden due to its utilization of a branch and bound (BNB) technique for efficient exploration of the SE(3) space. As a result, the computational resources required for executing this approach may be substantial.

2.2. Local Feature matching

An alternative approach to point cloud registration is to extract local geometric features from the point cloud and identify correspondences based on feature similarity. This approach eliminates the need for accurate initialization and mitigates the issue of local minima. Researchers commonly employ techniques such as Point Feature Histograms (PFH) [15] or Fast Point Feature Histograms (FPFH) [14] to extract local features. Subsequently, optimization methods such as Random Sampling Consensus (RANSAC) [6] are utilized to identify correct correspondences by evaluating the similarity of the extracted features.

Recent advancements in point cloud registration involve the utilization of deep learning algorithms to extract rich features from point clouds and match them for correspondence. For instance, 3DMatch [25] employs a 3D ConvNet to extract features from voxelized 3D patches, identifies key points, and matches them using RANSAC. However, approaches that rely on RANSAC for hard matching assignment may not be suitable for integration into learning pipelines due to their non-differentiable nature.

In contrast, PPFNet [5] utilizes PointNet [12] to extract features from multiple local patches, which are then aggregated using a max-pool function to capture global context. RPMNet [23] incorporates Sinkhorn normalization to produce soft correspondences based on local and spatial features extracted using PointNet.

Furthermore, attention mechanisms and transformer architectures [17] have also been applied in point cloud registration. PCAM [3] achieves multi-level contextual information by fusing cross-attention matrices at different levels. Both DCP [18] and PRNet [19] employ transformer architectures to extract conditioned features from point clouds, and utilize soft correspondences and a differentiable SVD approach to estimate rigid transformations. Similarly, REGTR [24] utilizes transformer cross-encoders to predict overlapping points and their correspondences using an MLP decoder. These approaches leverage the power of deep learning and transformer architectures to enhance the accuracy and robustness of point cloud registration.

2.3. Global Feature alignment

In another approach to point cloud registration, some works skip the correspondence matching step and opt to align point clouds using global features instead. PointNetLK [1] uses PointNet encodings to extract global features and then utilizes the Lucas & Kanade algorithm [11] to find the rigid transformation all in an iterative fashion. FMR [10] uses the global feature distance of the point clouds instead of their geometric distance to optimize the pose. Similar to PointNetLK, PCRNet [16] utilizes PointNet encodings but uses Fully connected MLP regressors to estimate the final transformation. Our work extends PCRNet and utilizes attention residual connections to improve the feature extraction process while using the same MLP architecture to estimate rigid transformations from global feature encodings.

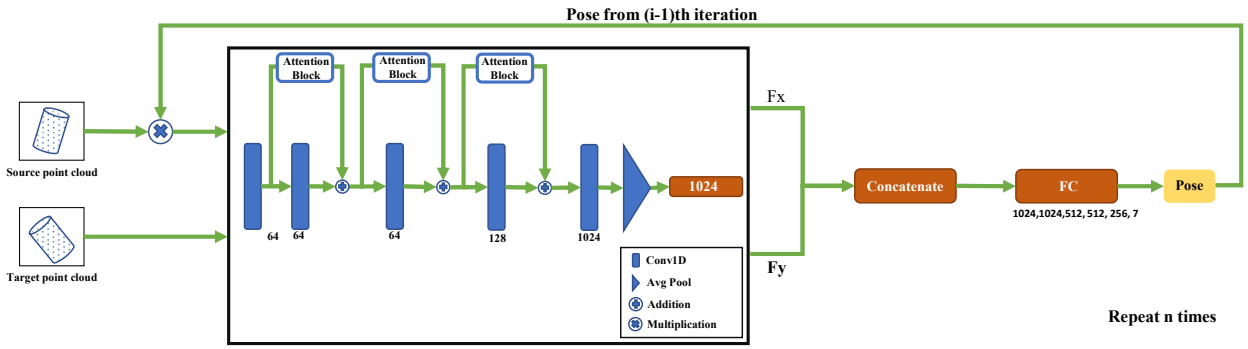

Figure 1. Complete iterative pipeline of the network. In each iteration, the source point cloud is transformed using the previous iteration's estimated rigid transformation. F_x and F_y represent encoded global features from the source and target point clouds respectively.

3. Problem Statement

Our objective is to determine an unknown rigid transformation R^*, t^* that aligns two point clouds, denoted as X and Y in $\mathbb{R}^{n\times 3}$, in a manner that minimizes the distance between them. We can express this goal mathematically as:

$$R^*, t^* = \underset{R\in SO(3), t\in\mathbb{R}}{\arg\min} \; d(X, RY + t) \tag{1}$$

In this equation, R^* and t^* represent the optimal rotation matrix and translation vector, respectively, selected from the set of all possible combinations of R and t. The goal is to find the combination that minimizes the distance metric d between the transformed point cloud $RY + t$ and the reference point cloud X.

4. Methodolgy

Similar to the methodology utilized in PCRNet [16], our network architecture incorporates PointNet [12] as the foundational structure for our global feature extraction. PointNet employs shared Multilayer Perceptron (MLP) layers to process each point within the point cloud. In practice, this can be effectively emulated using 1D convolutions. By leveraging these shared MLP layers, we obtain higher-dimensional representations for each individual point in the point cloud.

Subsequently, a pooling function is applied to aggregate all the points, resulting in a single representation that encapsulates the global characteristics of the point cloud. This pooling operation plays a critical role in generating a comprehensive feature representation that captures the overall state of the point cloud. Additionally, the symmetrical pooling function ensures permutation invariance, enabling consistent and reliable feature extraction regardless of the ordering or arrangement of the points.

4.1. Network architecture

The complete network pipeline is illustrated in Figure 1. Initially, the source and target point clouds are fed into the AttentionPointResnet Module, which utilizes PointNet-style encoding to extract global features. Within this module, a series of 1D convolutions is applied, followed by an average pooling function that aggregates the features and produces a single global feature for each point cloud. Notably, our contribution lies in the integration of skip connections within the PointNet architecture. These residual skip connections, inspired by [9], enhance the model's expressive power by preserving both low-level and high-level features. Additionally, by incorporating attention mechanisms within these skip connections, we can assign greater importance to features that have a more significant impact on the registration results.

As depicted in Figure 2, we use scaled dot-product attention as introduced in [17] within the attention block. The input is projected into three different spaces with the same dimensions, and energy scores are calculated using the key and query, followed by normalization of the results as expressed in Eq. (2).

$$a(Q, K) = \frac{QK^T}{\sqrt{d_k}} \tag{2}$$

Following normalization, the value projection is multiplied by the softmax of the energies, attention weights, and passed through a final projection and non-linearity. as expressed in Eq. (3). and Eq. (4).

$$Attention(Q, K, V) = \frac{\exp\left(a(Q, K)\right)}{\Sigma_{i=1}^{N} \exp\left(a(Q, K)\right)} V \tag{3}$$

$$FFN(x) = max(0, xW + b) \tag{4}$$

The resulting attended skip connection is added to the main output of the shared MLP layer and propagated to subsequent layers in the network.

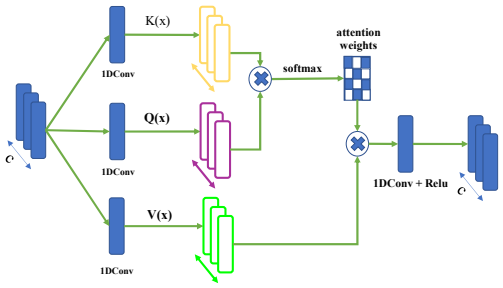

Figure 2. Attention block used in the architecture. $K(x)$, $Q(x)$, $V(x)$, represent the key, query, and value projections of the input respectively.

4.2. Loss Functions

To train our network we opt to use a combined rigid motion and cycle loss. Rigid motion loss is defined as the distance between the predicted and ground truth Rotation matrix and translation vector. In practice, we have noticed that in order for the network to achieve better results it is beneficial to give more importance to the rotation error, and therefore we parameterize our rotation error by λ_r. The rigid motion loss is defined in Eq. (5).

$$L_m = \lambda_r \|R_{XY}^T R_{XY}^* - I\|^2 + \|t_{XY} - t_{XY}^*\|^2 \quad (5)$$

where R and t are the predicted rotation matrix and translation vector aligning Point cloud X to point cloud Y while R^* and t^* are the ground truth. We empirically set λ_r to 1.5.

Additionally, we use Cycle consistency loss as used in [28], to encourage the model to learn the transformation in both ways and understand that the mapping from source to target point clouds is bidirectional. The cycle consistency loss is defined in Eq. (6).

$$L_c = \|R_{XY} R_{YX} - I\|^2 + \|t_{XY} - t_{YX}\|^2 \quad (6)$$

The total loss defined in Eq. (7) incorporates the two loss functions and uses another parameter to adjust the importance of rigid motion loss w.r.t cycle consistency loss.

$$L_t = L_m + \lambda_c L_c \quad (7)$$

4.3. Training

We train our network on the ModelNet40 [21] dataset, containing CAD Models of 40 classes mostly from synthetic household objects. Each object has 2048 data points and the dataset is split into a train set with 9840 objects and a test set of 2464 objects which we use for both validation and testing. We transform each source point cloud to a randomly sampled rotation using Euler angles in the range of $[0, 45]$ degrees and a translation in the range of $[-0.5, 0.5]$ on each axis, and our goal is to find the transformation that best aligns the source and randomly transformed point cloud. We Train Our network for 200 epochs using the ADAM optimizer with an initial learning rate of 0.001 which is multiplied by 0.1 every 50 epochs. It is also important to mention that we use 3 iterations for both our model and PCRNet. All experiments are performed on an NVIDIA GeForce RTX 2060 GPU and Intel Core i7 CPU @ 3.60 GHz.

5. Experiments

In this section, we will investigate the performance of our model and compare it to ICP [2], FGR [26], RANSAC [7] with FPFH [14] and PCRNet [16] in different experiments. For ICP, RANSAC and FGR the implementation in Open3D [27] is used and FGR and RANSAC share the same FPFH features. For ICP, we allow 100 iterations and we choose a threshold of 0.1 m for both ICP and FGR since this gives us the best results. We train our own PCRNet model, and therefore our results may be different from what is reported in [16]. We evaluate our work on the test set from ModelNet40 and report Root Mean Squared Error (RMSE) and mean absolute error (MAE) for both rotation and translation and we also report the mean isotropic rotation and translation errors as defined in Eq. (8).

$$Error(R) = \angle(R_{GT}^{-1}\hat{R}), \quad Error(t) = \|t_{GT} - \hat{t}\|_2 \quad (8)$$

However, using these metrics the model is penalized in the case of given alternate correct solutions for symmetric objects. Therefore we also report the mean Chamfer distance (CD) as defined in Eq. (9). CD is a fundamental metric utilized in point cloud registration to quantify the dissimilarity between two point cloud sets. It involves computing the squared distances between each point in one set and its nearest neighbor in the other set, and then summing these distances. In the case of a correct registration of the point clouds these metrics should all be close to zero.

$$d_{cd}(X, Y) = \sum_{x \in X} \min_{y \in Y} \|x - y\|_2^2 + \sum_{y \in Y} \min_{x \in X} \|x - y\|_2^2 \quad (9)$$

Finally using the isotropic rotation and translation errors we report the recall of each model and compare their Area Under the Curve (AUC).

In the following sections We conduct several experiments to evaluate the performance of our model in different settings.

Method	RMSE(R)	MAE(R)	RMSE(t)	MAE(t)	CD	Error(R)	Error(t)
ICP	25.0565	22.2392	0.0039	0.0033	0.0550	42.3357	*0.0067*
FGR	8.0063	6.7940	0.0173	0.0147	0.0058	12.3826	0.0299
RANSAC + FPFH	14.0333	11.7258	0.0325	0.0269	0.0173	20.0793	0.0564
PCRNet	*5.6540*	*4.6893*	0.0094	*0.0081*	*0.0030*	9.3470	0.0162
AttentionPointResnet(Ours)	**2.9603**	**2.3939**	**0.0018**	**0.0015**	**0.0010**	**4.9669**	**0.0030**

Table 1. Results on the unseen clean shapes. The best and second best results are marked in **Bold** and *italic* respectively.

Method	RMSE(R)	MAE(R)	RMSE(t)	MAE(t)	CD	Error(R)	Error(t)
ICP	25.7722	22.9029	*0.0039*	*0.0034*	0.0530	43.4826	*0.0068*
FGR	8.3803	7.1072	0.0180	0.0153	0.0060	12.7237	0.0312
RANSAC + FPFH	12.5457	10.3044	0.0278	0.0233	0.0133	18.3097	0.0482
PCRNet	*7.4626*	*5.9902*	0.0103	0.0088	*0.0052*	*12.4981*	0.0179
AttentionPointResnet(Ours)	**5.0023**	**4.0151**	**0.0026**	**0.0023**	**0.0026**	**8.3352**	**0.0046**

Table 2. Results on unseen categories. The model is trained on the first 20 categories and tested on the other 20 unseen categories.

5.1. Unseen shapes

In this experiment, the model is trained on clean data from the official training set and preprocessed as discussed in Sec. 4. The model is then tested on unseen shapes from the test set. As reported in Tab. 1, ICP fails in correctly registering the point clouds, due to the poor initial alignment of the point clouds. Furthermore, PCRNet outperforms FGR and RANSAC in all metrics and even achieves a smaller chamfer distance. Finally, our AttentionPointResnet achieve better results than the other methods in all metrics.

5.2. Unseen categories

To evaluate the generalizability of the model we conduct an experiment where the model is trained on the first 20 categories of the training data and tested on the other 20 unseen categories of the test data. Based on the results, reported in Tab. 2, as expected, there is no significant change in the performance of traditional methods, ICP and FGR, since they are not learning-based. Our two learning-based methods experience an increase in error metrics, being outperformed by ICP in metrics related to translation errors. While our AttentionPointResnet model experiences a decrease in performance, it still achieves better results than other methods.

5.3. Adding Gaussian noise

To evaluate the ability of our model to handle noisy data, in this experiment we add noise sampled from $\mathcal{N}(0, 0.01)$ and clipped in the range of $[-0.05, 0.05]$, similar to previous literature. The results presented in Tab. 3, show that FGR and RANSAC are extremely sensitive to noise and suffer marginally in performance. Also while ICP performs equally poorly as the previous experiments, the two learning-based methods do not show a large decrease in performance. Again our AttentionPointResnet model outperforms the other models performing twice as well as PCRNet and significantly better than the traditional methods.

6. Results and Discussion

As reported in the previous section our method shows resilience to noise and generalizability to unseen categories and shapes, outperforming the other tested methods.

In Fig. 3, we show the recall of the four mentioned models in the case of different thresholds for rotation. We evaluate results for rotations smaller than 45 degrees because any rotation error larger than that would be outside the range of random rotations we applied in the preprocessing step. As shown in Fig. 3 RANSAC has a higher recall for lower thresholds, outperforming other models. This means that RANSAC has more successful matches with finer registration. FGR also has a higher recall than our model in the lower thresholds, however, with the increase of the rotation threshold, our model begins to perform better and achieve higher success rates as compared to FGR and other models. This shows that FGR and RANSAC are able to generate registration estimations that are more accurate, however, these methods are not more robust than our method since they have less success with looser thresholds.

We use another metric deemed as Area-Under-The-Curve (AUC) to show the success of each model with varying rotation thresholds. For this metric, we calculate the area under each curve in Fig. 3 and normalize it to the range of 0 to 1 by dividing it by the rotation threshold range of 45. The results in Tab. 4 indicate that our model achieves the highest AUC and is overall the most successful in predicting correct rotation matches. Tab. 4 also shows that while

Method	RMSE(R)	MAE(R)	RMSE(t)	MAE(t)	CD	Error(R)	Error(t)
ICP	25.0878	22.2419	**0.0033**	**0.0029**	0.0559	42.3920	**0.0057**
FGR	43.1656	36.2656	0.0688	0.0574	0.0370	64.3014	0.1191
RANSAC + FPFH	69.1318	56.7046	0.1477	0.1205	0.0750	100.1127	0.2558
PCRNet	*5.4843*	*4.5311*	0.0133	0.0113	*0.0036*	*8.9706*	0.0230
AttentionPointResnet(Ours)	**5.2657**	**4.4307**	*0.0089*	*0.0078*	**0.0031**	**8.7693**	*0.0155*

Table 3. Results on unseen shapes with added Gaussian noise sampled from 0,0.01 and clipped at [-0.05,0.05]. The baseline chamfer distance using the ground-truth transformation is 0.00053

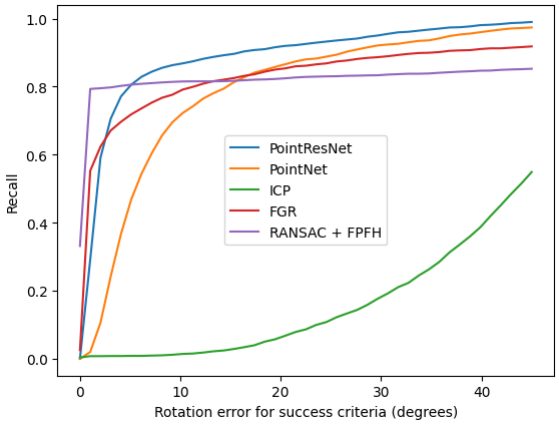

Figure 3. Overall registration recall(y-axis) on the ModelNet40 dataset with varying rotation threshold (x-axis).

PCRNet had lower rotation error metrics in general, as presented in Tab. 1, FGR and RANSAC have a higher AUC and are able to predict more correct rotation matches.

Example correct registration results from our model are shown in Fig. 5 and Fig. 4 shows two examples of incorrect registration results of our model. We notice that our model faces challenges with symmetric and semi-symmetric shapes, like the examples in Fig. 4. We think that this is because our loss function does not take into account symmetric shapes and penalizes the model even when it generates a correct registration. One solution to this problem can be adding a Chamfer loss to our loss function during training to account for symmetric shapes as well.

7. Conclusion and Future work

In this work, we propose AttentionPointResnet, a novel approach for point cloud registration based on global features. Our model incorporates attentional residual skip connections within the PointNet architecture to extract global features from a given point cloud, thereby enhancing the quality of the registration results. We conduct extensive experiments on the widely used ModelNet40 dataset to train and evaluate our model.

To assess the performance of our proposed method, we compare it against several registration models, including ICP, FGR, RANSAC, and PCRNet. We evaluate the models using isotropic and anisotropic metrics, capturing different aspects of registration accuracy. Through comprehensive experimentation, we demonstrate that our model consistently outperforms the other tested models across different evaluation metrics. These results highlight the effectiveness and superiority of our proposed AttentionPointResnet model for point cloud registration tasks.

To further evaluate the performance of our proposed feature extractor, there are several additional experiments that can be conducted. One possible enhancement is to replace the Multilayer Perceptron (MLP) head for rigid transformation estimation with an SVD module. This modification can

Method	AUC
ICP	0.1449
FGR	*0.8083*
RANSAC + FPFH	0.8026
PCRNET	0.7646
AttentionPointResnet(Ours)	**0.8658**

Table 4. AUC for recall vs rotation threshold on the ModelNet40. Our model has higher success in estimating correct rotations. The best and second best results are marked in **Bold** and *italic* respectively.

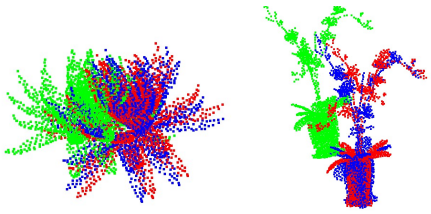

Figure 4. Incorrect registration results. Green: original source point cloud, Blue: ground truth target point cloud, red: estimated target point cloud

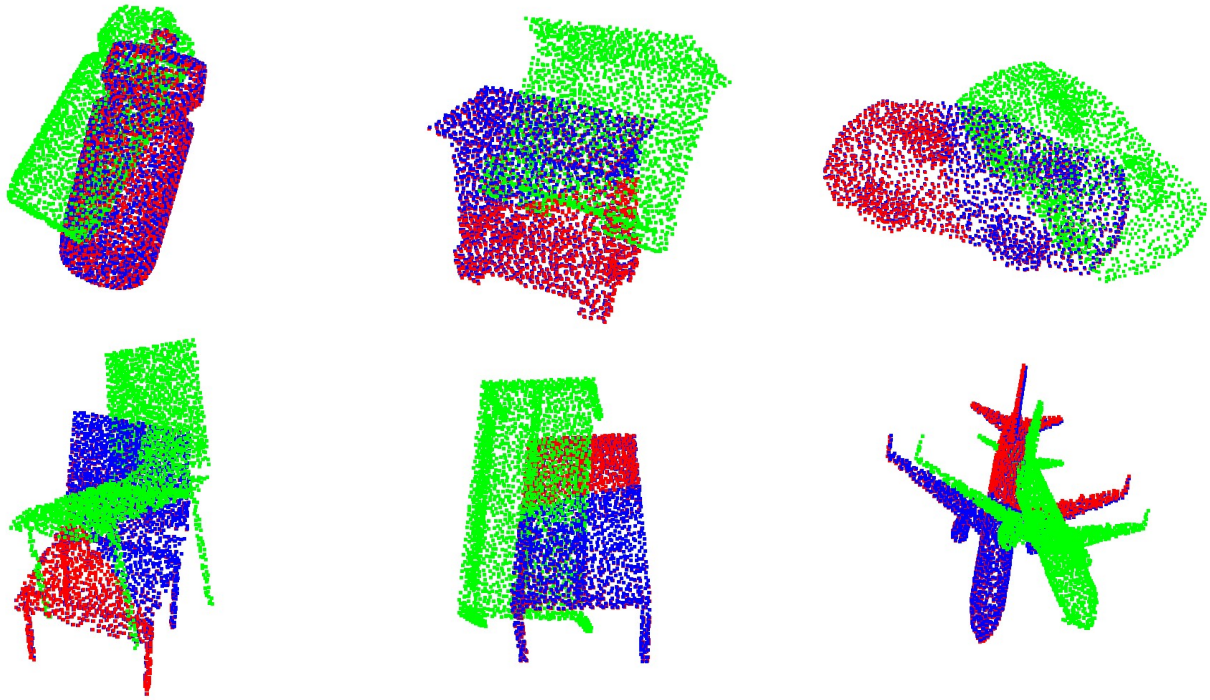

Figure 5. Correct registration results using our AttentionPointResnet model on samples from the ModelNet40 dataset. .

potentially improve the accuracy and stability of the transformation estimation process.

Moreover, it would be valuable to test our feature extractor on more challenging real-world datasets such as 3DMatch [25] and KITTI [8]. These datasets are known for their complexity and variability, providing a more rigorous evaluation of the model's performance under realistic conditions. Additionally, exploring the application of our feature extractor in other related problems, such as point cloud classification, would be beneficial. By leveraging the global feature extraction capabilities of our model, it could potentially yield improvements in the classification accuracy of point cloud data. Lastly, integrating our feature extractor into feature-matching approaches would enable an investigation of correspondence-based registration. By combining our global feature extraction with local feature matching techniques, we can explore the synergy between global and local information, potentially enhancing registration accuracy and robustness.

References

[1] Yasuhiro Aoki, Hunter Goforth, Rangaprasad Arun Srivatsan, and Simon Lucey. Pointnetlk: Robust & efficient point cloud registration using pointnet. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7163–7172, 2019. 2

[2] P.J. Besl and Neil D. McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992. 1, 2, 4

[3] Anh-Quan Cao, Gilles Puy, Alexandre Boulch, and Renaud Marlet. Pcam: Product of cross-attention matrices for rigid registration of point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13229–13238, 2021. 2

[4] Y. Chen and G. Medioni. Object modeling by registration of multiple range images. In *Proceedings. 1991 IEEE International Conference on Robotics and Automation*, pages 2724–2729 vol.3, 1991. 2

[5] Haowen Deng, Tolga Birdal, and Slobodan Ilic. Ppfnet: Global context aware local features for robust 3d point matching. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 195–205, 2018. 2

[6] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24:381–395, 1981. 2

[7] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 4

[8] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012. 7

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceed-*

ings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016. 3

[10] Xiaoshui Huang, Guofeng Mei, and Jian Zhang. Feature-metric registration: A fast semi-supervised approach for robust point cloud registration without correspondences. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11366–11374, 2020. 2

[11] Bruce D Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *IJCAI'81: 7th international joint conference on Artificial intelligence*, volume 2, pages 674–679, 1981. 2

[12] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation, 2017. 1, 2, 3

[13] S. Rusinkiewicz and M. Levoy. Efficient variants of the icp algorithm. In *Proceedings Third International Conference on 3-D Digital Imaging and Modeling*, pages 145–152, 2001. 2

[14] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (fpfh) for 3d registration. In *2009 IEEE International Conference on Robotics and Automation*, pages 3212–3217, 2009. 2, 4

[15] Radu Bogdan Rusu, Nico Blodow, Zoltan Csaba Marton, and Michael Beetz. Aligning point cloud views using persistent feature histograms. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3384–3391, 2008. 2

[16] Vinit Sarode, Xueqian Li, Hunter Goforth, Yasuhiro Aoki, Rangaprasad Arun Srivatsan, Simon Lucey, and Howie Choset. Pcrnet: Point cloud registration network using pointnet encoding. *arXiv preprint arXiv:1908.07906*, 2019. 1, 2, 3, 4

[17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. 2017. 2, 3

[18] Yue Wang and Justin M Solomon. Deep closest point: Learning representations for point cloud registration. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3523–3532, 2019. 2

[19] Yue Wang and Justin M Solomon. Prnet: Self-supervised learning for partial-to-partial registration. *Advances in neural information processing systems*, 32, 2019. 2

[20] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. 1

[21] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015. 4

[22] Jiaolong Yang, Hongdong Li, and Yunde Jia. Go-icp: Solving 3d registration efficiently and globally optimally. In *2013 IEEE International Conference on Computer Vision*, pages 1457–1464, 2013. 2

[23] Zi Jian Yew and Gim Hee Lee. Rpm-net: Robust point matching using learned features. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11824–11833, 2020. 2

[24] Zi Jian Yew and Gim Hee Lee. Regtr: End-to-end point cloud correspondences with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6677–6686, 2022. 2

[25] Andy Zeng, Shuran Song, Matthias Nießner, Matthew Fisher, Jianxiong Xiao, and Thomas Funkhouser. 3dmatch: Learning local geometric descriptors from rgb-d reconstructions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1802–1811, 2017. 2, 7

[26] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Fast global registration. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 766–782, Cham, 2016. Springer International Publishing. 4

[27] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3d: A modern library for 3d data processing. *arXiv preprint arXiv:1801.09847*, 2018. 4

[28] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017. 4