Yonsei University, Department of Computer Science

Capstone Design, 2023 Fall

Final Report

3D Point Cloud Registration With Graph-based analysis

Team Name: Doves and Ravens

Team Members:

Shen Jing Jun	(2020147594)
Yu Aoze	(2021147568)
Kim Sang Won	(2020147600)
Kim Si Yeol	(2019145121)
Hwang Jun Ho	(2017147577)

Contents

Abstract

1. Introduction

(1) 3D Point Cloud Registration (3D PCR)

(2) Formal Definition of PCR

2. Methods

(1) Wavelet Transform

(2) Third Order Graph (TOG)

(3) Multi-scale Clustered Graph (MCG)

3. Results

(1) Wavelet Transform

(2) Third Order Graph (TOG)

(3) Multi-scale Clustered Graph (MCG)

4. Conclusion and Discussion

(1) Conclusion

(2) Limitations

(3) Further Research Plans

References

Abstract

3D Point Cloud Registration is one of the fundamental technologies used for medical imaging and virtual reality, etc., by aligning different 3d point clouds for the same object as much as possible. Various methods such as deep learning have been attempted for this, but a non-deep learning, algorithm-based method named Maximal Cliques algorithm proposed by Zhang et al. [1], recently attracted attention for its very good performance.

Their method is divided into five stages: Input Correspondences, Graph Construction, Search Maximal Cliques, Hypothesis Generation and Evaluation, and 3D registration. We presented and applied several alternative methods for first two of these stages to observe whether the performance was improved in each case, and we have obtained the results that can give some meaningful conclusions.

1. Introduction

(1) 3D Point Cloud Registration (3D PCR)

In recent years, the emergence of advanced sensing technologies has led to an unprecedented surge in the availability of three-dimensional (3D) point cloud data. Point clouds, representing dense sets of spatial points in a given environment (Fig 1), are acquired through various sensing modalities such as laser radar, stereo vision, and structured light. The wealth of information embedded in these 3D point clouds has propelled their extensive application across domains including robotics, computer vision, medical imaging, and virtual reality.

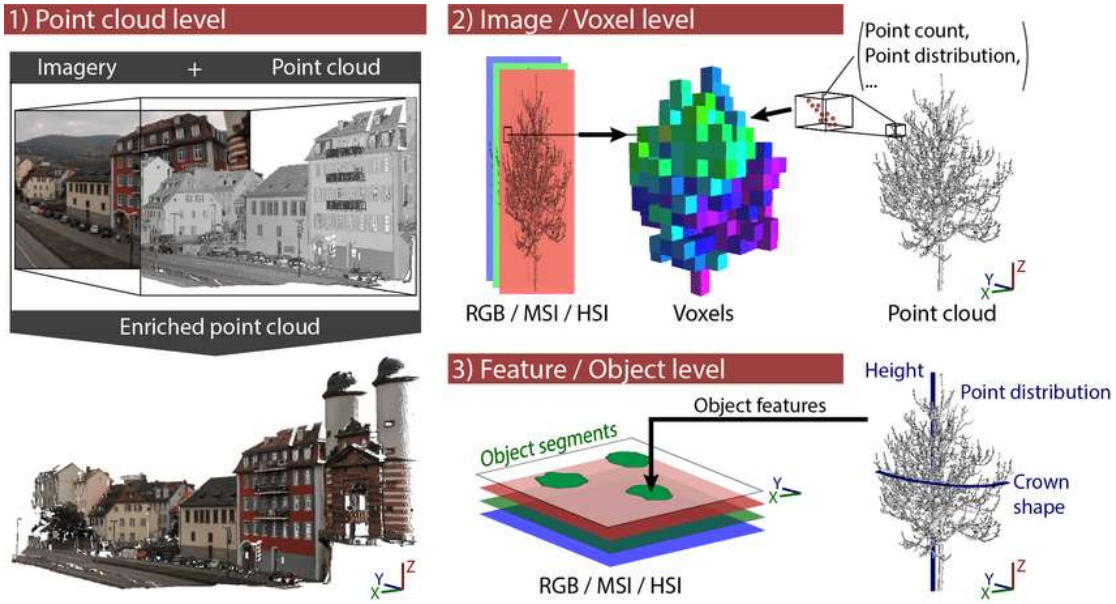
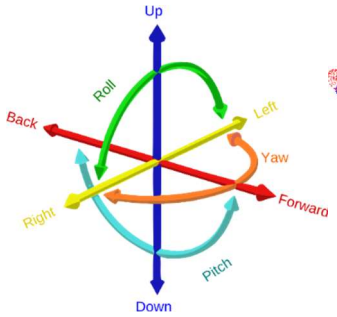


Figure 1. Extraction of point cloud [2]

However, the effective utilization of 3D point cloud data often relies on the seamless integration of multiple scans or viewpoints, necessitating the critical process of point cloud registration or alignment. Point cloud registration (PCR) involves the transformation and alignment of disparate point clouds into a common coordinate system, enabling the creation of a unified and accurate representation of the underlying environment. This process, known as point cloud registration, plays a pivotal role in harnessing the full potential of point cloud data for various applications. The ultimate goal of PCR is to find an appropriate transformation matrix that allows this alignment, that is, a linear transformation matrix with 6 degrees of freedom.



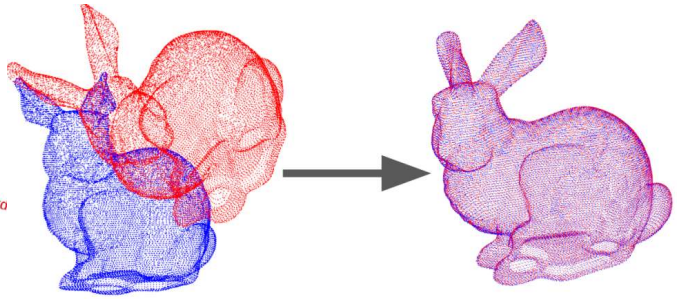


Figure 2. 6-degree-of-freedom [3] **Figure 3. Intuitive example of PCR [4]**

(2) Formal definition of PCR

To express the PCR problem more strictly, two different point cloud sets for the same object $\mathcal{P} = \{p_i \in \mathbb{R}^3 \mid i = 1, \dots, N\}$, $\mathcal{Q} = \{q_i \in \mathbb{R}^3 \mid i = 1, \dots, N\}$ it can be defined as the problem of finding a rigid transformation $\mathrm{T} = \{\mathrm{R}, \mathrm{t}\}$ that aligns $\mathcal{P}$ and $\mathcal{Q}$ [2]. At this time, $\mathrm{R} \in SO(3)$ represents 3D rotation, and $\mathrm{t} \in \mathbb{R}^3$ represents 3D translation.

In other words, the goal of the problem is to minimize the sum of square-Euclidean distances between the corresponding elements of Q by transforming each element of P. Therefore, the optimal $\mathrm{R}^*, \mathrm{t}^*$ is defined as follows.

$$(\mathbf{R}^*, \mathbf{t}^*) = \underset{\mathbf{R},\mathbf{t}}{\operatorname{argmin}} \sum_{(x_i, y_i) \in \mathcal{C}^*} \left\| \mathbf{R} \cdot \mathbf{p}_{x_i} + \mathbf{t} - \mathbf{q}_{y_i} \right\|_2$$

In this case, $\mathcal{C}^* = \{(x_i, y_i) \mid i = 1, \dots, K\}$ is a set representing the correspondence between $\mathcal{P}$ and $\mathcal{Q}$. For example, in the case of Figure 2, $\mathcal{C}^* = \{(i, i) | i = 1, 2, 3, 4\}$. However, since $\mathcal{C}^*$ is information that is not given in real situations, predicting an appropriate $\mathcal{C}^*$ before calculating $\mathrm{R}^*, \mathrm{t}^*$is also an important part of solving the PCR problem.

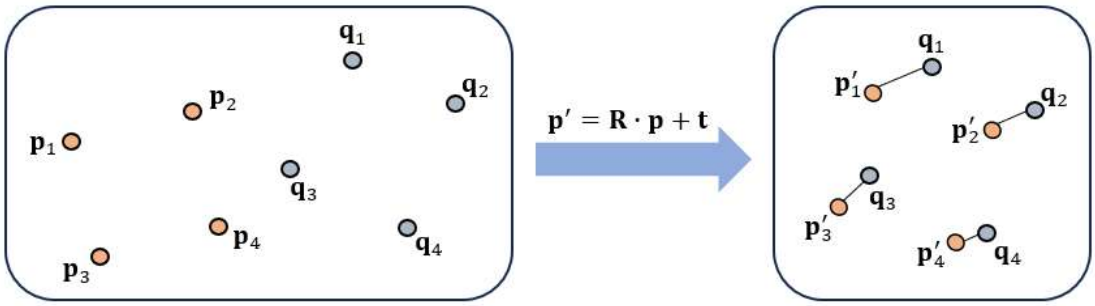


Figure 4. Applying appropriate rotation $\mathbf{R}^*$ and translation $\mathbf{t}^*$ transformations to P to minimize the sum of its distances from $\mathcal{Q}$.

The challenges associated with 3D point cloud registration are diverse, encompassing issues such as sensor noise, occlusions, resolution variations, and dynamic environmental conditions. (Fig.5) Addressing these challenges requires sophisticated algorithms and methods capable of robustly aligning point clouds, providing a foundation for downstream tasks such as object recognition, environment modeling, and motion analysis.

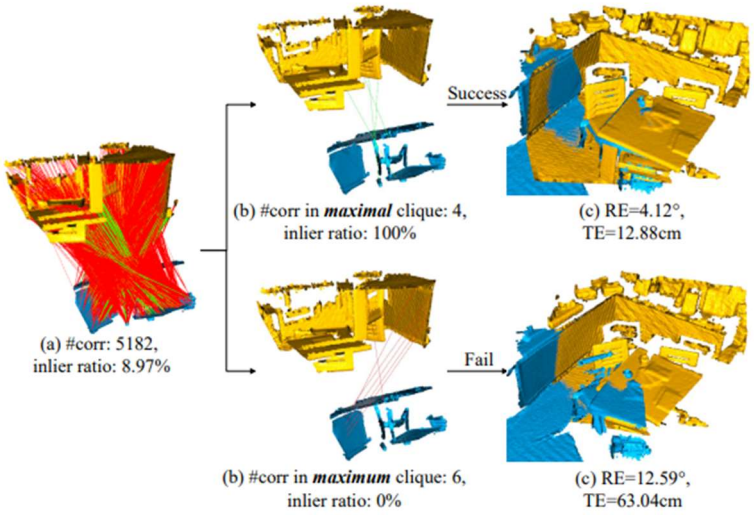


Figure 5. Diverse challenges with 3D PCR [1]

In the paper titled "3D Registration with Maximal Cliques," the authors introduce a novel Point Cloud Registration (PCR) algorithm, termed Maximal Cliques-based Registration (MAC). [1] This algorithm leverages the concept of maximal cliques to significantly enhance the accuracy of point cloud registration, surpassing various state-of-the-art methods and elevating the performance beyond that achieved by deep learning approaches. Our research team has undertaken enhancements to the Input Correspondences and Graph Construction aspects of the PCR algorithm (Fig 6) proposed in the paper. These improvements aim to achieve more accurate 3D registration and reduce the computational time required for graph construction. In the Input Correspondences phase, the team employed the preprocessing

method using Wavelet transform. In the Graph Construction phase, the team utilized two distinct approaches, namely Third Order Graph (TOG) and Multiscale Clustered Graph (MCG), to enhance the algorithm's performance.

In our research, we assess the effectiveness of the proposed accuracy-enhancing methods by comparing the original and improved metrics, including Rotation Error (RE), Translation Error (TE), and Registration Recall (RR). These metrics serve as benchmarks to evaluate the precision of the proposed enhancements.

Furthermore, to validate the impact on computational efficiency, we conduct a comparative analysis of the time consumption associated with key components such as graph construction, clique searching, clique selection, and pose estimation. This evaluation aims to determine whether the introduced methods contribute to an enhancement in processing speed. By juxtaposing the performance metrics and computational timings before and after the proposed improvements, we aim to provide a comprehensive assessment of the efficacy of our approach in achieving both heightened accuracy and accelerated processing in the context of point cloud registration.

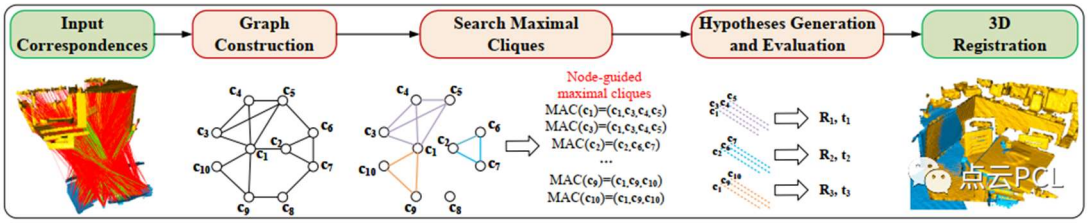


Figure 6. Five stages of the pipeline suggested in MAC research. [1]

2. Methods

(1) Preprocessing Based on Wavelet Transform

Wavelet Transform, or wavelet series is a representation of a square-integrable function by a certain orthonormal series generated based on “Wavelet”. Wavelet series is defined as below,

$$f(x) = \sum_{j,k=-\infty}^{\infty} c_{jk}\psi_{jk}(x)$$

where c_{jk} is wavelet coefficient and ψ_{jk} is an orthonormal wavelet if in can be used to define complete orthonormal system. [5] A wavelet is wave-like oscillating function with an amplitude that begins with zero, increases or decreases, and then returns to zero one or more time. [6] One example of wavelet function is shown below.

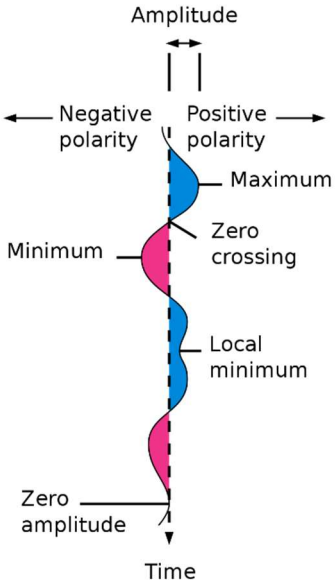


Figure 7. Seismic wavelet

Wavelet transform is widely used for processing signals. When it comes to visual images, a result of wavelet transform can show a frequency intensity based on wavelet basis. This can be compared to Fourier transform which is well-known scheme for signal processing. Wavelet transform has its own strength in locality. Unlike Fourier transform, consists of sine and cosine basis, wavelet basis offers an analysis of time step or position. [7] Our idea is to apply wavelet transform to original point clouds before constructing correspondences. We expected that this procedure will make point cloud data sparser, and feature extracted which means to lessen the number of the point that compose the whole point cloud and choose more important points for representing its feature which can contribute to build correspondence more. These can lead overall registration more correct and faster.

A method we proposed is carried out in the following steps.

1. Finding wavelet coefficients
2. Calculating threshold value of coefficients
3. Discarding points whose wavelet coefficient is less than the threshold
4. Reconstructing point cloud which has extracted feature by third step

The overall procedure is implemented with MATLAB. For each dimension of the original point cloud (x, y, z), I found wavelet coefficient by “Wavedec” function of MATLAB with wavelet basis of “Haar”. [8] Then, we made a histogram of coefficients and calculated threshold value for input point cloud by Otsu’s method.[9] The algorithm of Otsu’s method returns a single intensity threshold that separates pixels into two classes.[10] The method typically used for clustering pixels on image, but we applied this on classifying points. Points whose absolute value of wavelet coefficient is lower than the threshold are discarded and remaining points are high-frequency element which means it shows edges or corners; features of the point cloud.

(2) Third Order Graph

As a part of Graph construction stage, previous work [1] proposed a method called Second Order Graph (SOG) for eliminating unnecessary edges from First Order Graph (FOG). SOG is sparser than FOG, and therefore beneficial in making the search of the cliques more rapid.

We propose Third Order Graph (TOG), which is similar to SOG, but amplifies the weight of each edge belonging to a clique more than SOG. The weight matrix of SOG and TOG are calculated as:

$$\mathbf{W}_{SOG} = \mathbf{W}_{FOG} \odot \mathbf{W}_{FOG}^2$$

$$\mathbf{W}_{TOG} = \mathbf{W}_{FOG} \odot (\mathbf{W}_{FOG}^2 + \mathbf{W}_{FOG}^3)$$

where $\odot$ is Hadamard product. Here, we can see that $\mathbf{W}_{TOG}$ is a simple modification of $\mathbf{W}_{SOG}$, adding a term cubed by the weight matrix of FOG.

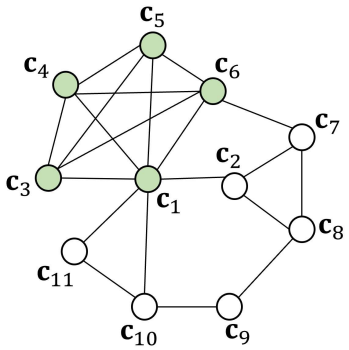


Figure 8. An example of FOG.

To analyze the effectiveness of TOG, we can interpret $\mathbf{W}_{FOG}^k(i,j)$ as the sum of weights of all paths from i to j with step k. If i and j belong to the same clique, the number of paths from i to j increases exponentially as k increases. For example, as shown in Figure 1, the green nodes form a clique of size 5. Regarding the paths from $\mathbf{c}_1$ to $\mathbf{c}_3$, the number of step 2 paths is 3: $(\mathbf{c}_1, \mathbf{c}_4, \mathbf{c}_3)$, $(\mathbf{c}_1, \mathbf{c}_5, \mathbf{c}_3)$, $(\mathbf{c}_1, \mathbf{c}_6, \mathbf{c}_3)$, while the number of step 3 paths is 6: $(\mathbf{c}_1, \mathbf{c}_4, \mathbf{c}_5, \mathbf{c}_3)$, $(\mathbf{c}_1, \mathbf{c}_4, \mathbf{c}_6, \mathbf{c}_3)$, $(\mathbf{c}_1, \mathbf{c}_5, \mathbf{c}_6, \mathbf{c}_3)$, $(\mathbf{c}_1, \mathbf{c}_5, \mathbf{c}_4, \mathbf{c}_3)$, $(\mathbf{c}_1, \mathbf{c}_6, \mathbf{c}_4, \mathbf{c}_3)$, $(\mathbf{c}_1, \mathbf{c}_6, \mathbf{c}_5, \mathbf{c}_3)$.

More specifically, for a clique of size s, the number of step k paths between any two nodes is $\frac{(s-2)!}{(s-k-1)!} = \theta(s^{k-1})$. The larger number of paths means that there is a high possibility of increasing the weight by that amount. Therefore, due to the $\mathbf{W}_{FOG}^3$ term of $\mathbf{W}_{TOG}$, we can expect $\mathbf{W}_{TOG}(i,j)$ to be much larger than $\mathbf{W}_{SOG}(i,j)$.

According to our experiments, the performance of SOG was slightly better than TOG, but in some cases, TOG performed better. To improve the overall performance, we attempted to combine the strengths of SOG and TOG. As shown in Figure 2, we searched the maximal

cliques for both SOG and TOG, and then used both sets of cliques for evaluation.

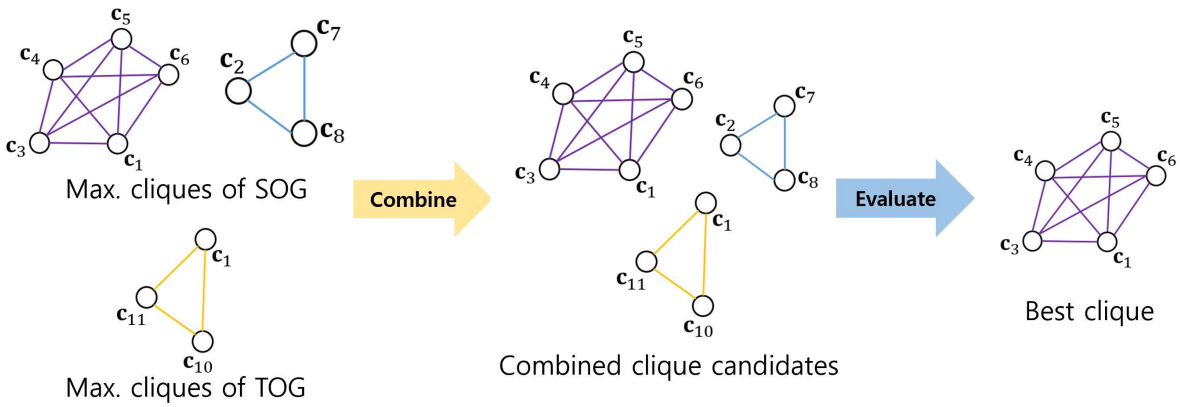


Figure 9. Procedure of combined method of SOG and TOG.

(3) Multi-scale Clustered Graph

We propose another method for Graph construction stage, called Multi-scale Clustered Graph(MCG). MCG utilizes a clustering method to remove unnecessary edges in FOG. Although there are various clustering methods, we used the wavelet-based clustering method presented in [11]. The connectivity between two nodes is calculated by "graph wavelets", which provides a compact representation of the local environments of those nodes in the graph. If the graph wavelets of the two nodes are similar, they are considered to be close in Fourier domain. Since graph wavelet has a scale parameter, we can adjust it to find global or local clusters. As shown in Figure 3, the larger the scale, the more global clusters can be found, and the smaller the scale, the more local clusters can be found.

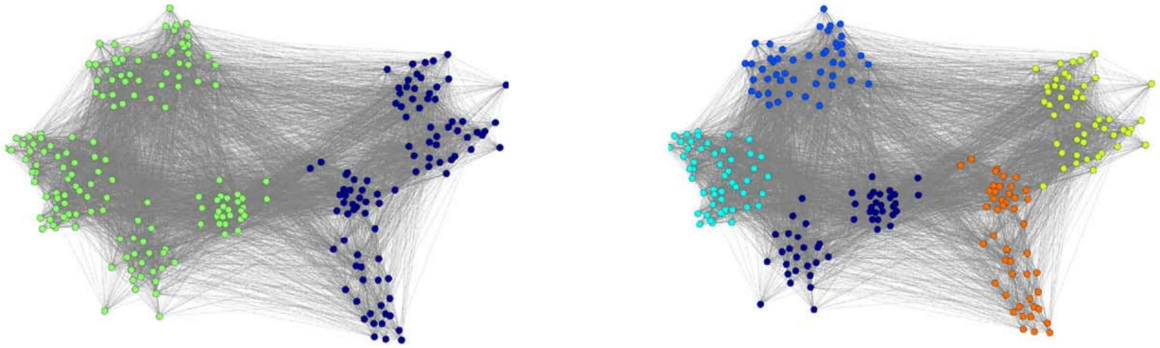


Figure 10. Global vs Local clusters in multiscale clustering.

Now, we will introduce the specific method of computing MCGs and its implementation. For a given FOG with weight matrix $\mathbf{W}$, the normalized Laplacian matrix $\mathcal{L}$ is defined as:

$$\mathcal{L} = \mathbf{I}_N - \mathbf{D}^{-\frac{1}{2}}\mathbf{W}\mathbf{D}^{-\frac{1}{2}}$$

where $\mathbf{D}$ is a diagonal matrix with $\mathbf{D}_{ii} = \sum_{j \neq i} \mathbf{W}_{ij}$.

Since $\boldsymbol{\mathcal{L}}$ is symmetric, we can apply eigenvalue decomposition (EVD) to get its real-valued eigenvalues $(\lambda_l)_{l=1\ldots N}$ and eigenvectors $\boldsymbol{\chi} = (\boldsymbol{\chi}_1 | \boldsymbol{\chi}_2 | \ldots | \boldsymbol{\chi}_N)$.

Let g be a band-pass wavelet filter kernel function defined as:

$$g(x; \alpha, \beta, x_1, x_2) = \begin{cases} x_1^{-\alpha} x^{\alpha} & (x < x_1) \\ p(x) = ax^3 + bx^2 + cx + d & (x_1 \leq x \leq x_2) \\ x_2^{\beta} x^{-\beta} & (x > x_2) \end{cases}$$

Here, α, β, x_1, x_2 are the parameters to choose for the kernel design. For that, we set $\alpha = 2, \beta = 1/\log_{10}\left(\frac{\lambda_3}{\lambda_2}\right), x_1 = 1, x_2 = 1/\lambda_2$ following [11]. Since g must be differentiable, the four conditions must be met: $g(x_1) = p(x_1), g(x_2) = p(x_2), g'(x_1) = p'(x_1), g'(x_2) = p'(x_2)$. Then the coefficients a, b, c, d of the cubic polynomial $p(x)$ is uniquely determined by solving the system of equations.

The graph wavelet basis at scale s is then calculated as:

$$\boldsymbol{\Psi}_s = (\boldsymbol{\psi}_{s,1} | \boldsymbol{\psi}_{s,2} | \ldots | \boldsymbol{\psi}_{s,N}) = \boldsymbol{\chi} \boldsymbol{G}_s \boldsymbol{\chi}^{\mathrm{T}}$$

where $\boldsymbol{G}_s = \mathrm{diag}(g(s\lambda_1), \ldots, g(s\lambda_N))$.

The wavelet distance between two nodes a and b at scale s is:

$$\boldsymbol{D}_s(a, b) = 1 - \frac{\boldsymbol{\psi}_{s,a}^{\mathrm{T}} \boldsymbol{\psi}_{s,b}}{\|\boldsymbol{\psi}_{s,a}\|_2 \|\boldsymbol{\psi}_{s,b}\|_2}$$

Finally, we can generate MCG at scale s by assigning the weight matrix as follows:

$$\mathbf{W}_{MC\ s}(a, b) = \mathbf{W}(a, b)[\boldsymbol{D}_s(a, b) < 1]$$

Now, we choose a log-spaced sample of scales $\mathcal{S} = \{s_1 = s_{min}, s_2, \ldots, s_M = s_{max}\}$ to generate multi-MCGs. Following [11], we set the minimum scale $s_{min} = 1/\lambda_2$, the maximum scale $s_{max} = 1/\lambda_2^2$, the number of scales $M = 10 \log_2 N$. In the experiments, we just used the three scales $s_1, s_{M/2}, s_M$ (low scale, mid scale, and high scale respectively) for generating MCGs to reduce the number of computations. Then for all MCGs, the maximal cliques are computed and used for evaluation.

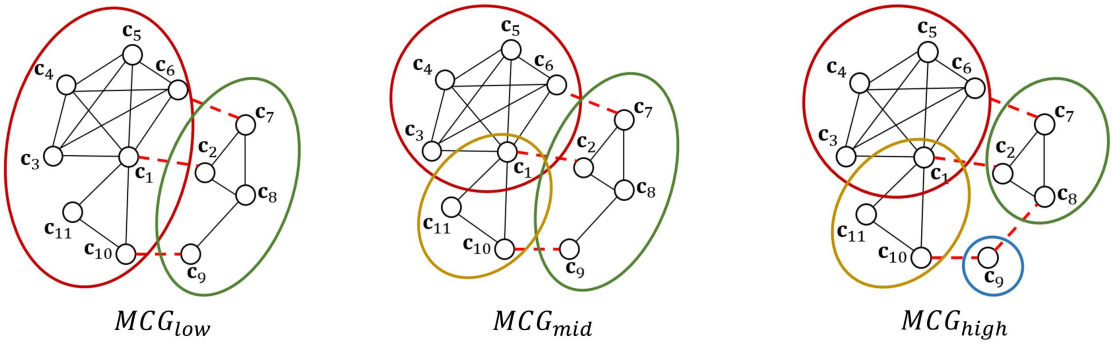


Figure 11. An example of MCGs of low-scale, mid-scale, high-scale respectively.

Due to the numerous steps involved in MCG computation, it requires a large amount of processing, especially for operations like matrix multiplication and eigenvalue decomposition. For basic linear algebra operations including matrix multiplication, we used the functions of C++ Eigen library [12] which supports parallel processing. Eigen also provides EVD, however, it is CPU-based and thus very slow. To boost the speed, we decided to utilize GPUs for EVD computation, and implemented our own CUDA Kernel which performs EVD on a symmetric matrix using cuSolver [13] library. For $N = 5000$, Eigen-based EVD took more than 5 minutes, whereas CUDA-based EVD only took 3 seconds (with NVIDIA RTX 2080 TI GPU). This significantly reduced the time required for our experiments.

3. Results

(1) Preprocessing Based on Wavelet Transform

We conducted comparative experiment between preprocessed point cloud and original one. We used 3Dmatch and 3dLoMatch dataset for the experiment. FCGF descriptor is used for correspondence building and SOG (Second Order Graph) is used for constructing graph of correspondences. it shows less accuracy than cases of original point cloud and better speed in overall registration task as tables show below. Unlike image processing, point cloud data only contains positional information. [14] Therefore, applying the method used in the image area directly to such data isn't that appropriate with respect to accuracy. However, there was extreme benefits in time which is approximately 48 times better in 3dMatch and 40 times better in 3dLoMatch.

	RR(%)	RE(°)	TE(cm)
ORIGINAL	**93.72**	**1.887**	**6.013**
TRANSFORMED	74.00	2.712	7.328

Table 1. Result on 3Dmatch (Accuracy)

	RR(%)	RE(°)	TE(CM)
ORIGINAL	**60.08**	**3.497**	**9.721**
TRANSFORMED	35.04	4.341	10.727

Table 2. Result on 3dLoMatch (Accuracy)

	GRAPH CONSTRUCTION	CLIQUE SEARCHING	CLIQUE SELECTION	POSE ESTIMATION	TOTAL
ORIGINAL	1590.68	357.60	9.20	185.30	2142.78
TRANSFORMED	**33.64**	**9.66**	**0.24**	**0.81**	**44.35**

Table 3. Result on 3DMatch (Time, ms)

	GRAPH CONSTRUCTION	CLIQUE SEARCHING	CLIQUE SELECTION	POSE ESTIMATION	TOTAL
ORIGINAL	1754.06	732.89	19.08	421.23	2927.26
TRANSFORMED	**56.76**	**15.43**	**0.42**	**1.39**	**74.00**

Table 4. Result on 3DLoMatch (Time, ms)

(2) Third Order Graph

We conducted comparative experiments on SOG, TOG, and SOG+TOG (the combined method of SOG and TOG). For the datasets, we used 3DMatch [15], 3DLoMatch [16] and KITTI [17] datasets, and for the initial correspondences, FCGF [18] descriptor was employed in all cases. In 3DMatch and 3DLoMatch datasets, the performance of SOG was generally better, but in KITTI dataset, SOG+TOG showed better performance.

	RR(%)	RE(°)	TE(cm)
SOG	**93.72**	**1.887**	**6.013**
TOG	92.98	1.897	6.074
SOG+TOG	**93.72**	1.905	6.062

Table 5. Result on 3DMatch dataset.

	RR(%)	RE(°)	TE(cm)
SOG	**60.08**	**3.497**	**9.721**
TOG	59.80	3.499	9.726
SOG+TOG	59.74	3.502	9.750

Table 6. Result on 3DLoMatch dataset.

	RR(%)	RE(°)	TE(cm)
SOG	**97.12**	0.366	7.820
TOG	**97.12**	0.371	7.908
SOG+TOG	**97.12**	**0.362**	**7.787**

Table 7. Result on KITTI dataset.

(3) Multi-scale Clustered Graph

We conducted comparative experiments on SOG and MCG. For the datasets, we used 3DMatch, 3DLoMatch and KITTI datasets, and for the initial correspondences, FCGF descriptor was employed in all cases. Overall, the RR was slightly higher for SOG, but there was no dominance in TE or RE for either side.

	RR(%)	RE(°)	TE(cm)
SOG	**93.72**	1.887	**6.013**
MCG	93.65	**1.882**	6.046

Table 8. Result on 3DMatch dataset.

	RR(%)	RE(°)	TE(cm)
SOG	**60.08**	**3.497**	9.721
MCG	59.91	3.504	**9.698**

Table 9. Result on 3DLoMatch dataset.

	RR(%)	RE(°)	TE(cm)
SOG	**97.12**	0.366	**7.820**
MCG	**97.12**	**0.364**	8.013

Table 10. Result on KITTI dataset.

4. Conclusion and Discussions

(1) Conclusion

Based on the research performed by Zhang et al [1], we conducted a study to improve the time and accuracy of 3D Point Cloud Registration. Among the pipelines presented in the paper, improvement measures for the first two steps were devised and applied, and the results as

mentioned in result section were obtained.

By experimenting with 3DMatch and 3DLoMatch dataset with a model using Wavelet transform, which can be seen as a kind of data preprocessing, we were able to achieve a definite improvement in the calculation speed (a decrease of approximately 48, 40 times the computation time, respectively) although the accuracy (Registration Recall, RR) was slightly lower (a decrease of approximately 19 and 25 percent, respectively). This can lead to improvements in areas of 3D vision tasks that require fast or real-time calculation, for example, 3D map calculation in autonomous driving system.

The experimental results of applying the Third Order Graph (TOG) method, which improved the process of forming the correspondence graph, to KITTI data set confirmed that TOG method fits well on some data instances which were not well solved when only Second Order Graph (SOG). Although the average performance of the TOG method itself was worse than that of the SOG method, it was a glimpse into its potential. Therefore, we also applied the method of merging maximal cliques obtained from SOG and TOG together to the evaluation, and as a result, it was seen that the values of RE and TE, which are indicators of accuracy, were improved compared to using only SOG.

In the process of graph construction, experiments that replaced SOG with MCG, a graph of a completely different method, also showed better REs for KITTI dataset, resulting in not that great but somewhat meaningful results. In the case of the following two methods with better accuracy, it is thought that accuracy will be meaningful in areas where accuracy is important, such as the field of medical vision dealing with a life.

(2) Limitations

Although it was mentioned in conclusion that the time was reduced a lot when Wavelet transform was applied, in fact, considering the time spending to perform Wavelet transform, it can be said that it is a clear limitation of our study that the immediate time gain will not be significant. However, what this methodology suggests and contributes to this field and study is that if these kind of “important points” used in the model, it can bring great time gains without significantly degrading performance. Therefore, in any other way, devising a way to sort out important points in a reasonable time could bring about time benefits.

In the case of MCG, it was not possible to apply various scales due to the time limitations, and experiments were conducted on only three scales. The advantage of MCG is that it can be analyzed for various scales, but it can be said that this advantage was not properly utilized in our research.

(3) Further Research Plans

In the case of method for Wavelet transform, as mentioned in the limitation section, finding a way to sort out the important points in a reasonable time may be considered a priority. Or alternatively, we will get good results if we come up with a way to design a descriptor that can have better input initial correspondence in the first stage of pipelines of the model. This is

because, like all other machine learning methods, the initial value will severely affect the final result.

For the Graph Construction stage of original study [1], it could be possible to improve the accuracy of the model by just enhancing the quality of FOG construction, since all SOG, TOG, and MCG are based on FOG. Our trials to improve the model by replacing this step with adding/altering some ideas (like TOG, MCG) result in slightly better results, but not that meaningful. Therefore, we thought that if improving FOG, which is fundamentally the starting point of these methods, can have a greater effect.

In the case of MCG, as mentioned above, the wavelet distance is determined according to the scale of the applied wavelet function, and the scale of locality reflected can be changed accordingly. This method has the advantage of being able to analyze various scales in that it will be able to reflect from small regional characteristics to wide regional characteristics. Therefore, it can be thought that applying additional multiple scale diversification will lead to better results.

It is not a pipeline stage that we have touched in this study, but we think it is possible to apply new ideas to the process of searching for Maximum Cliques. It is thought that utilizing Maximum Weight Cliques (MWC) algorithm, which reflects edge weights instead of clique size in simple Maximal Cliques algorithm, can effectively leverage the TOG's capability to amplifying edge weights.

References

[1] Xiyu Zhang, Jiaqi Yang, Shikun Zhang, Yanning Zhang, “3D Registration With Maximal Cliques”, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023

[2] Pedram Ghamisi et al., “Multisource and Multitemporal Data Fusion in Remote Sensing”, IEEE, 2018

[3] Wikipedia contributors. (2023, January 25). Six degrees of freedom. In Wikipedia. https://en.wikipedia.org/wiki/Six_degrees_of_freedom

[4] T. Zodage, “Point Cloud Registration as a Classification Problem,” M.S. thesis, Robotics Institute, Carnegie Mellon Univ., Pittsburgh, PA, USA, 2021

[5] Dunford, N.;Schwartz, J.T., “Linear operators, Parts I and II”, Wiley-Interscience, 1958

[6] Meyer, Yves, “Wavelets and Operators”, Cambridge, UK: Cambridge University Press, ISBN 0-521-42000-8, 1992

[7] Talebi, S., “Fourier vs. Wavelet Transform: What’s the Difference?”, Built In., 2022, https://builtin.com/data-science/wavelet-transform

[8] “Multilevel 1-D Discrete Wavelet Transform - MATLAB Wavedec,” n.d. https://mathworks.com/help/wavelet/ref/wavedec.html.

[9] M. Sezgin & B. Sankur, "Survey over image thresholding techniques and quantitative performance evaluation". *Journal of Electronic Imaging*., 2004

[10] Nobuyuki Otsu (1979). "A threshold selection method from gray-level histograms". IEEE Transactions on Systems, Man, and Cybernetics., 1979

[11] N. Tremblay and P. Borgnat, "Graph Wavelets for Multiscale Community Mining," in IEEE Transactions on Signal Processing, vol. 62, no. 20, pp. 5227-5239, Oct.15, 2014, doi: 10.1109/TSP.2014.2345355.

[12] Eigen, https://eigen.tuxfamily.org (accessed Dec. 12, 2023).

[13] cuSOLVER, https://docs.nvidia.com/cuda/cusolver (accessed Dec. 12, 2023).

[14] Levoy, M. and Whitted, T., *"The use of points as a display primitive"*.. Technical Report 85-022, Computer Science Department, University of North Carolina at Chapel Hill, January, 1985

[15] A. Zeng, S. Song, M. Nießner, M. Fisher, J. Xiao and T. Funkhouser, “3DMatch: Learning Local Geometric Descriptors from RGB-D Reconstructions”, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017

[16] S. Huang, Z. Gojcic, M. Usvyatsov, A. Wieser and K. Schindler, "PREDATOR: Registration of 3D Point Clouds with Low Overlap," 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 2021

[17] A. Geiger, P. Lenz and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Providence, RI, USA, 2012

[18] C. Choy, J. Park and V. Koltun, "Fully Convolutional Geometric Features," IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea (South), 2019