EANet: A Point Cloud Registration Network Based on EdgeConv with Dual Attention Mechanism

Chunrui Yang, Juan Zhu*, Xiaofeng Yue, Guolv Zhu
School of Mechanical and Electrical Engineering, Changchun University of Technology, Changchun, Jilin, China
**Corresponding Author*

Abstract: In machine vision, point cloud registration is one of the core elements, which has been applied to many fields such as robot localisation, medical image processing, and autonomous driving. The main problem solved by point cloud registration is to solve the rotation matrix and translation vectors from one point cloud to another. This paper proposes a point cloud registration network based on EdgeConv with spatial attention mechanism. EdgeConv can dynamically construct graph structure and build topological relationships within the point cloud, so that each point can obtain multi-level feature representation; the attention mechanism can capture contextual information and improve the accuracy of the registration. Experimental results show that EANet has higher registration accuracy, stronger generalisation ability and robustness compared to ICP, Go ICP, FGR, PCRNet and PointNetLK.

Keywords: 3D point cloud registration; deep learning; EdgeConv; Attention Mechanism; Machine Vision

1. Introduction

In the realm of machine vision, whose applications include 3D reconstruction, workpiece inspection, and robot material handling, point cloud alignment is a crucial component. The underlying idea behind point cloud alignment in the context of machine vision is that the source point cloud is acquired by the camera, the collected source point cloud and the pre-existing template point cloud are rotated and translated by the point cloud alignment algorithm, and the two point clouds are united under the same coordinate system, resulting in an error value of the two pieces of the point cloud that is less than the predetermined threshold value under the given error evaluation.

Researchers from all over the world have acknowledged a variety of deep learning techniques, such as autoencoder, recurrent neural networks, convolutional neural networks and so on. in the field of 2D photos, but they encounter particular difficulties while processing 3D point cloud data. Because 3D point clouds differ from 2D images in that they include replacement invariance and rotational invariance, neural networks—which are typically employed to analyse 2D images—will encounter significant difficulties while processing point cloud data. Therefore, it has become a popular area for research into how to efficiently analyze 3D point cloud data using neural networks. Researchers typically use the point cloud voxelisation approach to address this problem, which transforms point cloud data into normal voxel grids and feeds them into convolutional neural networks for processing. This method, however, results in more computing work and sparser data. A significant development in the field of point cloud deep learning and a solution to the issue of the mismatch between neural networks and point cloud data were brought about by the emergence of PointNet in 2016.The central concept of PointNet is the implementation of rotational invariance of point clouds through T-Net and the insertion of symmetric functions to achieve substitution invariance of point clouds. The first widely adopted deep learning network model for analyzing point cloud data is called PointNet. Later, a number of variation models based on PointNet, such PointNet++ and PointCNN, etc., appeared. A great variety of point cloud alignment techniques based on PointNet have developed as a result of the introduction of PointNet to the field of point cloud deep learning, including PointNetLK[1] and PCRNet[2] suggested in 2019. All of the

aforementioned models move the point cloud data to a high-dimensional space to extract features using the MLP in the PointNet model.To apply the optical flow LK algorithm, which is frequently used for image alignment, to the field of point cloud alignment, PointNetLK will treat PointNet as an imaging function.By analyzing the feature differences between two point clouds, it determines the rigid body motion parameters of a point cloud to achieve point cloud alignment. PCRNet employs PointNet to extract global features from the source and target point clouds, unlike PointNetLK, which treats PointNet as a feature extractor. Using the fully linked layer of PCRNet, the global features that were extracted using PointNet are directly fused. This results in the generation of a 1x7 feature vector.The last 3 bits of this feature vector represent the translation vector, and the first 4 bits represent the quaternion of the bit-pose transformation between two point clouds.The distinguishing feature of PCRNet is that it quickly aligns point clouds in this straightforward but effective manner, and it is significantly more resilient than PointNetLK.Through the use of PointNet, PCRNet recovers the global features and integrates them with the feature vectors produced by the FC layer. In order to accomplish correct point cloud alignment, this alignment concept may precisely estimate the positional transformation parameters between the source point cloud and the template point cloud.This approach has the advantage of avoiding the difficult iterative procedure and achieving an efficient alignment process by establishing the relationship between two point clouds directly from the feature space. Furthermore, PCRNet's robustness is a significant advantage in point cloud alignment, as it can handle point cloud data in complicated situations with noise, occlusion, and shape change. However, PCRNet mainly concentrates on the global features of the point cloud and overlooks the local features of the point cloud, resulting in PCRNet's inability to depict point clouds with complicated geometric features, and hence the alignment accuracy needs to be enhanced further.

To improve the registration accuracy of PCRNet, this paper proposes EANET, a novel point cloud registration network. To extract point cloud features, EANET employs the EdgeConv (DGCNN)[3]. EdgeConv can directly process point cloud data using this technique by considering it as graph-structured data and dynamically modifying the graph structure after each convolution operation. EANET successfully captures local properties of the point cloud while retaining replacement invariance by employing EdgeConv. EANET also has an attention mechanism in the feature fusion module. This addition considerably enhances the model's performance, resulting in improved registration accuracy for point clouds.

The following are our primary contributions:

Propose a new EdgeConv-based point cloud registration model that performs higher-level point cloud feature extraction by stacking EdgeConv modules to extract local point cloud features and employs the attention mechanism to increase model performance and registration accuracy.

Demonstrate the superior performance of the proposed model by comparing it with existing mainstream point cloud registration algorithms on a publicly available dataset and its data after adding noise.

2. Point Cloud Registration Algorithms

2.1 Traditional Point Cloud Registration Algorithms

Traditional point cloud registration algorithms usually consist of two steps: one is coarse, the other is fine registration. For coarse registration, the goal is to roughly align two sets of point clouds in order to provide an initial rotation matrix and translation vectors for the subsequent fine registration. Coarse registration methods can be classified into two main types: one is global search methods and the other is geometric feature description methods. Between them, global search methods, represented by the RANSAC algorithm and the 4PCS algorithm, align the two sets of point clouds by searching for a subset of possible matches in the point clouds to find the best rigid-body transformation model. Geometric feature description methods, represented by the FPFH algorithm, use geometric feature descriptors in coarse registration to quantify the similarity between point clouds.The FPFH algorithm is based on normal estimation of the point clouds and local feature histogram computation, and performs

the registration by comparing the similarity of the feature descriptors.

The Iterative Nearest Point (ICP) algorithm is a traditional method for fine-tuning point clouds. The technique finds the nearest point pairings between two pair of point clouds and iteratively updates the transformation matrix to minimize the distance between matched points. However, because the ICP algorithm is a non-convex optimisation problem, it is easily trapped in local minima, resulting in registration failure. Furthermore, the ICP algorithm is extremely sensitive to noise and outlier points, and it lacks resilience. Many scholars have enhanced the ICP algorithm to address the aforementioned issues. The Go-ICP[4] algorithm, which effectively overcomes the local minimum problem by introducing the branch-and-bound approach and the ICP algorithm alternately, is one of the significant advancements. However, it also lengthens the registration process. Furthermore, the Go-ICP algorithm is still affected by the beginning position of the point cloud.

2.2 Point Cloud Registration Algorithm Based on Deep Learning

Approaches for registering point clouds using deep learning are categorized into two groups: global feature-based approaches and point feature-point feature-based methods. The major distinction between these two approaches and conventional point cloud registration techniques is in the manner in which point characteristics are extracted. Neural networks are used to extract features from the point cloud and use those features in the matching process in point feature-point feature based registration approaches.This method is more expressive than conventional point cloud registration algorithms since it automatically learns point features using neural networks as opposed to other algorithms' proprietary feature extraction method. For instance, 3DMatch network, which uses a 3D convolutional network to extract local descriptors from input voxelised point cloud data. Another illustration is the DCP network. It solves 3D rigid body transformations using singular value decomposition (SVD) and uses DGCNN to extract point features.

In terms of algorithmic concepts, the global feature-based registration approach differs somewhat from the point feature-point feature-based registration method. The global feature-based registration technique extracts the global features of the point cloud through a neural network and solves the 3D rigid-body transform directly, as opposed to explicitly computing the correspondence between pairs of points. Through feature fusion and other techniques, this method registers the object. The first international feature-based registration technique is the PointNetLK network put forth by Aoki et al. in 2019. Through the use of a neural network to extract the point cloud's global features, this technique uses the classic LK algorithm to solve the 3D rigid-body transformation and perform point cloud registration. The global-feature based registration approach, in contrast to the point-feature-point-feature based registration method, executes the registration by collecting the global features of the point cloud rather than using an explicit point-pair matching process. The computational complexity is decreased and this method is more effective at handling large-scale point cloud data.

3. EANet Method

In this section, we will describe the EANet network topology and discuss the purpose of EdgeConv in point cloud registration networks, as well as its concepts.

3.1 EdgeConv

EdgeConv is a graph convolution operation for use in graph neural networks that enhances the representation of node features by considering edge information between nodes. Compared with the traditional graph convolution operation which only considers the connection relationship between nodes, EdgeConv can more fully utilize the neighbouring node features and edge features of a node.The computational procedure of EdgeConv is as follows:

- For each node p_i , obtain its set of neighbouring nodes $N(p_i)$.
- Calculate the directed graph $G(V,E)$, where V stands for the directed graph's vertices and E for its individual edges, for each neighbor node $p_j \in N(p_i)$.
- Denote the point feature of node p_i as $p_i = \Psi_{j:(i,j)\in\varepsilon} Y_\Theta(p_i, p_j - p_i)$ and the

edge feature of node p_i as $e_{ij} = Y_\Theta(p_i, p_j - p_i)$, $Y_\Theta : R^F \times R^F \to R^{F'}$, Y_Θ denotes a learnable nonlinear function containing Θ parameters, and Ψ denotes the symmetric aggregation operation.

- Assuming that the input to layer t is $P^t = \{p_1, p_2, \ldots p_n\} \subseteq R^F$, the directed graph of layer t is defined as $G^t(V^t, E^t)$, the edge features of layer t are $e^t{}_{ij} = h^t{}_\Theta(p^t{}_i, p^t{}_j - p^t{}_i)$, and the point features of layer t are $p_i{}^t = \Psi_{j:(i,j)\in\varepsilon^t} h^{t-1}{}_\Theta(p^{t-1}{}_i, p^{t-1}{}_j - p^{t-1}{}_i)$.
- Repeat the above process to achieve the construction of a dynamic graph structure and thus dynamic graph convolution.

The EdgeConv implementation process is shown in Figure 1.

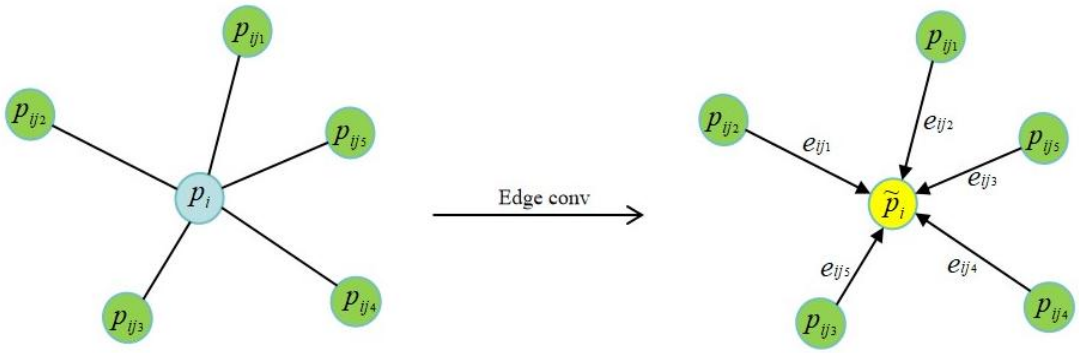

Figure 1. The Diagram of EdgeConv

3.2 Attention Mechanisms

The Channel Attention Mechanism is a technique for improving the performance of deep neural networks. Each channel in neural networks corresponds to a specific feature information extraction. However, not all channels are equally significant for the ultimate goal, and certain channels have characteristics that contribute more to the model's performance. By learning the weights of each channel, the channel attention mechanism allows the model to automatically focus on the channels that are more relevant to the task.

The Spatial Attention mechanism is used to improve the model's ability to attend to various spatial places in the input data. The spatial link between each point is critical during the point cloud registration process for comprehending the content and structure of the entire point cloud. When processing a point cloud, the Spatial Attention mechanism can assist the model in automatically focusing on significant spatial regions, enhancing model performance.

3.3 Network Structure

The EANet network topology corresponds to PCRNet, and the detailed network structure is depicted in Figure 2. Where the full name of CA is channel attention, it means the channel attention mechanism and the full name of SA is spatial attention, it means the spatial attention mechanism.

Suppose the source point cloud is $P_S = \{P_{Si} | i = 1, \ldots, n \subset R^3\}$ and the model point cloud is $P_T = \{P_{Ti} | i = 1, \ldots, n \subset R^3\}$. R means the rotation matrix. T means the translation matrix. Calculating $R_0 \in SO(3)$ and T_0 is the main purpose of point cloud registration. The error between the source point cloud $P_S = \{P_{Si} | i = 1,2,\ldots,n \subset R^3\}$ and the model point cloud $P_T = \{P_{Ti} | i = 1,2,\ldots,n \subset R^3\}$ will eventually reach a predetermined value after the rotation matrix $R_0 \in SO(3)$ multiplied by the stencil point cloud $P_T = \{P_{Ti} | i = 1,2,\ldots,n \subset R^3\}$, and the translation vector T_0 added.

To extract the point cloud features, EANet uses the EdgeConv module. First, it extracts

the features from the $N\times3$ point cloud and uses EdgeConv to integrate the point cloud's dimensions to $N\times64$, and then it extracts the features from the $N\times64$ point cloud and uses EdgeConv to integrate the dimensions to $N\times128$,and then take the point cloud's features, increase the dimensions in accordance with the aforementioned procedure, the remaining dimensions are $N\times256$ and $N\times512$, respectively,stitch the features extracted at each iteration together to produce the point cloud's $N\times1024$ dimensional features, and finally, by the Max pooling operation, integrate $N\times1024$ to 1×1024 to obtain the 1024 dimensional features of the point cloud F_{1024} .The precise formulas for extracting point cloud features are as follows:

$$F_{N\times64}=E(X_{N\times3}),F_{N\times128}=E(F_{N\times64}),F_{N\times256}=E(F_{N\times128}),F_{N\times512}=E(F_{N\times256}) \quad (1)$$

$$F_{1024}=Max(CA(Cat(F_{N\times64},F_{N\times64},F_{N\times128},F_{N\times256},F_{N\times512}))) \quad (2)$$

Where $X_{N\times3}$ denotes the original point cloud, $F_{N\times64}$ denotes the point cloud features mapped from 3 dimensions to 64 dimensions by EdgeConv, $F_{N\times128}$, $F_{N\times256}$, $F_{N\times512}$, $F_{N\times1024}$ denote the 128-dimensional, 256-dimensional, 512-dimensional, and 1024-dimensional point cloud features, respectively, the function $E(\cdot)$ denotes the EdgeConv module, and the function $Max(\cdot)$ denotes the maximum pooling. $CA(\cdot)$ denotes the channel attention mechanism, and $Cat(\cdot)$ denotes the point cloud feature splicing.

The feature $F_{X\,1024}$ is extracted from the source point cloud $X_{N\times3}$ and the feature $F_{Y\,1024}$ is extracted from the stencil point cloud Y_3 , the $F_{X\,1024}$ is fused with $F_{Y\,1024}$ using the concatenate operation to get the incorporated feature F_{2048} of the two point clouds, the F_{2048} is processed using the spatial attention mechanism (SA) to get the $F_{SA\,2048}$, and the $F_{SA\,2048}$ is inputted into a fully connected layer (FC) with dimensions of (1024, 1024, 512, 512, 256) fully connected layer (FC) with dimension 7 and the output layer for regression processing to obtain the 1×7 vector $P_{q,t}$ .The specific formula for the point cloud positional pose regression is as follows:

$$P_{q,t}=FC(SA(concatenate(F_{X\,1024},F_{Y\,1024}))) \quad (3)$$

The first four bits of $P_{q,t}$ are the quaternion q , and the last three bits are the translation vector t . Finally, the quaternion q is converted to the rotation matrix $R_0\in SO(3)$, and the point cloud is processed using $R_0\in SO(3)$ and t_0 to achieve point cloud registration.The iterative process of EANet is the same as that of PCRNet, and will not be repeated here. The specific formula for point cloud registration regression is as follows:

$$P_S=R_0P_T+t_0 \quad (4)$$

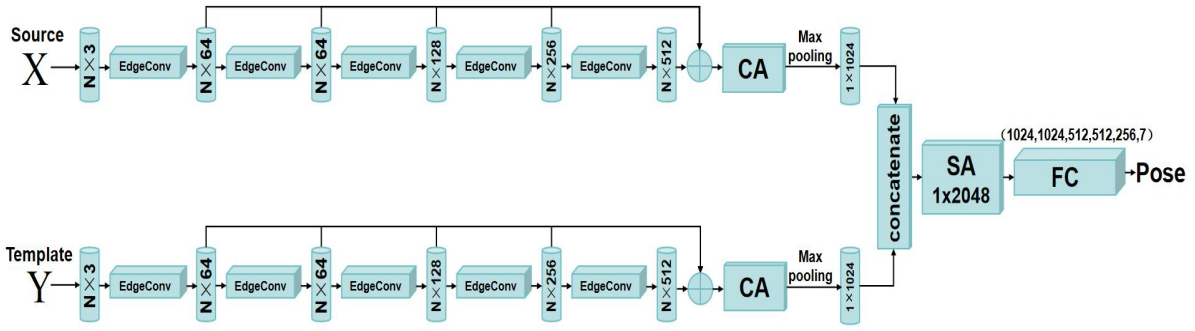

Figure 2. The Framework of the Network Model

3.4 Loss

The specific formula for the loss function of EANet is as follows:

$$\text{Loss} = \left\| R_{xy} R^{g}{}_{xy} - I \right\|^2 + \lambda \left\| t_{xy} - t_{xy}{}^{g} \right\|^2 \quad (5)$$

where R_{xy} and t_{xy} denote the predicted rotation matrix and translation vectors, $R^{g}{}_{xy}$ and $t^{g}{}_{xy}$ denote the really rotation matrix and translation vectors, and λ is the translation correction factor.

4. Experimental Results

Use ModelNet40 standard dataset to train the network model, running under Ubuntu 20.04 environment with PyTorch 1.10.0 as the deep learning framework. The Adam (Adaptive Moment Estimation) gradient descent optimiser was used with an initial learning rate of 0.001 and the learning rate was multiplied by 0.1 for every 50 epochs, with a batch_size of 24 and 250 epochs for both the training and test sets.For the hardware part, an NVIDIA A40 with 48G of memory was used.

In this section, we conduct tests to compare EANet to five well-known point cloud techniques. The experiments make use of data from[5] and. Performance is measured using the metrics RMSE(R) and MAE(R) for rotation error and RMSE(t) and MAE(t) for translation error. These metrics allow us to properly measure the discrepancy between the expected values and the actual values.

The results of EANet's point cloud registration are shown in Figure 3. The source point cloud is shown in green, the model point cloud is shown in red, and the registered point cloud is shown in blue in the figure.

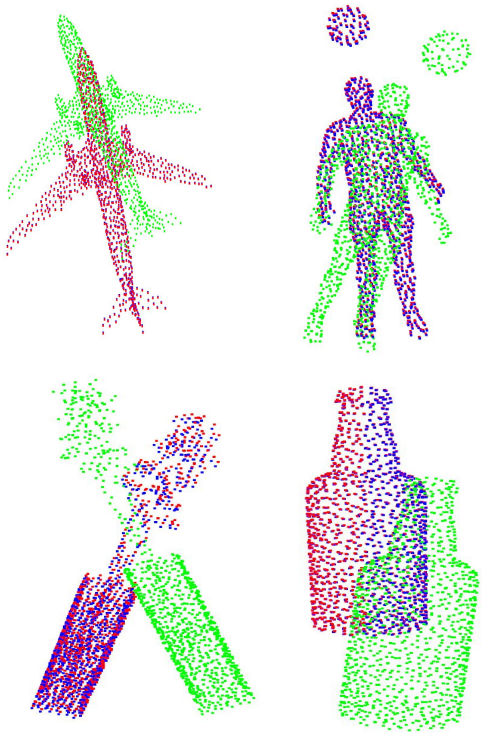

Figure 3. Registration Results

On the original dataset, Table 1 shows the experimental results for our model and five other widely used techniques using the evaluation metrics. In terms of RMSE(t) and MAE(t), where it performs much better than the other approaches, our model exceeds them in all four evaluation criteria. This shows that our method effectively improves point cloud registration accuracy while reducing translation error.

Table 1. Results from the Original Dataset

Model	RMSE(R)	RMSE(R)	RMSE(t)	MAE(t)
ICP	29.914835	23.544817	0.290935	0.248755
Go-ICP	11.852313	2.588463	0.025665	0.007092
FGK	9.362772	1.999290	0.013939	0.002839
PointNetLK	15.095374	4.225304	0.022065	0.005404
PCRNet	4.177078	2.628245	0.015526	0.010579
Ours	**2.642963**	**1.163153**	**0.004366**	**0.001428**

Table 2 displays the experimental findings of our model and the remaining five widely used algorithms on the dataset with invisible categories using the above evaluation techniques. In all four evaluation methods, our model produces the best results, but the results of our MAE(R) are comparable to those of FGK's MAE(R), and the results of this RMSE(t), MAE(t), are significantly better than those of the other five methods, indicating that our method has a stronger capacity for generalization.

Table 2. Results for the Invisible Category Dataset

Model	RMSE(R)	MAE(R)	RMSE(t)	MAE(t)
ICP	29.876431	23.626110	0.293266	0.251916
Go-ICP	13.865736	2.914169	0.022154	0.006219
FGK	9.848997	1.445460	0.013503	0.002231
PointNetLK	17.502113	5.280545	0.028007	0.007203
PCRNet	4.281822	2.693274	0.013912	0.009988
Ours	**2.771542**	**1.188554**	**0.005188**	**0.001991**

Table 3 displays the experimental findings for our model and the remaining five widely used algorithms utilizing the Gaussian noise datasets and the above evaluation techniques. Though our MAE(R) and MAE(t) results do not clearly distinguish us from the other techniques, our model still produces the best results using all four evaluation methods, demonstrating the method's superior noise immunity and robustness

Table 3. Results for Gaussian Noise Dataset

Model	RMSE(R)	MAE(R)	RMSE(t)	MAE(t)
ICP	29.707983	23.557217	0.290752	0.249092
Go-ICP	11.453493	2.534873	0.023051	0.004192
FGK	24.651468	10.055918	0.027393	0.002231
PointNetLK	16.004860	4.595617	0.021558	0.005652
PCRNet	4.219825	2.675704	0.015368	0.010528
Ours	**2.693031**	**1.163334**	**0.004649**	**0.001974**

5. Conclusions

EANet outperforms ICP, Go ICP, FGR, PCRNet, and PointNetLK in terms of registration accuracy, generalisation ability, and resilience. This demonstrates that combining EdgeConv with the spatial attention mechanism can significantly increase model performance, and it demonstrates that it is viable and effective to include a module that can extract local features during the point cloud feature extraction process.

Acknowledgments

This work was supported by department of science and technology of Jilin Province (20220203091SF).

References

[1] Aoki, Y., Goforth, H., Srivatsan, R. A., & Lucey, S. (2019). Pointnetlk: Robust & efficient point cloud registration using pointnet. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 7163-7172).

[2] Sarode, V., Li, X., Goforth, H., Aoki, Y., Srivatsan, R. A., Lucey, S., & Choset, H. (2019). Pcrnet: Point cloud registration network using pointnet encoding. arXiv preprint arXiv:1908.07906.

[3] Wang, Y., & Solomon, J. M. (2019). Deep closest point: Learning representations for point cloud registration. In Proceedings of the IEEE/CVF international conference on computer vision (pp. 3523-3532).

[4] Yang, J., Li, H., Campbell, D., & Jia, Y. (2015). Go-ICP: A globally optimal solution to 3D ICP point-set registration. IEEE transactions on pattern analysis and machine intelligence, 38(11), 2241-2254.

[5] Wang, Y., Sun, Y., Liu, Z., Sarma, S. E., Bronstein, M. M., & Solomon, J. M. (2019). Dynamic graph cnn for learning on point clouds. ACM Transactions on Graphics (tog), 38(5), 1-12.

