The Impact of the Number of k-Means Clusters on 3D Point Cloud Registration

Peter Ankomah
Manchester Metropolitan University
Department of Computing And Maths
All Saints Building
United Kingdom, Manchester
p.ankomah@mmu.ac.uk

Peter Vangorp
Utrecht University
Department of Information and Computing Sciences
Utrecht, The Netherlands
p.vangorp@uu.nl

ABSTRACT

Point cloud registration plays a crucial role in many applications, from robotics and autonomous navigation to medical imaging and 3D scene reconstruction. While the Iterative Closest Point (ICP) algorithm is a well-known shape registration choice, its efficiency and accuracy can be affected by the vast search space for point correspondences. *k*-means clustering emerges as a promising solution for partitioning the search space into smaller clusters to reduce the computational complexity and increase the performance of the matching. However, the number and size of these clusters and how they affect the registration remains a critical and yet not fully explored factor. This paper delves into the relationship between the number of *k*-means clusters and point cloud registration accuracy. To determine the effect of the number of *k*-means clusters on registration accuracy and efficiency and to understand any emerging pattern, *k*-meansICP is developed to use the *k*-means algorithm to cluster the correspondence search space. Two sets of 3D molecular shapes with differing complexities are matched using initial rotation angles 15, 30, and 60 degrees with 2 to 10 *k*-means clusters. The results are then compared with the original ICP algorithm.

Keywords

3D Point Cloud, 3D Shape Registration. Iterative Closest Point, *k*-Means Clustering

1 INTRODUCTION

Point cloud registration is one of the most important yet challenging areas of computer vision [Pot06, Goj20, MA23]. This field is related to tasks in several domains such as molecular biology for characterising biological interactions [Ovr23], medical image processing [Wu18] for medical image visualisations, robotics [Lyu24, Kri15] for 3D object recognition through reconstruction, and human pose estimation and tracking [Che11], and for optimising cutting tool positioning in the context of Computer Numerical Control (CNC) machining [Bo19].

The iterative closest point algorithm (ICP) [Bes92] is the most popular point cloud registration algorithm [Lyu24, Li22] because of its simplicity and modular nature, allowing enhancements to different aspects of the registration process as required. ICP aims to align two point clouds by finding a transformation that minimises the distance between the point clouds. The algorithm repeatedly alternates between finding the closest point in the target point cloud for each point in the source point cloud and computing a transformation that minimises the distance between the corresponding points [Zha22].

1.1 Limitations of existing solutions

ICP requires a good initial transformation [Lyu24] to converge to a good solution. The algorithm can perform poorly with a large initial transformation resulting in a less optimal convergence [Wu18]. The presence of outliers (non-uniform or non-conforming points), noise produced from point cloud acquisition devices [Du15], and partial overlaps, such as registering subsets of a point cloud, can also decrease the efficiency of the registration process driven by false matches [Pom13]. The nature of ICP as an optimisation algorithm is to find approximate solutions to the registration problem by converging to a locally optimal solution. This makes the quality of the registration results heavily dependent on the initial transformation. An ICP registration process with less optimal initial transformation can lead to incorrect correspondence, and smaller transformation convergence regions making the registration process unstable [Sal24]. In some instances, ICP may not converge at all.

1.2 Outline of this work

This paper evaluates how the number of clusters in a k-means clustered point cloud, and indirectly the number of points in each cluster, affects the registration accuracy. An experiment is set up using two 3D molecular structures from the protein data bank [Ber00], 2HAX and 2JKF. These structures represent different forms of the common cold virus's protein molecule. The experiment registers each structure to a copy of itself by partitioning the search space from 2 to 10 clusters, each time setting an initial rotation of 15, 30, or 60 degrees along one axis. Partitioned clusters reduce the point correspondence search space. This reduction then decreases the computational complexity of the registration process.

Section 3 details an implementation of ICP [Bes92] using k-means clustering to reduce the search space. Section 4 details the experimental setup and design as well as the results, analysis, and implications. Section 5 discusses the findings from the experiment and the performance of the algorithms. Section 6 presents the conclusion of this paper with the reiteration of the problems with ICP [Bes92], how a k-means based implementation improves the matching, further limitations of this approach, and our ongoing and future work.

2 RELATED WORK

2.1 The ICP Algorithm

The original ICP algorithm [Bes92] repeatedly computes the transformation that aligns two point clouds. The algorithm does a correspondence search by finding the closest point in the target point cloud for each point in the source point cloud. A transformation computed from these correspondences is then applied to the position of the source point cloud to bring it in alignment with the target point cloud. The algorithms do the next correspondence search from this transformed source point cloud until the convergence criteria are reached, which is a set number of iterations or a set mean squared error threshold.

2.2 Improvements to the ICP Algorithm

Hundreds of variations of ICP have been developed demonstrating the need for improved versions of the algorithm to handle the vast and diverse datasets [Pom13, Rus01]. The modular nature of ICP allows improvements to be made at different stages including the closest point search, more elaborate distance metrics than the Euclidean distance between points, and weighting of matching point pairs as a way of rejecting less likely matches [Wan17]. One recent approach to using ICP to match protein structures focused on enhancing the registration quality using metadata knowledge of the point cloud [Ank20]. This included extracting rotation-invariant features such as the k nearest neighbours of each point, and other available point labelling information to aid in the correspondence search process.

Algorithm 1 Iterative Closest Point algorithm

```
1:  function ITERATIVE CLOSEST POINT(P, X)
2:      P_0 ← P
3:      for iteration i := 0 to i_max do
4:          closest points Y_i
                ← CLOSEST POINT SEARCH(P_i, X)
5:          transformation M_i, MSE d_i
                ← REGISTRATION(P_0, Y_i)
6:          P_{i+1} ← TRANSFORM(M_i, P_0)
7:          if change in MSE d_{i-1} − d_i < threshold
            then
8:              terminate the for loop
9:          end if
10:     end for
11:     return P_{i+1}, M_i, d_i
12: end function
```

2.3 Use of k-Means Clustering

The k-means clustering algorithm [Kri99] is an iterative optimisation algorithm to cluster any point cloud into a chosen number of k clusters. The algorithm is domain-independent making it an ideal choice for use in specific tasks in pattern recognition and machine learning.

The algorithm accepts the desired number k of partitions and then clusters the data points into k clusters, assigning each point to the closest cluster. This is done by computing the squared Euclidean distance between each data point and the centroid of each cluster, and then assigning each point to the cluster with the closest centroid [Iko23]. There are several variants of the k-means clustering algorithm based on modifications to characteristics such as the distance measure, ways to reduce the number of distance calculations, and the reduction of the dimension of the data set [Iko23]

PF-ICP [Sal24] was proposed as an improvement to the traditional ICP [Bes92] algorithm. PF-ICP uses k-means clustering to help correctly estimate the transformation during registration. The research utilises the $k-means$ clustering method to guide the initialisation of the transformation. The $k-means$ algorithm was employed to segment the data point cloud into clusters, and the initial transformation was obtained by doing a correspondence match of the centroids of the generated clusters. This process was found to reduce the computational complexity and increase registration speed. The transformations computed from matching the centroids of the clusters were then used as the initial transformation for the point cloud registration. In PF-ICP [Sal24], $k-means$ is primarily used for clustering and estimating the initial transformation, whilst $k-means$ partitions the search space into clusters to reduce the

computational complexity of the registration process by reducing the correspondence search space.

3 THE k-MEANSICP ALGORITHM

Our algorithm is based on the original ICP algorithm [Bes92] (Algorithm 1). The ICP algorithm finds the correspondence for each point by computing the distance from each point in the model point cloud to each point in the data point cloud. This process is computationally expensive. Several measures have been proposed to decrease the computational cost of the distance calculation such as by using kd-tree [Bes92], and partitioning the search space using metadata of the points [Ank20]. [Kan02] proposed an efficient k-means clustering algorithm that leverages Voronoi diagrams to significantly reduce time complexity, resulting in substantial accelerations compared to standard k-means clustering methods.

Our algorithm improves computational cost by partitioning the correspondence search space using the k-means clustering algorithm [Kri99]. This means that each point does two sets of distance calculations: The first is to pick the closest cluster by finding the closest centroid to the point. The second is to find the distance from the point to each point belonging to the chosen cluster. This implies every point does point-to-cluster and then point-to-point distance calculations. The number of clusters (k) is a user-configurable constant that can be increased or decreased based on the performance of the registration. Although there are two sets of distance calculations involved in our algorithm, the computational complexity is reduced.

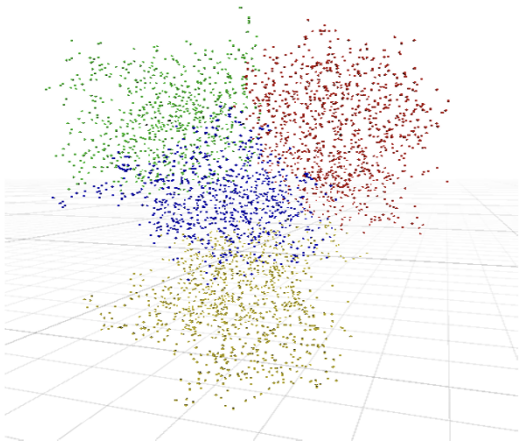

Figure 1: 3D point cloud clustered into 4 colour coded regions

Assume that the point cloud P_i is clustered into k clusters with centroids X_c, where $c \in 1 \ldots k$. For each point $p \in P_i$ the algorithm first finds point p's closest cluster C based on the distances between p and the cluster centroids. The algorithm then finds the closest point to p among the points $x \in C$ by performing a naive search comparing the distance d of every point x and finding the minimum distance d_{min}. The closest point y is appended to the list of closest points Y_i as the correspondence for p.

Algorithm 2 k-meansICP closest point search

```
1:  function CLOSEST POINT SEARCH(P_i, X)
        ▷ X is clustered into k clusters with centroids X_c
2:      closest points Y_i ← empty list
3:      for all points p in P_i do
4:          cluster C ← CLOSEST CLUSTER(p, X_c)
5:          closest point y ← empty
6:          closest point distance d_min ← ∞
7:          for all points x in C do
8:              distance d ← DISTANCE(p, x)
9:              if d < d_min then
10:                 closest point y ← x
11:                 closest point distance d_min ← d
12:             end if
13:         end for
14:         append closest point y to closest points Y_i
15:     end for
16:     return Y_i
17: end function
```

3.1 Computational complexity

k-meansICP leverages the clustering information to reduce the search space. It achieves this by first finding the nearest cluster for each data point (p) and then searching only within that cluster (C) for the closest neighbour (x). This approach reduces complexity by a factor depending on the number of clusters and their uniformity.

k-means clustering tends to create clusters of equal volume. The effectiveness of k-meansICP depends on the distribution of points within the clusters. Evenly distributed points lead to equal-sized clusters in terms of cardinality. An evenly distributed point cloud with cardinality N would be partitioned into k clusters of cardinality N/k. For each of the N points we first find the nearest of the k cluster centroids using $O(k)$ distance calculations, and then then nearest of the on average N/k points in that cluster using $O(N/k)$ distance calculations. Hence the computational complexity of k-meansICP becomes $O(N(k+N/k))$. In theory the optimal choice of k would be $k = \sqrt{N}$ to reduce the total computational complexity to $O(N^{1.5})$, which would be a significant reduction from $O(N^2)$ for the original ICP algorithm. More generally, for non-uniform point clouds the computational complexity of k-meansICP depends on the actual cardinalities of the clusters and

on the probability that each cluster is selected as the closest to the point p.

However, the accuracy of the matches might be affected, particularly for points near cluster boundaries. This is because k-meansICP limits the search space to the chosen cluster C. While C is the cluster with the closest centroid to p, it may occur that the actual closest point is not in C but in a different cluster. In such cases, a suboptimal closest point will be chosen from C. The iterative nature of the ICP algorithm makes it robust to small amounts of such suboptimal correspondences. More suboptimal correspondences may delay or even prevent convergence. To keep the amount of suboptimal correspondences small we keep k relatively small, in practice $k = 2$ to 10. This is typically less than optimal for the theoretical computational complexity, but we have found that it is a better trade-off in terms of convergence in practice.

4 EXPERIMENT

k-meansICP was implemented in Microsoft C#. Unity3D was used to visualise the registration simulation and generate the analysis data.

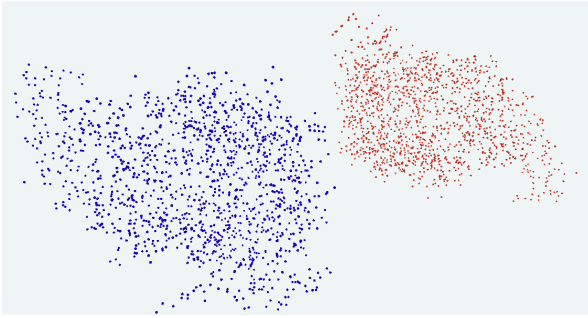

Figure 2: 3D point cloud for molecules 2JKG (left) and 2JKF (right)

4.1 Experimental Design

This experiment investigates the performance of the k-meansICP algorithm for protein structure matching. The focus of the experiment was on how the number of k-means clusters [Kri99] can affect registration accuracy and convergence.

4.2 Data Acquisition and Pre-Processing

For this study, two protein structures from the Protein Data Bank (PDB) [Ber00] serve as the source for protein structure data. The PDB IDs are 2JKF and 2JKG (provided for reproducibility) consisting of 1420 and 1452 atoms respectively. Each protein structure shape will be matched to a rotated version of itself. Each structure was imported into Unity using the CellUnity [Geh15] package to facilitate the visualisation and manipulation of protein structures. The package automatically generated a 3D representation based on the coordinates of each atom in the PDB file. Atoms were visualised as spheres, allowing for spatial analysis of the protein molecule.

4.3 Experimental Setup

The experiment evaluated the performance of the ICP and k-meansICP algorithms under varying initial rotation angles and different numbers of clusters for the k-meansICP algorithm. Specifically, the source and target copy of a protein structure were aligned by their centres of mass, followed by giving one copy a starting rotation along one axis at predefined angles (15°, 30°, and 60°). Each test case was run with 100 iterations for k-meansICP and original ICP. However, in the case of k-meansICP, each test case was also run for $k = 2$ to 10 clusters.

4.4 Rationale

Large initial rotations pose a challenge for ICP algorithms. The introduction of an initial rotation angle was to add some complexity to the registration process.

The original ICP algorithm served as the baseline for comparison, isolating the effects of the proposed improvements within the k-meansICP. Integrating k-meansICP with other ICP variants would potentially introduce confounding factors from their inherent improvements, hindering the accurate measurement of our specific contributions related to the number of clusters and the initial rotation angles.

4.5 Rotational angles and number of clusters

While both algorithms consider translation during the matching process, this experiment primarily focused on rotational alignment and cluster sizes. The pre-alignment of structures at their centres of mass minimizes the impact of the translation vector and is a common pre-processing step in ICP.

5 ANALYSIS AND EVALUATION

To assess and compare the performance of k-meansICP with each number of clusters, as well as the original ICP, two types of graphs are presented. The convergence graphs show the mean squared alignment error at each iteration. This data is generated at the end of each iteration and it presents an analysis of the gradual progress of the registration process towards convergence. It can also show the iteration count at convergence, as well as how smooth the transition is towards convergence.

The match quality graphs show the cumulative histogram of the number of points that have a root mean squared alignment error less than a value on the horizontal axis. This data represents the final quality of the match when the registration process is completed. By

looking at the cumulative number of points at a particular root mean square error, we can analyse the effectual quality of the match. A good match quality would have a steep rise to nearly 100% at small alignment errors. The vertical line that sometimes appears at the right edge of a match quality graph is an artefact of Matplotlib [Hun07] that indicates the maximum alignment error, where the cumulative histogram reaches 100% of points. Note that the horizontal axis is scaled to fit the range of alignment errors. If the match is almost perfect and the maximum alignment error is very small, e.g., 1e-8, then the match quality curve has an irregular shape caused by floating point precision and does not represent significant alignment errors.

5.1 ICP

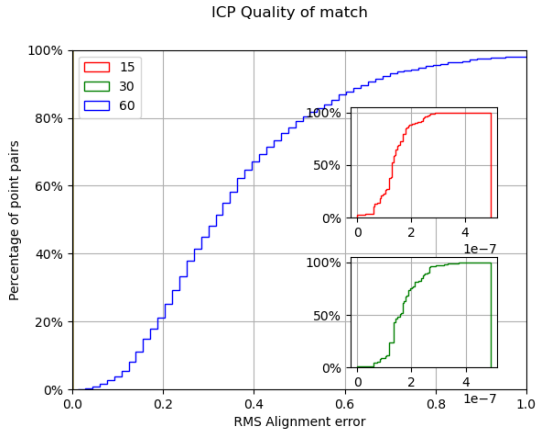

Figure 3: Match quality results for molecule 2JKF using ICP at initial rotation angles 15°, 30°, and 60°

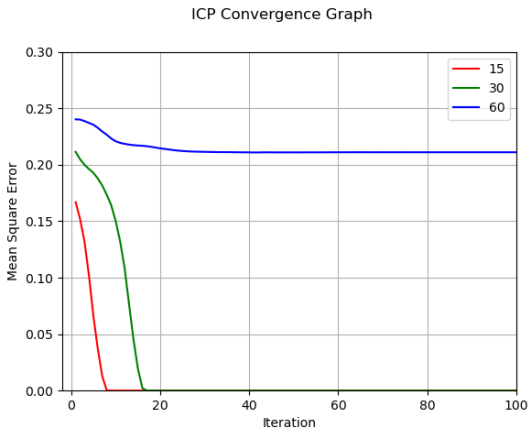

Figure 4: Convergence results for molecule 2JKF using ICP at initial rotation angles 15°, 30°, and 60°

From the matching results for ICP for molecule 2JKF (Figure 3), we realize a good match for initial rotation angles 15 and 30 degrees and a comparatively worse one for angle 60 degrees. This is typical of the ICP algorithm's behaviour of worse convergence with a poor initialisation and thus requiring a good initial transformation. The convergence graph (Figure 4) confirms that performance with the 15 and 30-degree initial rotation angles converging faster at approximately 10 iterations and 18 iterations respectively, whereas the angle 60 degrees has reached a plateau after approximately 30 iterations with a substantial remaining alignment error.

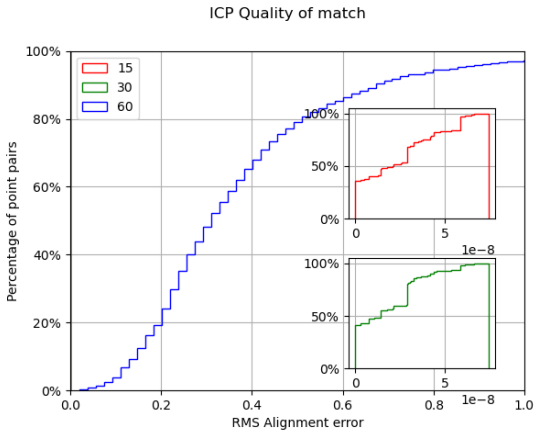

Figure 5: Match quality results for molecule 2JKG using ICP at initial rotation angles 15°, 30°, and 60°

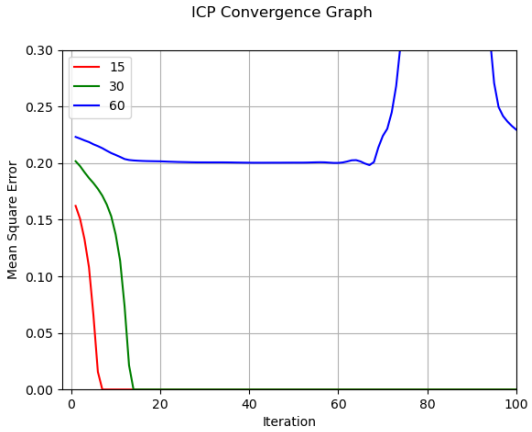

Figure 6: Convergence results for molecule 2JKG using ICP at initial rotation angles 15°, 30°, and 60°

Similarly, for molecule 2JKG, we realize a good match for initial rotation angles of 15 and 30 degrees and a worse match for the angle of 60 degrees (Figure 5). The convergence graph (Figure 6) shows similar convergence results for 15 and 30 degrees. For an initial rotation angle of 60 degrees, the convergence is much worse compared to 2JKF, even showing intermittent divergence around iterations 70–100.

5.2 *k*-MeansICP

It can be observed that the match quality of *k*-meansICP with 2 clusters (Figure 7) was comparable to ICP (3) but only for the 60 degrees initial rotation. ICP performed much better at converging at a lower mean squared

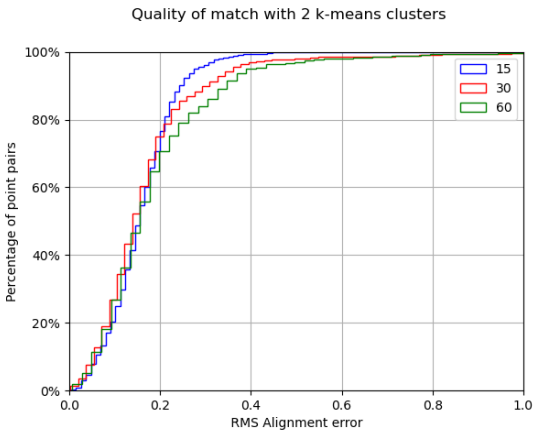

Figure 7: Match quality results for k-meansICP with $k = 2$ clusters

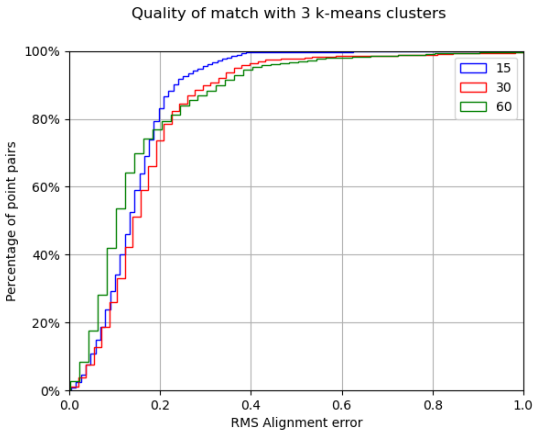

Figure 8: Match quality results for k-meansICP with $k = 3$ clusters

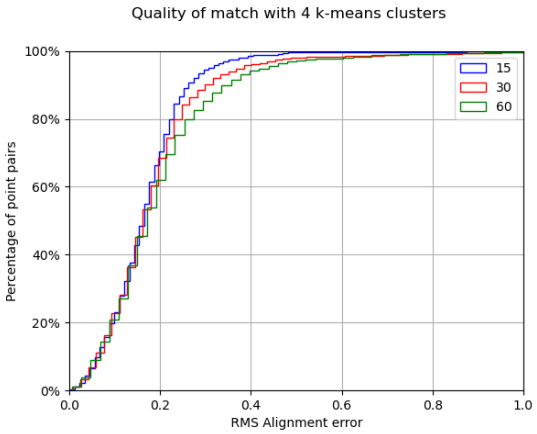

Figure 9: Match quality results for k-meansICP with $k = 4$ clusters

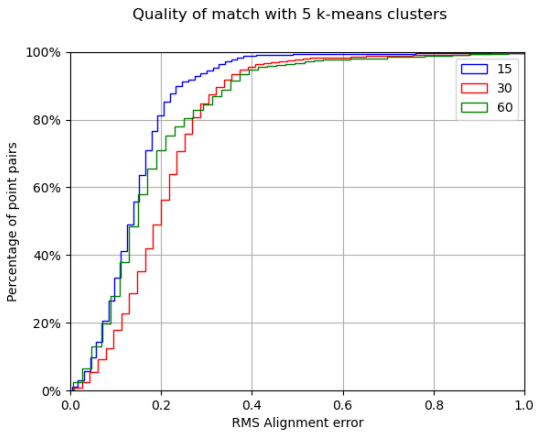

Figure 10: Match quality results for k-meansICP with $k = 5$ clusters

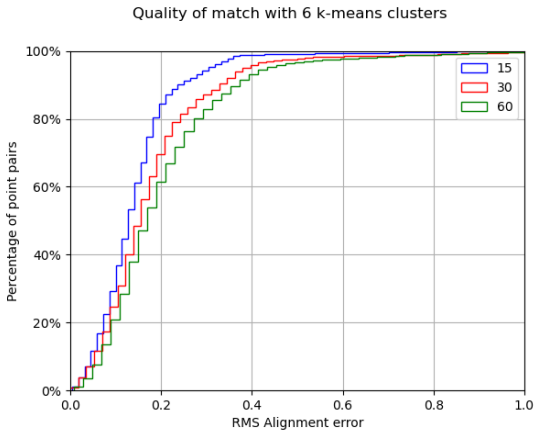

Figure 11: Match quality results for k-meansICP with $k = 6$ clusters

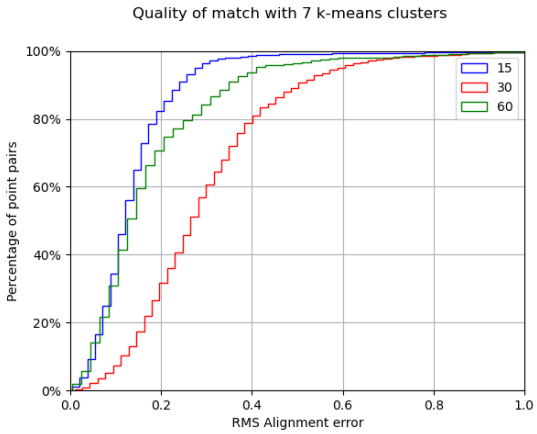

Figure 12: Match quality results for k-meansICP with $k = 7$ clusters

error for all initial rotation angles as compared to k-meansICP (Figure 16). This trend is seen across all the other match quality (Figures 7–15) and convergence graphs (Figures 16–24) for increasing number of clusters, showing lower final point pair alignment errors. This further illustrates the robust nature of the ICP algorithm for registering the same shapes with no noise. Although k-meansICP makes use of k-means clustering to reduce the search space and make the registration faster, this improvement did not correlate with improved registration results at lower convergence errors. k-means clustering [Kri99] is used to accelerate the closest point

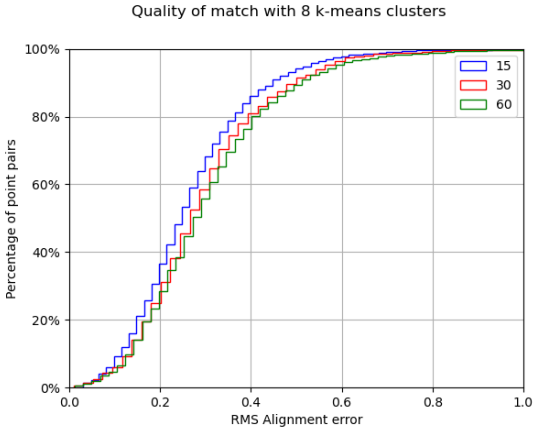

Figure 13: Match quality results for k-meansICP with $k = 8$ clusters

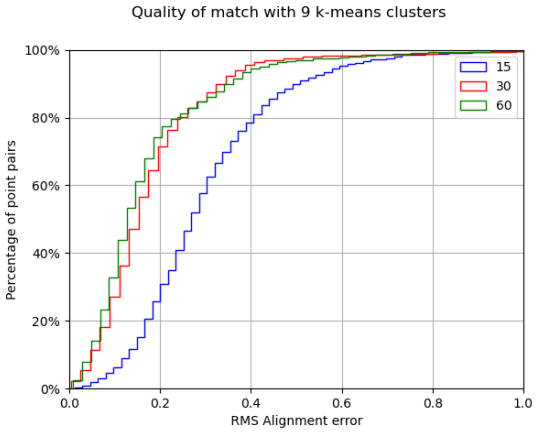

Figure 14: Match quality results for k-meansICP with $k = 9$ clusters

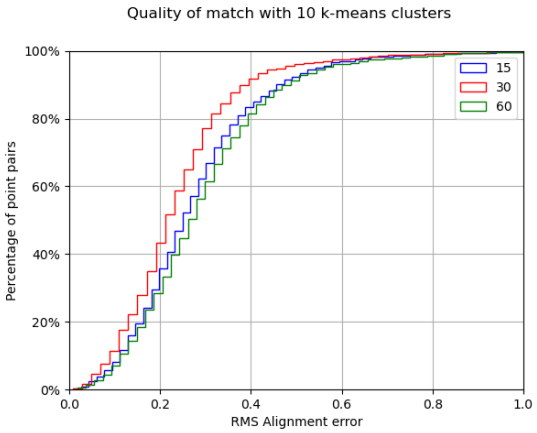

Figure 15: Match quality results for k-meansICP with $k = 10$ clusters

search but can result in less suitable matches leading to less optimal performance.

We also observed that the number of clusters had minimal impact on match quality and convergence.

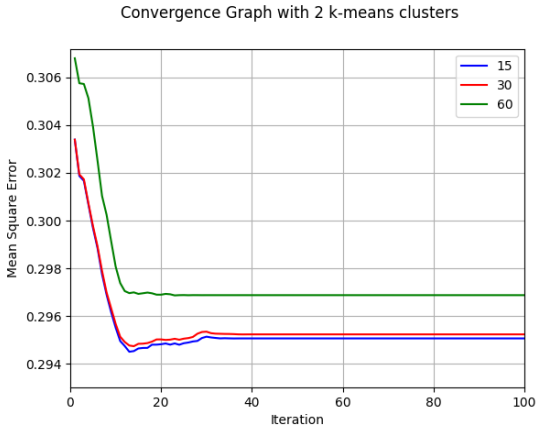

Figure 16: Convergence results for k-meansICP with $k = 2$ clusters

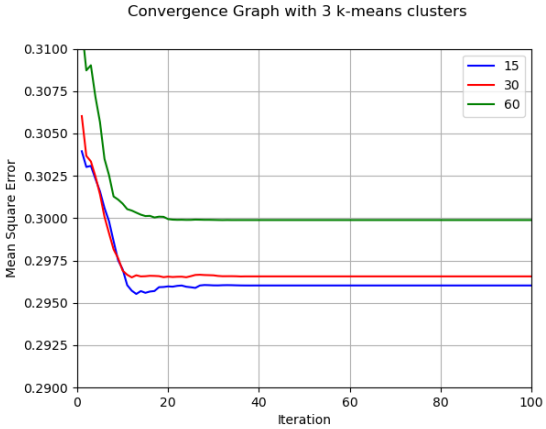

Figure 17: Convergence results for k-meansICP with $k = 3$ clusters

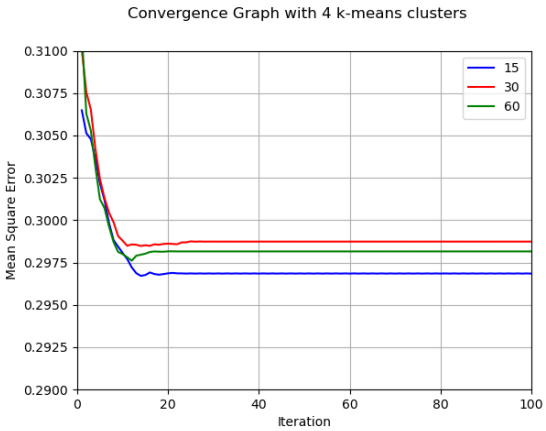

Figure 18: Convergence results for k-meansICP with $k = 4$ clusters

6 CONCLUSIONS AND FUTURE WORK

The original ICP algorithm [Bes92] is the most common algorithm used for point cloud registration, however, it has limitations such as the requirement for a good initial transformation, the computational com-

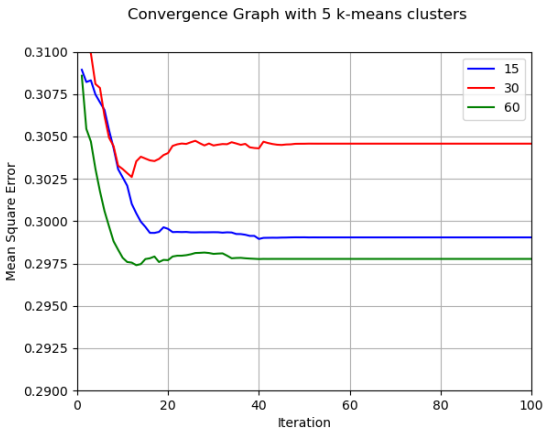

Figure 19: Convergence results for k-meansICP with $k = 5$ clusters

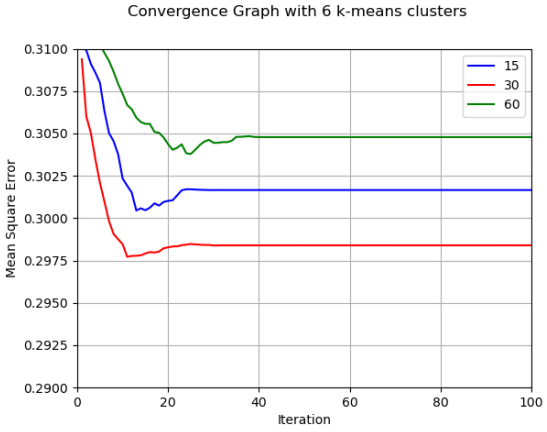

Figure 20: Convergence results for k-meansICP with $k = 6$ clusters

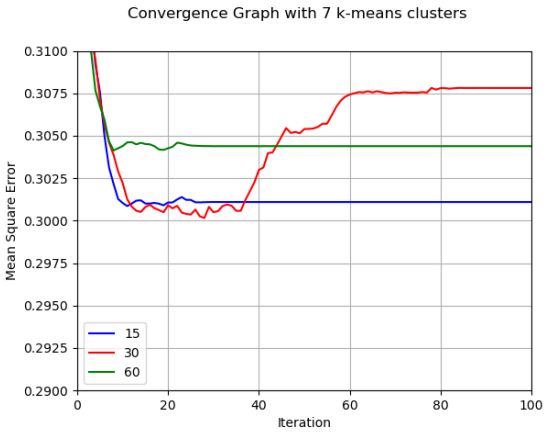

Figure 21: Convergence results for k-meansICP with $k = 7$ clusters

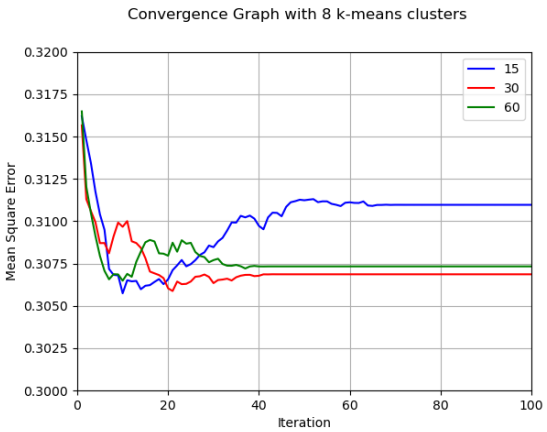

Figure 22: Convergence results for k-meansICP with $k = 8$ clusters

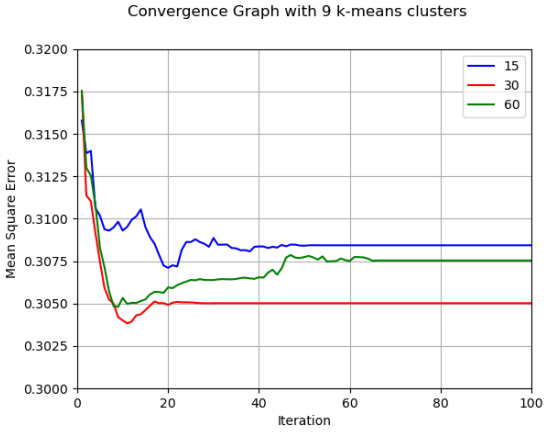

Figure 23: Convergence results for k-meansICP with $k = 9$ clusters

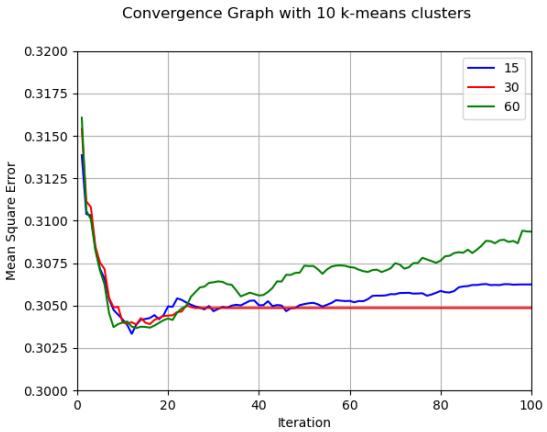

Figure 24: Convergence results for k-meansICP with $k = 10$ clusters

plexity of the correspondence search process, and poor performance in the presence of noise and outliers.

This paper presented the k-meansICP algorithm to improve the performance of ICP by reducing the correspondence search space. The approach partitions the data point cloud using k-means clustering [Kri99] resulting in a decrease in the computational complexity. This work also investigated the impact of the number of k-means clusters on point cloud registration. The results did not show a clear correlation between the number of k-means clusters and the match quality and convergence using the k-meansICP algorithm.

Results indicate a lower quality match and slower convergence compared to ICP. However, the impact of the number of *k*-means clusters on 3D point cloud registration can be investigated further to consider variables such as the symmetry of the point cloud, and implementation options such as using *k*-meansICP for an initial coarse alignment before a fine alignment using ICP.

7 REFERENCES

[Ank20] Ankomah, P., Vangorp, P., Behera, A., and Liu, Y. Tagged-ICP: An Iterative Closest Point Algorithm with Metadata Knowledge for Improved Matching of 3D Protein Structures. In Proc. Irish Machine Vision and Image Processing conference, 2020.

[Ber00] Berman, H., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T., Weissig, H., Shindyalov, I., and Bourne, P. The Protein Data Bank. Nucleic Acids Research 28, no. 1, pp. 235–242, 2000.

[Bes92] Besl, P.J. and McKay, N.D. A method for registration of 3-D shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence 14, no. 2, pp. 239–256, 1992.

[Bo19] Bo, P. and Barton, M. On initialization of milling paths for 5-axis flank CNC machining of free-form surfaces with general milling tools. Computer Aided Geometric Design 71, pp. 30–42, 2019.

[Che11] Chen, J., Xiaojun, W., Wang, M., and Fuqin, D. Human Body Shape and Motion Tracking by Hierarchical Weighted ICP. In Proc. International Symposium on Visual Computing, 2011.

[Du15] Du, S., Liu, J., Zhang, C., Zhu, J., and Li, K. Probability iterative closest point algorithm for m-D point set registration with noise. Neurocomputing 157, pp. 187–198, 2015.

[Geh15] Gehrer, D. CellUnity - an Interactive Tool for Illustrative Visualization of Molecular Reactions. In Proc. Central European Seminar on Computer Graphics, 2015.

[Goj20] Gojcic, Z., Zhou, C., Wegner, J.D., Guibas, L.J., and Birdal, T. Learning multiview 3D point cloud registration. In Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1759–1769, 2020.

[Hun07] Hunter, J.D. Matplotlib: A 2D graphics environment. Computing in Science & Engineering 9, no. 3, pp. 90–95, 2007.

[Iko23] Ikotun, A.M., Ezugwu, A.E., Abualigah, L., Abuhaija, B., and Heming, J. K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data. Information Sciences 622, pp. 178–210, 2023.

[Kan02] Kanungo, T., Mount, D.M., Netanyahu, N.S., Piatko, C.D., Silverman, R., and Wu, A.Y. An efficient k-means clustering algorithm: Analysis and implementation. IEEE transactions on pattern analysis and machine intelligence 24, no. 7, pp. 881–892, 2002.

[Kri99] Krishna, K. and Narasimha Murty, M. Genetic K-means algorithm. IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics) 29, no. 3, pp. 433–439, 1999.

[Kri15] Kriegel, S., Rink, C., BodenmÃ¼ller, T., and Suppa, M. Efficient next-best-scan planning for autonomous 3D surface reconstruction of unknown objects. Journal of Real-Time Image Processing 10, no. 4, pp. 611–631, 2015.

[Li22] Li, J., Hu, Q., Zhang, Y., and Ai, M. Robust symmetric iterative closest point. ISPRS Journal of Photogrammetry and Remote Sensing 185, pp. 219–231, 2022.

[Lyu24] Lyu, M., Yang, J., Qi, Z., Xu, R., and Liu, J. Rigid pairwise 3D point cloud registration: A survey. Pattern Recognition 151, p. 110408, 2024.

[MA23] Monji-Azad, S., Hesser, J., and Löw, N. A review of non-rigid transformations and learning-based 3D point cloud registration methods. ISPRS Journal of Photogrammetry and Remote Sensing 196, pp. 58–72, 2023.

[Ovr23] Ovrebo, O., Ojansivu, M., Kartasalo, K., Barriga, H.M.G., Ranefall, P., Holme, M.N., and Stevens, M.M. RegiSTORM: channel registration for multi-color stochastic optical reconstruction microscopy. BMC Bioinformatics 24, no. 1, p. 237, 2023.

[Pom13] Pomerleau, F., Colas, F., Siegwart, R., and Magnenat, S. Comparing ICP variants on real-world data sets: Open-source library and experimental protocol. Autonomous Robots 34, no. 3, pp. 133–148, 2013.

[Pot06] Pottmann, H., Huang, Q.X., Yang, Y.L., and Hu, S.M. Geometry and convergence analysis of algorithms for registration of 3D shapes. International Journal of Computer Vision 67, pp. 277–296, 2006.

[Rus01] Rusinkiewicz, S. and Levoy, M. Efficient variants of the ICP algorithm. In Proc. 3-D Digital Imaging and Modeling, pp. 145–152, 2001.

[Sal24] Saleh, A.R. and Momeni, H.R. An improved iterative closest point algorithm based on the particle filter and K-means clustering for fine model matching. The Visual Computer ,

2024.

[Wan17] Wang, F. and Zhao, Z. A survey of iterative closest point algorithm. In 2017 Chinese Automation Congress (CAC), pp. 4395–4399. IEEE, Jinan, 2017.

[Wu18] Wu, M.L., Chien, J.C., Wu, C.T., and Lee, J.D. An Augmented Reality System Using Improved-Iterative Closest Point Algorithm for On-Patient Medical Image Visualization. Sensors 18, no. 8, p. 2505, 2018.

[Zha22] Zhang, J., Yao, Y., and Deng, B. Fast and Robust Iterative Closest Point. IEEE Transactions on Pattern Analysis and Machine Intelligence 44, no. 7, pp. 3450–3466, 2022.