Chapter 2

A Rigid Point Cloud Registration Method Based on Global Feature Learning and Graph Representation

Sara Monji Azad
Mannheim Institute for Intelligent Systems in Medicine
Heidelberg University, Germany
sara.monjiazad@medma.uni-heidelberg.de
ORCID: 0000-0002-2742-9961

Marvin Kinz
Mannheim Institute for Intelligent Systems in Medicine
Heidelberg University, Germany
marvin.kinz@std.uni-heidelberg.de
ORCID: 0009-0008-8226-5905

Mehran Fotouhi
Department of Computer Engineering
Sharif University of Technology, Iran
mehran.fotouhi@alum.sharif.edu

Jürgen Hesser
Mannheim Institute for Intelligent Systems in Medicine
Interdisciplinary Center for Scientific Computing (IWR)
Central Institute for Computer Engineering (ZITI)
CZS Heidelberg Center for Model-Based AI
Heidelberg University, Germany
juergen.hesser@medma.uni-heidelberg.de

DOI: 10.54103/milanoup.282.c634

2.1 Abstract

Point cloud registration involves estimating spatial transformations between point sets. Traditional registration approaches are categorized into rigid and non-rigid transformations. This paper introduces a rigid point cloud registration method leveraging the PointNet method to learn global shape features, focusing on graph representation during the registration step. Informative points are extracted as global features, represented as graphs, and then matched to compute a transformation matrix for registration. Experimental results demonstrate that the proposed algorithm significantly improves the mean square error (MSE) from 4.94×10^{-5} to 1.74×10^{-5} on average across 10 categories of the ModelNet10 dataset. Additionally, the method achieves superior accuracy for the chair category in the ShapeNet dataset compared to GP-Aligner and Norm-IP, while also reducing computational time through the graph intermediate representation.

2.2 Introduction

A point cloud is a set of data points for the representation of 2D or 3D shapes. Point sets find their use in various applications, namely 3D localization [1], 3D reconstruction [2], free-viewpoint generation [3], augmented/virtual reality [4], etc.

Point cloud registration is one of the most fundamental challenging areas in computer vision [5]. The primary goal of point cloud registration is to estimate the transformation between two or more corresponding point sets. Hence, the error between the transformed point cloud and the target one should be minimized. The registration approaches are categorized into rigid and non-rigid transformations. To be more accurate, in rigid registration the geometric extensions and shape of an object are invariant under affine transformations such as translation and rotation while for non-rigid registration the deformation field for the source point cloud should be determined [6].

A point cloud registration method based on rigid transformation is proposed in this paper. The proposed approach consists of three main steps, namely feature extraction, corresponding point determination based on a graph representation, and the transformation matrix calculation. The first step is based on PointNet [7] to extract global features which are the informative points of the input point clouds. In the second step, a graph representation of the extracted informative points is constructed. In the mentioned graph, each point represents an informative point. To this end, graph matching is applied to find the initial corresponding points. Then, the initial correspondences are used to estimate the transformation matrix. All steps of the proposed method are explained in Section 2.5 in more detail.

A summarization of the proposed method's novel parts can be discussed in several points:

- Considering the fact that PointNet [7] is able to extract critical points which can show the structure of an object, the proposed method extracts the efficient landmarks of an object by taking the advantage of PointNet.
- The proposed method constructs a star graph based on the extracted landmarks which are efficient to determine the initial corresponding points.
- In the proposed approach, the registration problem is solved using the graph-matching approach to increase the time efficiency as well as improving the accuracy of the matching step.
- The proposed method calculates the system equation to determine the affine transformation matrix. The equation systems are computed based on precise initial corresponding points which increasingly decrease the transformation matrix error.

This paper makes the following contribution:

- The proposed method calculates the accurate transformation by using the graph-matching approach.
- Experiments show the proposed method is robust to different affine transformations (rotation and translation).
- The proposed graph intermediate representation reduces the computational time.

The rest of this paper is organized as follows. Section 2.3 outlines the related work. Section 2.4 provides the problem definition of rigid point cloud registration. The proposed method is explained in Section 2.5. The experimental dataset and evaluation performance are described in Section 2.6. A discussion on the potential and limitations of the proposed method is presented in Section 2.7. Finally, Section 2.8 concludes the paper.

2.3 Related work

The registration problem is typically tackled by methods that are using either non-learning approaches or learning-based methods. Non-learning approaches, in particular, suffer huge computational effort. Hence, using non-learning approaches to solve the registration problem in a real-time manner is the most significant challenge. There are various approaches to solving the registration problem based on learning approaches. Some of them focus on generating an accurate feature descriptor to take local and global features into account. For the same purpose, some others concentrate on finding the best representation for the point cloud.

From another point of view, learning-based methods use different network architectures. Some architectures are more common like MLP, CNN, and GNN. In the following, some of the most cited papers are studied. However, the available studies are not limited to the mentioned categorizations.

Given that point clouds are unordered structures that, unlike images, do not have neighborhood information, there are some approaches to provide different representations of point clouds that are used in various registration studies. The intermediate representation is a common approach to overcoming the unordered problem. Voxels, graphs, and 3DMeshs are some samples of the intermediate representation, to mention a few. In this way, applying CNN architecture, using point clouds as the network's input, can be a challenge [8]. There are different approaches to overcoming these challenges. Deep closet point (DCP) [9], TIF-Reg [10], and the deep global registration (DGR) method [11] are some studies of rigid point cloud registration based on the CNN network.

GNN is another frequently used network to learn neighborhood information. Usually, each graph's node represents its local features which is one of the reasons for achieving good accuracy. GNN is not only used as a local feature representation, but also as a matching step. FIRE-Net is a rigid and interactive representation learning-based method [12]. G^2 Net [13] is another geometric guided network for both full and partial rigid point cloud registration problems. Also, a graph matching method (RGM) is proposed in [14].

In another categorization, point cloud registration methods are categorized into coarse and fine approaches. The coarse-based registration methods calculate an initial geometric alignment. However, estimating a transformation to solve registration problems as precisely as possible is called the fine-based registration method. Iterative Closest Point (ICP) [15] and Coherent Point Drift (CPD) [16] are two of the most well-known fine registration approaches.

2.4 Problem Definition

A point cloud P is a set of a fixed number n of 3D points $P = \{p_i \in P | i = 1,2,...,n\}$. The source point cloud (S) and the target point cloud (T) are defined as a set of $Data = \{(S,T)|S,T \in \mathbb{R}^{N\times n}\}$ where N is the dimension and n is the number of points.

A transformation is a parameterized mapping from the source to the target domain, in particular, the transform is defined by a rotation matrix $R(\theta)$, and a translation vector $\vec{t}$; we define $\alpha = \{R(\theta), \vec{t}\}$ as the free parameters of the transform $Trans(\alpha, S)$ where S is the source point cloud that should be transformed.

The metric defines the similarity of point clouds (T), $Trans(\alpha, S)$; whereby for our purpose, we use the Euclidean distance measure as the criterion of best fit. In particular, we assume that for point cloud registration, for each $s_i \in S$ there is a corresponding $t_i \in T$ as the corresponding point.

2.5 Proposed Method

In our approach, point cloud registration is based on identifying the corresponding points in the source and destination point clouds, and calculating the

transformation matrix, subsequently.

As shown in Figure 2.1 our method consists of three main steps, namely feature extraction, corresponding point determination based on a graph representation, and transformation matrix calculation. In the following, each step is discussed in detail.

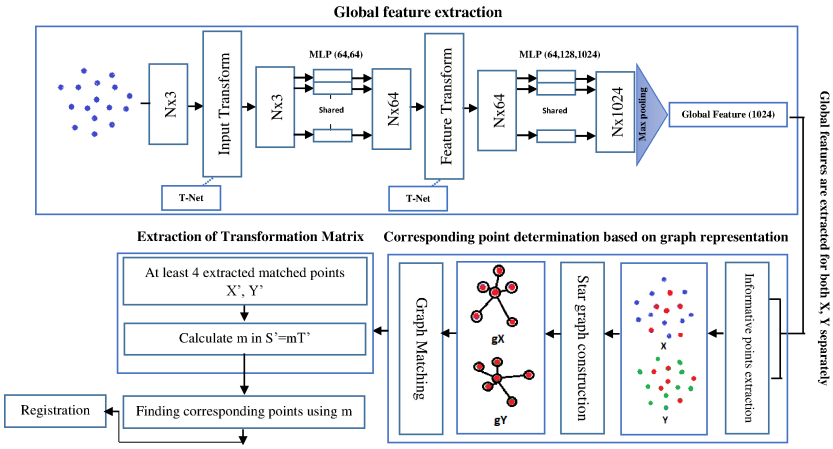

Figure 2.1 *The overview of the proposed method. The proposed method has three main steps: global feature extraction based on PointNet [7], corresponding point determination based on graph representation, and transformation matrix calculation.*

2.5.1 Feature Extraction

The learning-based feature descriptor is the first step of the proposed method. For the given point clouds, global features are extracted using a PointNet-based architecture. PointNet [7] is a very efficient method for feature extraction in learning-based registration methods.

The input is 3D point cloud $\{P_i | i = 1, ..., n\}$ where each point P_i is a vector $(x, y, z) \in \mathbb{R}^3$. The feature extraction step has two main modules, one is a T-net and the other one is a max pooling layer.

T-net [17] is a mini-network to predict an affine transformation matrix to map a point cloud to canonical space before processing. Then, the estimated transformation matrix is directly applied to the coordinates of input points. T-net is applied two times in our network architecture. For the first time, it is applied to $N \times 3$ coordinates, and the second time when feature vectors are transformed within the feature space with the dimension of $N \times 64$.

Max pooing as the second module is a function used to extract an order-invariant descriptor. It calculates the maximum value for patches of a feature map

and creates a downsampled or pooled feature map. In this paper, the output of the max pooling layer is called global features or informative points. Global features consist of a vector with 1024 elements which is calculated based on a $N \times 1024$ matrix from the previous layer, where N is the number of points in one point set. In Figure 2.2, the output of max pooling is shown.

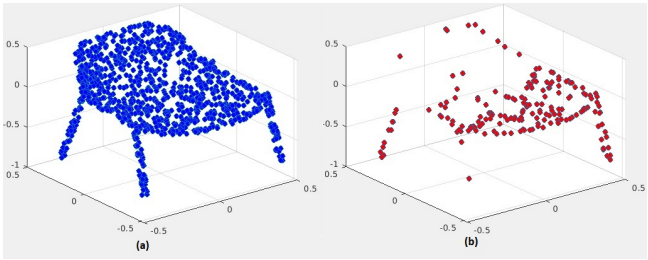

Figure 2.2 *Informative points representation to show the output of global shape features (a) Input point cloud (b) global feature representation*

2.5.2 Corresponding point determination based on graph representation

After extracting the global features from the first step of the proposed method, the extracted informative points are used to provide a new representation of the input point cloud. This representation is coded as a graph for source and target point clouds respectively. The informative points are hereby the vertices $v_i \in V$ of the graph $G = (V, E)$, and the edges $e_j \in E$ are the connectives between the central vertex to other vertices by defining a central node $v_c \in V$ and edges $e_j = (v_c, v_i), i \neq c, v_i \in V$. To reduce the computational time, instead of using a fully connected graph or a graph with a high number of edges, a star graph of informative points is constructed. Furthermore, using the median as a robust estimate for the point cloud center gives us the first certain corresponding point which is invariant under translation and rotation. The star graph is a type of graph in which the degree of $n-1$ vertices is 1 and the degree for one vertex is degree $n-1$. To be more precise, $n-1$ vertices are connected to a central vertex. A sample of the constructed star graph from the chair informative points is illustrated in Figure 2.3. For determining the corresponding points, star graphs for both source and target point clouds are constructed. Considering the fact that a rigid transformation (translation and rotation) does not change the distance between points, finding the edges v_c with unique weights can be used to estimate initial accurate corresponding points. Therefore, first, the unique edges (v_c, v_i) are extracted for both source and target point clouds, separately. Then, their vertices v_c can be considered as initial corresponding points.

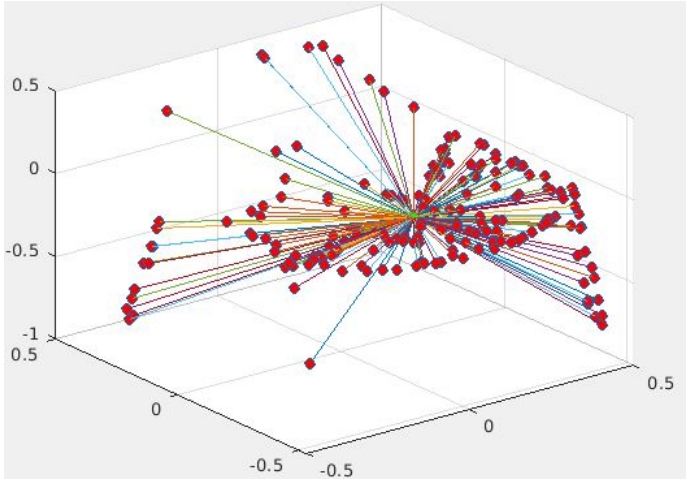

Figure 2.3 *A constructed star graph of the informative points of a chair*

2.5.3 Transformation matrix calculation

The transformation between the source and target point cloud is calculated through the initial corresponding points. Although at least four corresponding points are needed for calculating the transformation matrix, the number of unique edges is usually more than this value. Therefore, the limitation of the number of points is not a problem for the proposed method. To calculate the transformation matrix, the coordinates of the source and target point clouds are considered to be homogeneous. To be more precise, each point is shown as $(x, y, z, 1)$. Then, the transformation matrix m between the source and target point clouds is a 4 matrix which is a combination of various transformations.

$$m = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.1}$$

By considering S as the source point (or a vector) and T as the target point (or a vector), the matrix m presents an affine transformation that transforms S from one coordinate system to T in another coordinate system. The m shown in Equ. 2.1, is combined of translation, rotations, scalings/reections, and shears. As discussed in Section 2.4, this paper is limited to calculating the translation and rotation. In this regard, the elements [$a11$, $a12$, $a13$, $a21$, $a22$, $a23$, $a31$, $a32$, $a33$] represent the rotation transform and the elements of the last column [$a14$, $a24$, $a34$] represent the translation transform. To calculate the transformation between the source point set (S) and the target one (T), matrix m should be computed. Matrix m is obtained by

calculating the system equation which is

$$T = mS \tag{2.2}$$

The transformation of the point x to point x' is thus written as $x' = mx$. This mapping can be shown as a matrix by

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{2.3}$$

For calculating the elements of matrix m, the system of matrix equation that is shown in Equ. 2.2 needs to be solved. The equations can be shown by

$$\begin{cases} x' = a11x + a12y + a13z + a14 \\ y' = a21x + a22y + a23z + a24 \\ z' = a31x + a32y + a33z + a34 \end{cases} \tag{2.4}$$

These are the equations that define the relationship between the matrices T, m, and S. To find the matrix m, we can solve these equations for the elements of m in terms of elements of T and S. It is worth mentioning that points and vectors are represented as mathematical column vectors in homogeneous coordinates with the difference that points have a 1 in the fourth position whereas vectors have a zero of that. For vectors, the value in row number 4 is 0 instead of 1 to remove the translation operation by multiplying the 4th vector of matrix m by $0(1 = 0x+0y+0z+1)$.

After calculating m based on the accurate initial corresponding points, all points can be transferred from the source point cloud to the target one based on the calculated m. Then, the distance between the moving points and the target ones is calculated to show how accurate the proposed method is which will be discussed in Section 2.6.

2.6 Results

To show the accuracy of the proposed method, the approach is evaluated on the ModelNet10 dataset and the results are compared with a rigid registration method proposed in [18]. In the following, the used dataset to evaluate the accuracy of the method will be studied. Afterward, the results of the proposed method will be presented.

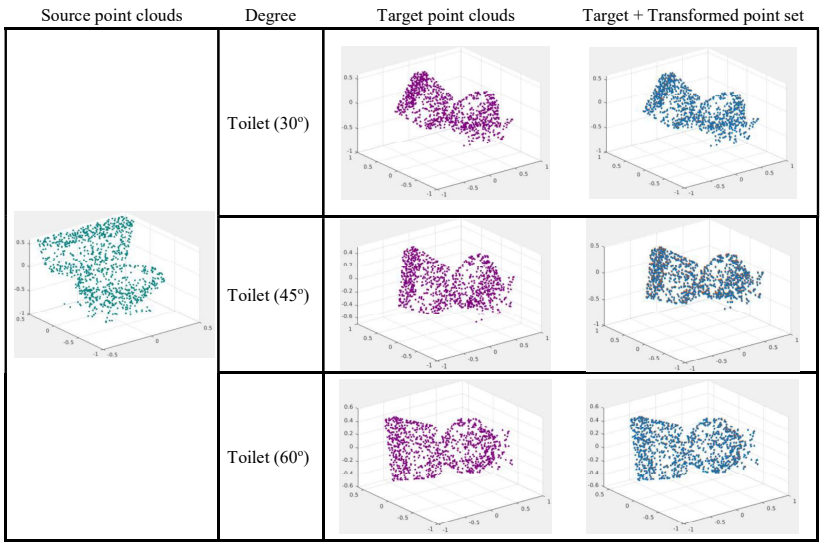

Figure 2.4 *The results of the proposed method on the toilet category. The results are shown for three different degrees of rotation, namely* $30°, 45°, 60°$. *The source point cloud is shown in the first column. Different degrees of target ones are shown in the second column. The third column shows the result after applying the transformation. The output of the proposed method is shown in the last column.*

2.6.1 Dataset

ModelNet [19] is one of the most used synthetic datasets which is provided in two subsets: ModelNet10 and ModelNet40. ModelNet40 contains 40 categories composed of 12311 CAD models while ModelNet10 contains 10 categories including 4899 CAD models in mesh format. The dataset is divided into train and test data, consisting of 3991 and 908 models, respectively. Furthermore, the Chair category of ShapeNet [20] is used to report the accuracy of the proposed method as well as the time efficiency.

In this paper, the test data from ModelNet10 and ShapeNet are considered as source point clouds. Then, to generate the target ones different degrees of rotation and random translation values for (x, y, z) are applied to the source point clouds. Moreover, to evaluate the method on the ShapeNet dataset, the objects are randomly selected from the armchair, folding chair, and swivel chair categorizations.

2.6.2 Experimental results

It is worth mentioning that to evaluate the proposed method, the data can be categorized into three parts, namely source point clouds, target point clouds, and

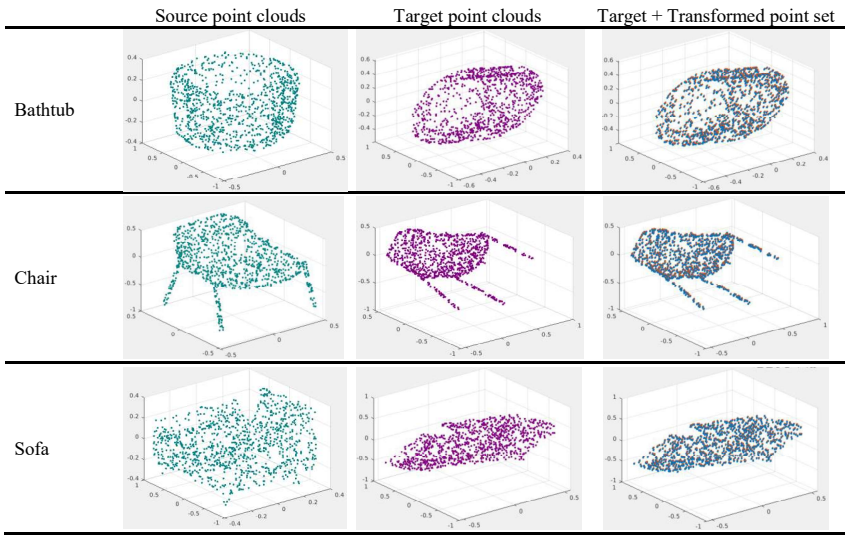

Figure 2.5 *The results of the proposed method. The results are shown for four categories: bathtub, chair, desk, and sofa. The source and target point clouds are shown in the second and third columns. The final results of the proposed registration method are shown in the last column which is called outputs.*

Table 2.1 *The accuracy of the proposed method is provided and compared with the method [18]. The results are shown as distance MSE for 10 categories of ModelNet10.*

Categories	Proposed Method	Method [18] (Distance MSE)	Method [18] (Rotation Error (deg.))
Bathtub	$1.21 \times 10^{-}5$	$4.92 \times 10^{-}5$	1.16
Bed	$2.28 \times 10^{-}5$	$4.41 \times 10^{-}5$	1.15
Chair	$2.39 \times 10^{-}5$	$5.03 \times 10^{-}5$	1.29
Desk	$0.77 \times 10^{-}5$	$5.26 \times 10^{-}5$	1.31
Dresser	$1.82 \times 10^{-}5$	$5.18 \times 10^{-}5$	1.04
Monitor	$1.33 \times 10^{-}5$	$3.83 \times 10^{-}5$	1.03
Night-stand	$1.75 \times 10^{-}5$	$5.04 \times 10^{-}5$	1.08
Sofa	$1.09 \times 10^{-}5$	$3.99 \times 10^{-}5$	1.10
Table	$2.89 \times 10^{-}5$	$6.36 \times 10^{-}5$	1.49
Toilet	$1.89 \times 10^{-}5$	$5.56 \times 10^{-}5$	1.15
Average	$\mathbf{1.74 \times 10^{-5}}$	$\mathbf{4.94 \times 10^{-5}}$	**1.18**

transformed point clouds. As source point clouds, the test data from ModelNet is considered. After applying different rigid transformations (different rotation and translation), the target point clouds are generated. Therefore, to study the proposed method's accuracy, a distance measurement is calculated between the target point clouds and the transformed ones. The Mean Square Error (MSE) is used to calculate

the distance error shown in Equ. 2.5.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \tag{2.5}$$

Furthermore, the results are evaluated in one more measurement called Chamfer distance (C. D.) which is shown in Equ. 2.6.

$$\begin{aligned} CD\,(S,T) = \frac{1}{|S|}\sum_{x\in S}\min_{x\in T}\|x-y\|_2^2 + \\ \frac{1}{|T|}\sum_{y\in T}\min_{x\in S}\|x-y\|_2^2 \end{aligned} \tag{2.6}$$

The performance of the presented method is analyzed by objective and subjective measures which will be discussed in the following.

Subjective measurement For subjective measurement evaluation, the results of the proposed method in different situations are demonstrated. As mentioned before, each target is generated in different degrees of rotation and random translation values for (x, y, z). Figure 2.4 shows an example of the proposed method output on the toilet category. In another evaluation, the final results for some categorizations are presented in Figure 2.5.

Objective measurement To demonstrate the accuracy of the calculated transformation matrix, the proposed method is implemented for all 10 categories. The MSE measurement is run on each category between the target point cloud and the output of the proposed method. The result should be minimized. Therefore, a value of zero corresponds to a registration accuracy of one hundred percent. The proposed method's accuracy is shown in Table 2.1. Furthermore, the results of method [18] are shown in this table to compare the accuracy of each category. In comparison with the other method, it can be concluded that the proposed method could achieve higher accuracy. Furthermore, results for the Chair category in the ShapeNet dataset in comparison with GP-Aligner [21] and Norm-IP [22] are presented in Table 2.2.

Table 2.2 *The accuracy of the proposed method in comparison with the methods [21] and [22]. The results are shown as Chamfer distance (C. D.) for the Chair category. The running- time of the methods is shown in the last column.*

Methods	C. D.	Time
Norm-IP [22]	$21.42 \times 10^{-}4$	4328s
GP-Aligner (Shape-Wise) [21]	$3.10 \times 10^{-}4$	16s
GP-Aligner (GroupWise) [21]	$2.20 \times 10^{-}4$	16s
Proposed Method	$2.159 \times 10^{-}5$	0.0024s

The amount of time an algorithm needs to execute is known as run-time efficiency. Various factors affect the execution speed such as the parallelism of the program, the hardware utilization, and the choice of programming language, to name a few. Regarding the mentioned factors, comparing the time efficiency of the approaches based on time can be unfair. All in all, in this paper, to provide an overview of the proposed method's running-time efficiency, the run-time in comparison with three other methods is presented. To study the time efficiency of the proposed method, the running time of the proposed method, GP-Aligner [21], and Norm-IP [22] are reported in Table 2.2.

It is worth mentioning that no parallelization has been done in the implementation of the proposed method and the presented time in Table 2.2 for all methods is for running the test dataset.

In the available studies beside reporting distance and loss function, the methods' robustness to noise also is reported. In this regard, two various dataset types are discussed in the literature, namely real-world datasets and synthetic ones. Real-world datasets which are commonly captured using LiDAR cameras or scanners suffer from different types of noise. This is not the case with synthetic datasets and it is usually added synthetically to the dataset to challenge proposed approaches [6]. In this paper, a synthetic dataset called ModelNet and ShapeNet is used and the noise challenge is not studied which can be considered as a future study for this paper.

2.7 Discussion

A rigid point cloud registration method based on global feature learning and graph representation is presented in this paper. The proposed method provides several contributions. One is the accurate transformation by using graph matching which is robust to different affine transformations (rotation and translation).

Considering the fact that the point cloud has an unordered structure, the proposed method takes the advantage of PointNet architecture, using max pooling, to overcome this challenge. Furthermore, PointNet architecture is used to extract global features as input for the next steps. Hence, PointNet makes the extracted features robust to different affine transformations. Additionally, the proposed corresponding point determination based on graph representation and graph matching makes the approach robust to different transformations. It is worth mentioning that regarding the start graph representation, the computation time is low in comparison with the complete graph.

2.8 Conclusion

In this paper, an efficient rigid point cloud registration method was proposed. The proposed method had three main steps, namely feature extraction, corresponding point determination based on graph representation, and transformation matrix

calculation. It is shown in the results that the proposed method has high accuracy in rigid registration and that it is robust to different degrees of rotation and various values of translation. In future work, it can be studied how the proposed method can be improved to estimate non-rigid transformations.

Acknowledgment

The authors gratefully acknowledge the data storage service SDS@hd supported by the Ministry of Science, Research and the Arts Baden-Württemberg (MWK) and the German Research Foundation (DFG) through grant INST 35/1314-1 FUGG and INST 35/1503-1 FUGG.

This work was partially funded by Zentrales Innovationsprogramm Mittelstand (ZIM) under grant KK5044704CS0.

Bibliography

[1] G. Elbaz, T. Avraham, and A. Fischer, "3D point cloud registration for localization using a deep neural network auto-encoder," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2017, pp. 4631–4640.

[2] R. Y. Takimoto, M. d. S. G. Tsuzuki, R. Vogelaar, T. de Castro Martins, A. K. Sato, Y. Iwao, T. Gotoh, and S. Kagei, "3D reconstruction and multiple point cloud registration using a low precision RGB-D sensor," *Mechatronics*, vol. 35, pp. 11–22, 2016.

[3] Z. Zhang, J. Sun, Y. Dai, D. Zhou, X. Song, and M. He, "A representation separation perspective to correspondences-free unsupervised 3D point cloud registration," *IEEE Geoscience and Remote Sensing Letters*, 2021.

[4] B. Mahmood and S. Han, "3D registration of indoor point clouds for augmented reality," in *Computing in Civil Engineering 2019: Visualization, Information Modeling, and Simulation*. American Society of Civil Engineers Reston, VA, 2019, pp. 1–8.

[5] S. Ao, Q. Hu, B. Yang, A. Markham, and Y. Guo, "Spinnet: Learning a general surface descriptor for 3D point cloud registration," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, 2021, pp. 11 753–11 762.

[6] S. Monji-Azad, J. Hesser, and N. Löw, "A review of non-rigid transformations and learning-based 3D point cloud registration methods," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 196, pp. 58–72, 2023.

[7] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3D classification and segmentation," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2017, pp. 652–660.

[8] W. Wu, Z. Qi, and L. Fuxin, "Pointconv: Deep convolutional networks on 3D point clouds," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, 2019, pp. 9621–9630.

[9] Y. Wang, and J. Solomon, "Deep closest point: Learning representations for

point cloud registration," in *Proc. of the IEEE/CVF Int. Conf. on Computer Vision*. IEEE, 2019, pp. 3522–3531.

[10] B. Chen, H. Chen, B. Song, and G. Gong, "TIF-Reg: Point cloud registration with transform-invariant features in se (3)," *Sensors*, vol. 21, no. 17, p. 5778, 2021.

[11] C. Choy, W. Dong, and V. Koltun, "Deep global registration," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*. IEEE, 2020, pp. 2511–2520.

[12] B. Wu, J. Ma, G. Chen, and P. An, "Feature interactive representation for point cloud registration," in *Proc. of the IEEE/CVF Int. Conf. on Computer Vision*, 2021, pp. 5530–5539.

[13] T. Min, E. Kim, and I. Shim, "Geometry guided network for point cloud registration," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7270–7277, 2021.

[14] K. Fu, S. Liu, X. Luo, and M. Wang, "Robust point cloud registration framework based on deep graph matching," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, 2021, pp. 8893–8902.

[15] P. J. Besl and N. D. McKay, "Method for registration of 3-D shapes," in *Sensor fusion IV: control paradigms and data structures*, vol. 1611. Spie, 1992, pp. 586–606.

[16] A. Myronenko and X. Song, "Point set registration: Coherent point drift," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 32, no. 12, pp. 2262–2275, 2010.

[17] M. Jaderberg, K. Simonyan, A. Zisserman *et al.*, "Spatial transformer networks," *Advances in neural information processing systems*, vol. 28, 2015.

[18] V. Villena-Martinez, M. Saval-Calvo, J. Azorin-Lopez, A. Fuster-Guillo, and R. B. Fisher, "Local-global based deep registration neural network for rigid alignment," in *Proc. of the Int. Joint Conf. on Neural Networks*, 2021, pp. 1–8.

[19] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3D shapenets: A deep representation for volumetric shapes," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2015, pp. 1912–1920.

[20] A. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su *et al.*, "Shapenet: An information-rich 3D model repository," *arXiv preprint arXiv:1512.03012*, 2015.

[21] L. Wang, N. Zhou, H. Huang, J. Wang, X. Li, and Y. Fang, "GP-Aligner: Unsupervised groupwise nonrigid point set registration based on optimizable group latent descriptor," *IEEE Trans. on Geoscience and Remote Sensing*, vol. 60, pp. 1–13, 2022.

[22] L. G. Sanchez Giraldo, E. Hasanbelliu, M. Rao, and J. C. Principe, "Groupwise point-set registration based on Renyi's second order entropy," *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 6693–6701, 2017.