3D Multi-Angle Point Cloud Stitching Using Iterative Closest-point Stitching and K-Nearest-Neighbors

Pankti Patel
Dept. of Electrical and Computer Engineering
Rowan University
Glassboro, USA
pankti2047@gmail.com

Ryan Hare
Dept. of Electrical and Computer Engineering
Rowan University
Glassboro, USA
harer6@rowan.edu

Ying Tang
Dept. of Electrical and Computer Engineering
Rowan University
Glassboro, USA
tang@rowan.edu
(corresponding author)

Nidhi Patel
Dept. of Electrical and Computer Engineering
Rowan University
Glassboro, USA
patelnidhig1@gmail.com

***Abstract*— The recent focus on virtual environments and 3D object scanning has highlighted the need for accurate and efficient methods to stitch concurrent point clouds into solid three-dimensional (3D) models. To address this need, we introduce a novel iterative approach for 3D multi-angle point cloud stitching using an iterative closest point (ICP) algorithm augmented with k-nearest neighbors (kNN). With this combined algorithm, our method focuses on minimizing the error between neighboring point clouds, allowing us to easily compute the necessary transformation to combine point clouds into one model. Thus, when given concurrent point clouds captured at multiple angles of the same object, our approach provides a single accurate 3D model. We evaluated the ability of the proposed framework to stitch multiple point clouds into a solid model by stitching a segmented model and comparing the root mean squared error to a standard iterative closest-point stitching algorithm. The experiments results shows that our method provides benefits in terms of efficiency and accuracy compared to a standard approach.**

Keywords—ICP, kNN, point-cloud stitching, 3D point clouds.

I. Introduction

In recent years, advancements in computer simulations and virtual environments have created an ever-increasing need for methods to scan real objects and reconstruct them into accurate three-dimensional (3D) models. Generating 3D models using legacy modelling software is time consuming, complicated, and expensive [1]. In virtual environments, reconstructing real objects significantly reduces the time required to create realistic 3D objects. Additionally, accurate scans of real objects also open new possibilities for computer-aided or entirely digitized methods for applications such as skin health diagnoses and manufacturing defect recognition. However, in all applications, easy and rapid object scanning is vital, as is the accuracy of the resulting 3D model.

One of the most common methods of 3D object scanning is using 3D laser scanners, which can accurately capture multiple angles of the same object, allowing a user to easily reconstruct an accurate 3D model. However, laser scanners are economically prohibitive and unwieldy, often requiring multiple people to successfully operate and scan objects [30]. More cost-effective and portable methods of 3D object scanning can be found in visual scanning using structured light methods and depth cameras. However, visual and depth camera methods do not automatically combine scanned objects into complete and solid 3D models. Instead, these methods return discrete point clouds from each angle captured on a target object.

It is necessary to stitch multiple point clouds to retrieve solid 3D models from discrete, segmented point clouds. Although this process can be performed manually, it is extremely time-consuming, and automatic methods for point cloud stitching would greatly improve the usability of both visual and depth camera-based scanning methods. Several methods [3][6] to stitch point clouds exist in the literature, but there are several issues with the existing methods: 1) methods require modified or transformed input data, adding additional complexity and time to the algorithm, and 2) methods are prone to high error or inaccuracy in stitching methods. As such, there is still room for fast and accurate methods for creating a 3D model from a series of concurrent point clouds.

A common approach for aligning neighboring point clouds is the iterative closest point (ICP) algorithm. The functional step is to determine the transformation that best matches the point clouds with a given correspondence [5]. ICP implicitly assumes that there is a good overlap between the source and target point clouds so that stitching can easily converge [3][4]. While simple and effective, ICP presents the following practical shortcomings owing to the alignment assumption: (1) high computational cost [15]; (2) it is prone to converge to local minima [14]; and (3) it is easily affected by outliers [16]. To address these limitations, researchers have devoted efforts to modifying the standard ICP algorithm, leading to several ICP variants in a range of applications in medical fields [19], remote sensing [12], autonomous driving [12][7], robotics [20], and aviation [21].

Among the modified approaches, one of the simpler yet effective modifications of ICP is to weigh point pairs to modify their impact on the transformation or to reject unmatched pairs outright [17]. Another method of Hybrid ICP [23] is proposed which optimizes the data association method and error metric based on the live image of an object and the current ICP estimate [23]. An alternative method to improve the registration of ICP is by introducing a point-to-plane metric that utilizes the surface information of the point sets to improve registration accuracy [16]. Another line of based on identified features. For example, Zong *et al.* applied an improved scale-invariant feature transform (SIFT) method to help match points [8]. However, as in [2][10], this method requires data manipulation to convert the 3D data into 2D images. Once the point cloud is converted, the 2D images are stitched together before being converted back. In this approach, the conversion to and from 2D has a high computational cost, and information such as the normal directions and colors of points can be lost in the conversion. Furthermore, this conversion may not always be possible because of different data formats [10]. Therefore, methods that operate directly on the 3D point cloud are desirable to avoid the afore mentioned issues.

Motivated by the issue of automatic point cloud stitching and recent advancements in machine learning methods, this study presents an efficient algorithm for stitching concurrent point clouds into single 3D models. Specifically, our proposed algorithm augments the iterative closest point (ICP) stitching algorithm with k-nearest-neighbors (kNN) clustering. Using this combined approach, our method iteratively minimizes the error between two neighboring point clouds, locating the corresponding points on both point clouds to combine and create a solid 3D model. Compared to a standard ICP algorithm, the addition of kNN allows us to obtain a more accurate matching between two neighboring point clouds.

Thus, with our proposed algorithm, this paper makes the following contributions:

1. We propose a general-purpose iterative algorithm for creating a solid 3D model from multiple concurrent point clouds scanned from the same object.
2. Our proposed method augments an iterative closest-point algorithm with k-nearest neighbors, creating a method that achieves high efficiency and reduced error compared to standard algorithms.
3. The proposed method operates directly on 3D point clouds, avoiding computationally expensive transformations.

The remainder of this paper is organized as follows. Section II outlines the proposed point-cloud stitching algorithm. Section III provides the experimental results of our method compared to the standard ICP, followed by the conclusions in Section IV.

II. PROPOSED STITCHING ALGORITHM

As stated, our proposed algorithm is a combination of the iterative closest point (ICP) algorithm and k-nearest neighbor (kNN) similarity measures. Fig. 1 shows a flowchart of the proposed method. The fundamentals of ICP are first presented in Section II.A, followed by the proposed kNN augmentation in Section II.B.

A. Fundamentals of Interactive Closest Point Algorithms

When two point clouds are given, target point cloud A and source point cloud B, the optimal alignment between them can be computed directly. Equation 1 shows the mathematical solution for determining the transformation t and rotation matrix R required to transform and rotate B to A.

$$\varepsilon(R,T) = \sum_{i=1}^{n_A} \sum_{j=1}^{n_B} \left\| a_i - \left(R \cdot b_j + t\right) \right\|^2 \quad (1)$$

where $|A| = n_A$ and $|B| = n_B$.

R= $\begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$ is the rotation matrix that describes the Euclidian space transformation used to rotate B to match A (as closely as possible). $t = [t_x, t_y, t_z]^T$ is the transformation matrix which describes the Euclidean space transformation used to move the center of mass of B to the center of mass of A. Finally, there is a set of paired points $\mathrm{P} = \{(a_i, b_j) | a_i \in A, b_j \in B\}$ and $|\mathrm{P}| = N$ is the number of points in P.

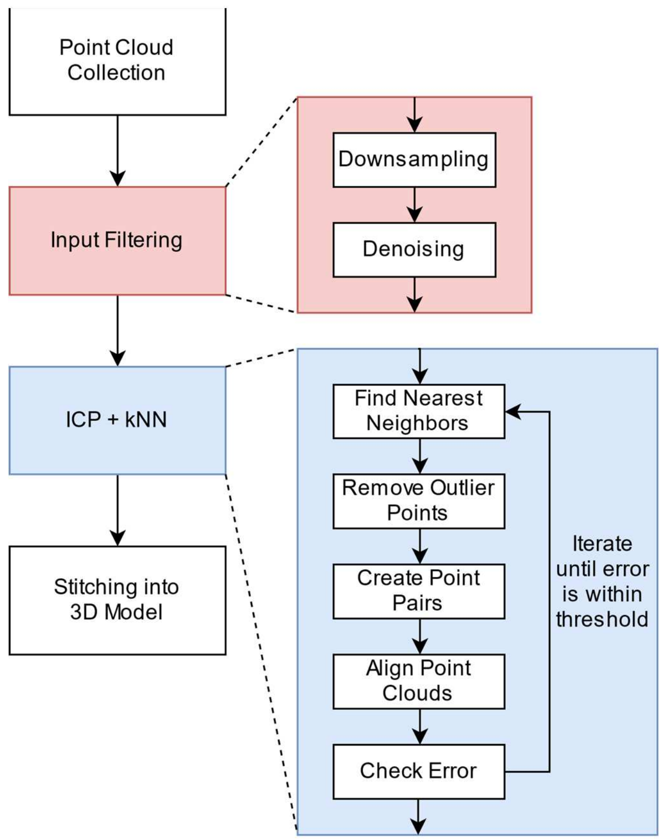


Fig. 1. Workflow of stitching Algorithm

However, this approach makes several assumptions. First, it is assumed that there exists a set of paired points P that define the corresponding points in both point clouds. This assumption is valid when aligning two known or identical point clouds because the paired points are either already known or are easy to determine. With point clouds captured from real objects, pairing is not known. The iterative closest point (ICP) approach is a means by which this pairing can be guessed to align two unknown point clouds.

ICP works through two basic steps. First, the paired points are estimated. This estimation is typically performed by using a similarity or distance metric. For example, for each point in B, the paired point is the nearest point in A. Using these pairs and Equation 1, a sub-optimal rotation and translation can then be computed, and point cloud B can be shifted in the direction of point cloud A. Finally, the approach is iterated until some stopping criteria are reached, typically minimum error, maximum iterations, or minimum shift per iteration. Through this approach, two unknown point clouds can be aligned without knowledge of the paired points. However, ICP can still run into issues owing to the lack of properly matched points, such as slow convergence times, incorrect final transformations, or converging upon local minima instead of global minima.

B. *Integration of ICP with kNN*

Owing to the aforementioned issues with the basic ICP, we propose augmenting the standard algorithm with kNN. Rather than estimating a single pair for all points, the proposed method finds a set of k candidate points. For each point in B, we first find the k nearest points in A using Euclidean distance, as shown in Equation 2. We can then consider all k points in A as paired points to a single point in B and add k pairs to the set of paired points.

$$d_i = \sqrt{\sum_{j=1}^{n_B} (a_i - b_j)^2} \quad (2)$$

Algorithm 1 Full Stitching Algorithm

Inputs: Source point cloud A
Target point cloud B
$minError$

1: **Initialize** $k = \sqrt{|A|}$
2: **While** $\varepsilon > minError$ **do:**
  a. Call **Algorithm 2** to get list of paired points P
  b. Compute t and R using Equation 1, and get total error ε
  c. Transform $B = B * R + t$
3: **End while**
4: Transform $B = B * R + t$
5: Combine $X = A + B$
6: Apply a box filter on X, merge all nearby points into a single point, and average color and normal information.
7: **Return** merged point cloud X

As the proposed method adds a much larger number of pairs to the paired list than the standard, we can also remove any outlier pairs. To remove outliers, we discarded any point pairs with distances above a certain threshold, typically to reduce the number of pairs to n_A. Algorithm 1 formalizes the entire process, and Algorithm 2 shows a single iteration of ICP with kNN. The proposed approach helps in making the experiments robust to noise and prevent convergence on local minima due to identifying best matching pair by considering more than one point. Furthermore, the method helps in optimizing the speed as we displace the cloud towards optimal location per iteration compared to a standard approach.

The final important aspect of the algorithm is selecting a value for k. When the value is too large, the number of additional pairs added to the pair list becomes very large, slowing the convergence of the algorithm. However, as k approaches 1, the algorithm behaves in a similar manner to the standard ICP. Through our testing, we determined that $k = \sqrt{|A|}$, where $|A|$ is the number of points in A, provides a balanced value for k with consistent performance throughout our testing.

Algorithm 2 One Iteration of ICP with kNN

Inputs: Source point cloud A
Target point cloud B
Neighbor amount k

1: **Initialize** distance set $D = \emptyset$ and pair set $P = \emptyset$
2: **For** each point $b_i \in B$ **do:**
  a. **For** each point $a_j \in A$ **do:**
    i. Compute d_j from equation 2 as the distance between a_i and b_j
    ii. Append $d_j \rightarrow D$
  b. **End for**
  c. Sort D by distance
  d. **For** $a_i \in A$ corresponding to the k smallest values in D **do**:
    i. Append $(a_i, b_j) \rightarrow P$
  e. End for
3: End for
4: From P, remove highest-distance pairs until $|P| = |A|$ to remove outlier pairs.
5: **Return** P

III. Experiments And Discussions

To fully understand the proposed approach, a series of experiments was conducted to evaluate its significance with respect to the basic ICP.

Our experimental setup consisted of an Intel RealSense F200 RGB-D camera placed on the same flat surface as the target object. As shown in Fig. 2, the camera was kept still, while the object was gradually displaced following the marked angle lines underneath. The data were then captured and collected using our project-camera calibration system [11].

Considering that the raw point cloud data have large number of points which are prone to noise, ungrouped points, and outliers, both downsampling [13] and denoising are used to preprocess the data. For downsampling, a box grid filter [18] is applied, where all points that fall within the box are merged into a single point, with the final color value and normal vector of the point obtained by averaging all other points. In our case, downsampling was applied through the *pcdownsample* function in MATLAB [22]. Denoising was then performed using MATLAB's *pcdenoise* function [9]. Finally, the root-mean-square error (RMSE) for the transformation defined below in Equation 3 was considered as the evaluation criterion.

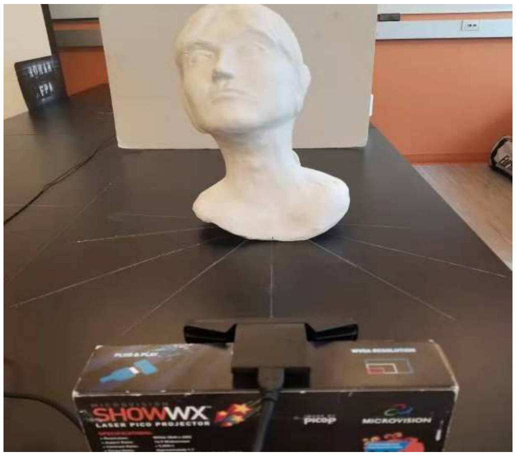


Fig. 2. Experimental Setup

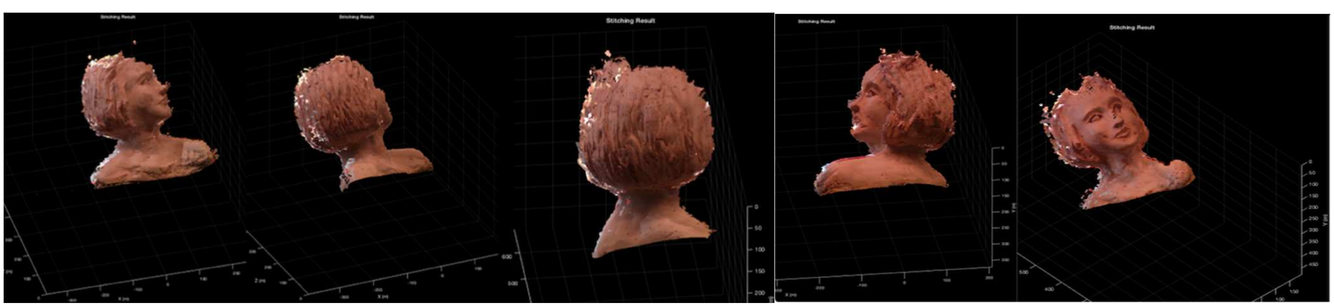
$$\mathrm{RMSE} = \sqrt{\frac{\sum_{j=1}^{M} \sum_{i=1}^{N_j} \left\| \chi_j(a_i) - \widehat{\chi}_j(a_i) \right\|^2}{\sum_{j=1}^{M} N_j}} \qquad (3)$$

where $\chi_j(a_i)$ is the coordinate of point a_i in the source cloud, and $\widehat{\chi}_j(a_i)$ is the corresponding coordinate transformed from the reference cloud using the derived parameters R and t. The total number of points considered was the sum of the points that could be paired in M clouds.

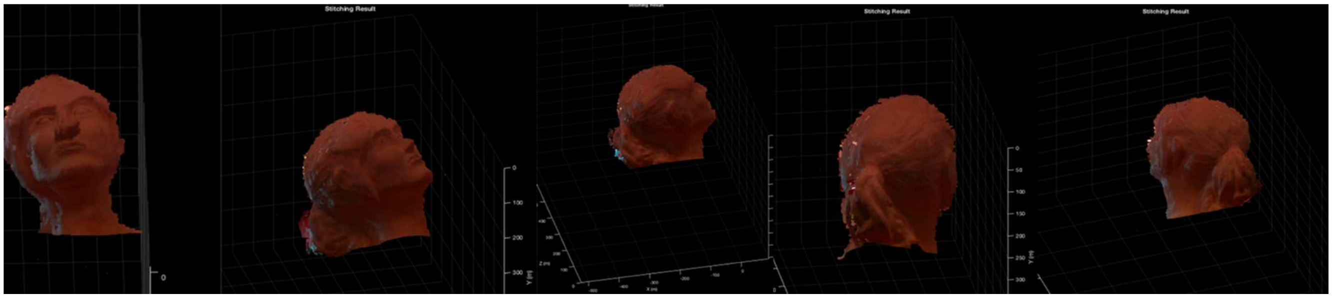
In the experiment, three busts of various features were used to populate sets of point clouds. First, the proposed *kNN*-ICP method was used to reconstruct them, as shown in Fig. 3. Second, the same datasets are fed into the basic ICP. The average of the iterations and RMSE were then calculated and compared between the two algorithms. As shown in Table I, the proposed method outperformed the basic ICP.

TABLE I. COMPARISON WITH RMSE OF STANDARD ICP AND MODIFIED ICP AND KN

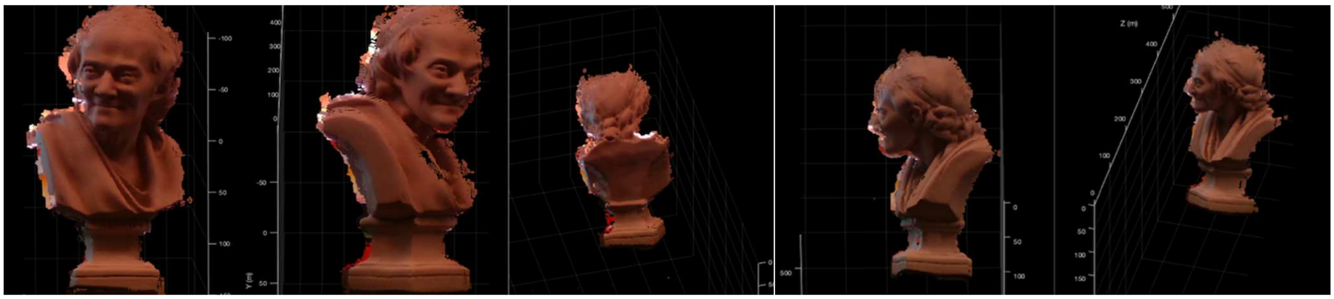
Object	Attribute	Average Iteration per Cloud	Root Mean square Error
	ICP	18	3.6987
	ICP with kNN	16	1.9403
	ICP	20	3.9789
	ICP with kNN	14	2.4637
	ICP	25	4.9372
	ICP with kNN	20	2.4738



(a)



(b)



(c)

Fig. 3. (a) Stitching Result 1. (b) Stitching Result 2. (c) Stitching Result 3

IV. CONCLUSION

Recent trends in 3D modelling have highlighted the need for more accessible methods for stitching real objects into 3D models. To address the need for more accurate and efficient stitching methods, this paper proposes an incremental approach for 3D multi-angle point cloud stitching using an iterative closest point algorithm augmented with k-nearest-neighbors. From our experimental results, we first visually demonstrate the models obtained by stitching together a series of neighboring point clouds. Compared to other stitching methods, our proposed algorithm has reduced error and computational intensity because it does not need to reformat the data and instead operates directly on the 3D point cloud. Furthermore, our quantitative results demonstrate that kNN augmentation leads to a lower root-mean-squared error compared to the standard iterative closest point algorithm. Overall, the proposed method is efficient, accurate, and robust for stitching 3D point clouds. Future work will focus on implementing and testing algorithms for less textured and flat surfaces.

REFERENCES

[1] X. Chen, Q. Wu and S. Wang, "Research on 3D Reconstruction Based on Multiple Views," 2018 13th International Conference on Computer Science & Education (ICCSE), 2018, pp. 1-5, doi: 10.1109/ICCSE.2018.8468705.

[2] P. Barrios and M. Adams, "Point set registration based on multi-object metrics," 2017 International Conference on Control, Automation, and Information Sciences (ICCAIS), Chiang Mai, 2017, pp. 245-250, doi: 10.1109/ICCAIS.2017.8217584.

[3] P. Kamencay, M. Sinko, R. Hudec, M. Benco and R. Radil, "Improved Feature Point Algorithm for 3D Point Cloud Registration," 2019 42nd International Conference on Telecommunications and Signal Processing (TSP), 2019, pp. 517-520, doi: 10.1109/TSP.2019.8769057.

[4] J. Procházková and D. Martišek, “Notes on Iterative Closest Point Algorithm,” 17th Conference on Applied Mathematics APLIMAT, Apr. 2018.

[5] A. Makovetskii, S. Voronin, V. Kober and A. Voronin, "Regularized point-to-point and point-to-plane functionals in the point clouds registration problem," 2021 International Conference on Information Technology and Nanotechnology (ITNT), 2021, pp. 1-6, doi: 10.1109/ITNT52450.2021.9649208.

[6] W. Hou, D. Li, C. Xu, H. Zhang and T. Li, "An Advanced k Nearest Neighbor Classification Algorithm Based on KD-tree," 2018 IEEE International Conference of Safety Produce Informatization (IICSPI), 2018, pp. 902-905, doi: 10.1109/IICSPI.2018.8690508.

[7] Authors, "A Brief Survey on 3D Semantic Segmentation of Lidar Point Cloud with Deep Learning," 2021 3rd Novel Intelligent and Leading Emerging Sciences Conference (NILES), 2021, pp. 405-408, doi: 10.1109/NILES53778.2021.9600493.

[8] B. Zhong and Y. Li, "Image Feature Point Matching Based on Improved SIFT Algorithm," 2019 IEEE 4th International Conference on Image, Vision and Computing (ICIVC), 2019, pp. 489-493, doi: 10.1109/ICIVC47709.2019.8981329.

[9] The MathWorks Inc., “Ptcloudin,” Remove noise from 3-D point cloud - MATLAB. [Online]. Available: https://www.mathworks.com/help/vision/ref/pcdenoise.html. [Accessed: 20-Jun-2022].

[10] S. Wang, Y. Wang, M. Jin, X. D. Gu and D. Samaras, "Conformal Geometry and Its Applications on 3D Shape Matching, Recognition, and Stitching," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 29, no. 7, pp. 1209-1220, July 2007, doi: 10.1109/TPAMI.2007.1050.

[11] B. Huang, Y. Tang, S. Ozdemir and H. Ling, "A Fast and Flexible Projector-Camera Calibration System," in IEEE Transactions on Automation Science and Engineering, vol. 18, no. 3, pp. 1049-1063, July 2021, doi: 10.1109/TASE.2020.2994223.

[12] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu and M. Bennamoun, "Deep Learning for 3D Point Clouds: A Survey," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 43, no. 12, pp. 4338-4364, 1 Dec. 2021, doi: 10.1109/TPAMI.2020.3005434.

[13] Y. Lin, Y. Huang, S. Zhou, M. Jiang, T. Wang and Y. Lei, "DA-Net: Density-Adaptive Downsampling Network for Point Cloud Classification via End-to-End Learning," 2021 4th International Conference on Pattern Recognition and Artificial Intelligence (PRAI), 2021, pp. 13-18, doi: 10.1109/PRAI53619.2021.9551070.

[14] H. Kim, H. K. Hong and B. H. Lee, "Improved Iterative Closest Point Algorithm using Truncated Signed Distance Function," 2018 18th International Conference on Control, Automation and Systems (ICCAS), 2018, pp. 1620-1623.

[15] D. Li, A. Wang, P. Ren and L. Wu, "An Allowance Optimal Distribution Method Based on Improved Iterative Closest Point Algorithm," 2018 10th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA), 2018, pp. 515-518, doi: 10.1109/ICMTMA.2018.00130.

[16] Q. Xie et al., "Precise Point Set Registration Algorithm for Indoor Scene Reconstruction," 2019 Chinese Automation Congress (CAC), 2019, pp. 4336-4339, doi: 10.1109/CAC48633.2019.8997195.

[17] Nishino K, Ikeuchi K. Robust, “Simultaneous registration of multiple range images comprising a large number of points,” Electronics & Communications in Japan, vol. 87(8), pp.61-74, 2004.

[18] M. J. Rakotosaona, V. La Barbera, P. Guerrero, N. J. Mitra, and M. Ovsjanikov, “pointcleannet: Learning to denoise and remove outliers from dense point clouds,” Computer Graphics Forum, vol. 39, no. 1, pp. 185–203, 2019.

[19] W. Wei, J. Xi and J. De, "Dental Hard Tissue Auto-registration Based on Modified ICP Algorithm," 2017 International Conference on Computer Systems, Electronics and Control (ICCSEC), 2017, pp. 1267-1274, doi: 10.1109/ICCSEC.2017.8446980.

[20] K. J. Weber and J. Seo, "A Mobile Robotic Application of Naive Multi-directional Stitching with SIFT," 2021 21st International Conference on Control, Automation and Systems (ICCAS), 2021, pp. 1889-1894, doi: 10.23919/ICCAS52745.2021.9649791.

[21] V. Renò, M. Nitti, M. di Summa, R. Maglietta and E. Stella, "Comparative analysis of multimodal feature-based 3D point cloud stitching techniques for aeronautic applications," 2020 IEEE 7th International Workshop on Metrology for AeroSpace (MetroAeroSpace), 2020, pp. 398-402, doi: 10.1109/MetroAeroSpace48742.2020.9160183.

[22] The MathWorks Inc., “Ptcloudin,” Downsample a 3-D point cloud - MATLAB. [Online]. Available: https://www.mathworks.com/help/vision/ref/pcdownsample.html. [Accessed: 20-Jun-2022].

[23] K. Dreczkowski and E. Johns, "Hybrid ICP," 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2021, pp. 6801-6808, doi: 10.1109/IROS51168.2021.9636600.