The Journal of the Society for Art and Science, Vol. 24, No. 2, pp. 6:1 – 6:11 (2025)

A Study on Automatic Registration of 3D Point Clouds Obtained from Archaeological Trench Investigations

Maral Enkhtugs[1)] **Mengbo You**[1)] **Maki Tarora** [2)]
Hiromi Hirakawa[3)] **Satoru Nakazono**[4)] **Kouichi Konno**[1)]

1) Graduate School of Art and Science, Iwate University
2) Kyushu National Museum
3) Nara National Research Institute for Cultural Properties
4) The International University of Kagoshima

{g0323212, ymb, konno21} (at) iwate-u.ac.jp

Abstract

Traditional archaeological excavations often record only limited details of artifacts and sites, making post-excavation analysis difficult. A novel approach, consistent 3D excavation, records every detail of the excavation process by scanning both artifacts and the excavation site to create 3D models. However, the current method of registering the artifact and site models is entirely manual operation, requiring significant labor and time, and the result depends heavily on the operator's skill. This paper proposes a method to automate the registration of 3D point clouds from an archaeological trench investigation site with individual artifacts (stones) in a virtual space. The method integrates histogram-based color segmentation and a region-growing algorithm to extract the topside of stones from the trench point cloud. The segmented stone points are then registered with the complete stone point cloud, using initial alignments to refine the Iterative Closest Point (ICP) results, creating a virtual representation of the ruin. To evaluate the effectiveness of the proposed method, experiments were conducted on two different trench datasets. The results show that the method achieves high segmentation and registration accuracy, while significantly reducing manual effort and improving efficiency.

Keywords: Point cloud, Trench investigation, Shape matching, Registration, Segmentation

1 Introduction

Archaeological excavation is fundamental for uncovering historical artifacts and understanding ancient cultures. Traditional excavation methods involve systematic digging and recording of artifact positions and contexts. Archaeologists have traditionally used manual techniques to document excavation sites. The conventional approach has often involved plane table surveys [1] in the past. More recently, commercial products like Total Station [2] have been used to record the xyz coordinates of each artifact more quickly. However, these methods lack the reproducibility needed for precise ruin documentation.

To address these limitations, a novel approach known as "Consistent 3D excavation" [3] has been introduced, which involves 3D scanning of excavation sites and each artifact, allowing for the creation of detailed 3D models. By registering these artifact models into the 3D model of the entire archaeological trench investigation site, archaeologists can achieve a more accurate and thorough analysis of the excavation data. Manual registration between the point cloud of the excavation site and the point cloud of the excavated artifact is necessary to ensure reproducibility.

Manual registration process is time-consuming and highly dependent on the skill and judgment of the operator, leading to inconsistencies and potential inaccuracies. Therefore, developing automated techniques for registering 3D artifact models with excavation site model is required. Automation will not only smooth the process but also ensure consistent and accurate integration of 3D point clouds, regardless of the operator.

This paper proposes a method to automate the registration of 3D point clouds obtained from archaeological trench investigations, focusing on segmentation and initial alignment. By implementing these automated techniques, we aim to enhance the efficiency and accuracy of consistent 3D excavation, making it a more practical and effective approach for archaeological research. Based on our dataset, hereinafter artifacts are referred to as 'stones' and the excavation site is referred to as a 'trench'.

2 Related works

2.1 Point cloud segmentation

Segmentation of 3D point clouds involves assigning the same label to points that share common characteristics. In general, there are five types of segmentation: edge-based, region-growing, hybrid, model-fitting, and machine learning [4][5].

Edge-based segmentation algorithms identify edges by detecting abrupt changes in normal vectors or point curvatures. The detected edge points are connected and the entire set is divided into independent multiple-point sets for segmentation. While edge-based methods are effective for point clouds with distinct edges, they often struggle with noisy data and smooth surfaces.

Region-growing methods classify nearby points with similar attributes using neighborhood information. This approach starts with a small seed region and expands by incorporating neighboring points that meet specific criteria, such as similar normal vectors, curvatures, or geometric features. Wang *et al.* [6] combined color information with normal angles for region-growing, while Kang *et al.* [7] used Gaussian curvature and normal angle to adjust thresholds. Zhen *et al.* [8] introduced octree-based voxelization into a traditional region-growing method and used color features for fine segmentation. These methods are more accurate than edge-based algorithms and are beneficial for segmenting smooth surfaces. However, region-growing methods can be sensitive to noise, resulting in over-segmentation or segmentation holes.

Hybrid segmentation methods combine two or more techniques to exploit the strengths of each method while bypassing their weaknesses.

Model-fitting approaches divide point clouds into predefined geometric shapes such as cylinders, spheres, cones, and planes. RANSAC is a widely used algorithm in this category. Model-based methods are effective for detecting and fitting regular geometric shapes but may struggle with complex or irregular shapes.

Machine learning segmentation uses algorithms that learn from data to segment point clouds. Common methods include hierarchical clustering,

K-means, and mean shift. These approaches often require substantial labeled training data and significant computation time.

Our dataset has smooth surfaces and does not conform to simple geometric shapes, making edge-based and model-fitting methods less suitable. Machine learning and hybrid segmentation methods are complex to implement and require additional time. Therefore, this paper utilizes a region-growing algorithm, which is suitable for smooth surfaces and colored point clouds.

2.2 Matching algorithm

Various methods, including deep learning, have been proposed for object recognition using 3D point clouds [9]. PointNet++ [10], PointRCNN [11], and PV-RCNN [12] utilize neural networks to extract features, enabling shape recognition for objects with distinct features. However, these methods are not reported for objects like stone tools, which lack prominent features [13]. Additionally, for matching 3D point clouds, Chang *et al.* [14] introduced a similarity measure based on the central axis of 3D shapes, while Liu *et al.* [15] developed a method for classifying 3D shapes from various angles. [14] and [15] are effective for classifying shapes with clear geometric features but face challenges with minimal distinct features data, such as stone tools [16].

2.3 Global alignment methods

Global alignment techniques for point clouds use features such as SHOT [17], FPFH [18], and SUPER 4PCS [19]. SHOT uses histograms of normal vectors within a spherical region for matching. FPFH creates histograms based on angular relationships of k-nearest neighbors' normal vectors. SUPER 4PCS uses sets of four coplanar points for efficient matching. While these methods require high overlap for effective matching, the dataset used in this paper has a low overlap ratio.

On the other hand, the Iterative Closest Point (ICP) algorithm is widely used for 3D shape registration. Furukawa *et al.* [20] applied ICP to combine incomplete laser point clouds with the points created by photogrammetry. Takahashi *et al.* [16] used ICP for partial matching to align the point clouds of stone tools with their surfaces. Principal Component Analysis (PCA) was used for initial alignment by aligning the local axes of the point clouds to provide a suitable starting point for ICP. Similarly, Tian *et al.* [21] developed a PCA-ICP method, where PCA provides coarse registration followed by ICP for fine alignment.

Therefore, this paper proposes a method for estimating initial positions through predefined patterns to rotate the segmented point clouds around the excavated stone.

3 Proposed Method

This paper proposes a method for registering the stone and trench point clouds, which involves two steps: ① Segmentation, segmenting the corresponding part of the stone from the trench and ② Initial alignment, aligning the stone and segmented point set using PCA and ICP algorithm.

3.1 Segmentation

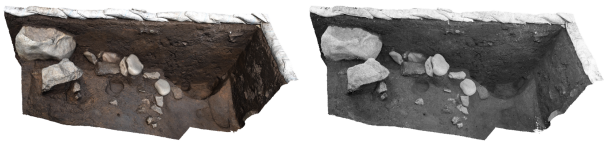
3.1.1 Color segmentation

Since the original point cloud measured from the trench area is very large, performing segmentation directly on this dataset has a high calculation cost. To address this issue and efficiently distinguish the stones from the surrounding environment, the RGB color information of the point cloud is utilized for rough segmentation.

This paper assumes that the trench point cloud has distinct color differences between the stone and soil parts, as shown in Figure 1 (a). Since region-growing segmentation based on distance and normal vector angles cannot effectively distinguish between stone and soil, only stones with colors distinguishable from soil, such as white-colored stones, are considered.

A grayscale histogram is utilized to distinguish the stones from the soil. First, the grayscale values are computed from RGB using the NTSC formula [22] in Equation (1).

$$I = 0.299 \cdot r + 0.587 \cdot g + 0.114 \cdot b \qquad (1)$$



(a) (b)

Figure 1: (a) Raw color data of Trench 1. (b) Converted grayscale data.

where r, g, b represent the RGB color components of a point and I is the calculated grayscale intensity. For example, the color data of Trench 1 and the resulting grayscale point cloud is shown in Figure 1.

Figure 2 shows the grayscale intensity histogram of all points in Trench 1. The horizontal axis represents the grayscale intensity (I), while the vertical axis indicates the histogram values (H). The first maximum corresponds to the ground colors (darker shades), while the second maximum represents the stone colors (brighter shades). Points with intensity values corresponding to the beginning of the second maximum are extracted using Equations (2) to (4).

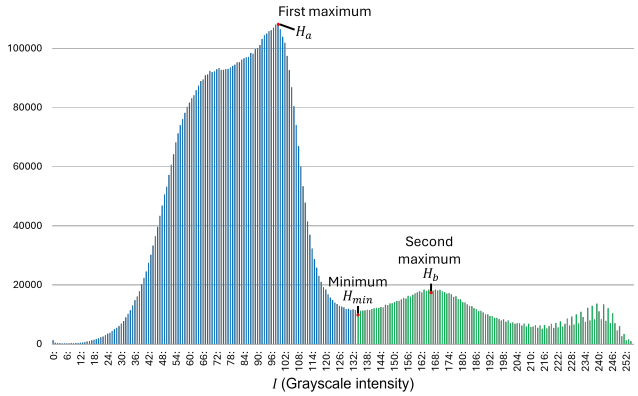


Figure 2: The grayscale intensity histogram.

To segment points based on intensity, maxima and minima in the grayscale histogram are identified first. A *maximum* is defined as a local maximum where the value is greater than its neighboring bins within a defined range:

$$H_e > H_{e-d} \quad \text{and} \quad H_e > H_{e+d}, \quad d \in [1, v] \tag{2}$$

where H_e represents the histogram value at I_e, and d ranges from 1 to v, the maximum allowable range, to check the neighboring bins. Empirically, v is set to 10.

After detecting maxima, a *minimum* is identified as the local minimum between two consecutive maxima:

$$I_{min} = \arg\min_e(H_e), \quad a < e < b \tag{3}$$

where a and b are the index of two adjacent maxima (H_a, H_b), and I_{min} is the intensity of minimum (H_{min}) between them.

Points satisfying the condition in Equation (4) are extracted.

$$I_i \geq I_{min} \tag{4}$$

where I_i represents the intensity of a point. In Figure 2, the histogram values selected for extraction are colored in green.

The result of the color segmentation is shown in Figure 3.

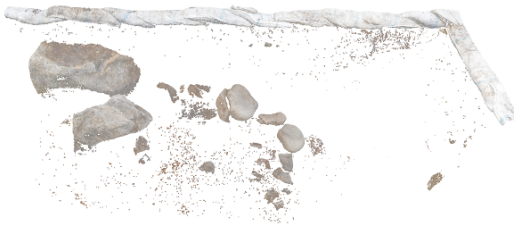


Figure 3: Color segmentation result.

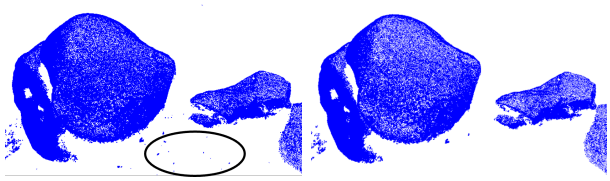
3.1.2 Denoising

As mentioned in Section 2.1, the region-growing algorithm is sensitive to noise, and from color segmentation, some points from the ground are also extracted as noise as shown in Figure 4 (a). To remove these noise points, the *Sparse Outlier Removal* method [23] is applied to determine the noise by evaluating the distance between neighboring points. Points that exceed a specified distance threshold, which is determined based on the average distance and deviation of the point cloud, are considered noise and are removed.

First, the average distance D_i between each point $\mathbf{P}_i$ and its neighboring points is calculated.

$$D_i = \frac{\sum_{j=1}^{k} \|\mathbf{P}_i - \mathbf{P}_j\|}{k} \tag{5}$$

where k represents the number of neighboring points. Then, the average distance of all points



(a) (b)

Figure 4: (a) Part of color segmentation result. (b) Denoising result.

D and the standard deviation σ are calculated as:

$$D = \frac{\sum_{i=1}^{n} D_i}{n} \tag{6}$$

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n} (D_i - D)^2}{n}} \tag{7}$$

Here, n is the total number of points in the point cloud. To identify and remove noise, points whose distance D_i is greater than the threshold $D + \sigma \times \epsilon_a$ is filtered out and ϵ_a is utilized to adjust the threshold based on the standard deviation. Figure 4 (b) shows the result of the denoising.

3.1.3 Region-growing algorithm

The region-growing algorithm segments the stone by starting from a seed point and iteratively detecting neighboring points that share similar characteristics. The segmentation starts with an unvisited seed point, which is assigned a unique region label. To determine whether neighboring points belong to the same region, the algorithm considers two criteria:

1. Distance Threshold: The average distance between the seed point and its nearest neighbors is computed by Equation (5). The average distance of the seed point, D_{seed}, is utilized as the threshold.

2. Surface Normal Angle Threshold: The angle between the normal vectors of the seed point and neighboring points, θ_a^{seed}, is evaluated to ensure smooth surface transitions.

The angle θ_j^i between normal vectors $\mathbf{n}_i$ and $\mathbf{n}_j$ is calculated using:

$$\theta_j^i = \arccos\left(\frac{\mathbf{n}_i \cdot \mathbf{n}_j}{\|\mathbf{n}_i\|\|\mathbf{n}_j\|}\right) \tag{8}$$

Next, for each point $\mathbf{P}_i$, an average angle θ_a^i is computed across its k neighboring points:

$$\theta_a^i = \frac{\sum_{j=1}^{k} \theta_j^i}{k} \tag{9}$$

A point i is added to the region if it satisfies both conditions of Equation (10) and Equation (11).

$$D_i \leq D_{seed} \tag{10}$$

where D_i is the average distance of i-th point.

$$\theta_a^i \leq \theta_a^{seed} \tag{11}$$

where θ_a^i is the average normal vector angle of the i-th point.

Figure 5 (a) shows the results of region-growing with regions in different colors.

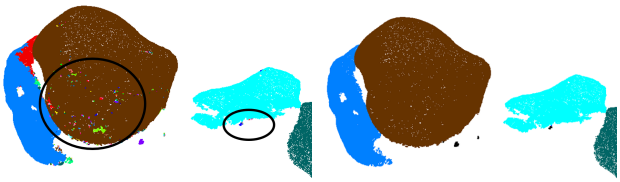
3.1.4 Merging regions

After the region-growing algorithm, some regions may be too small due to variations in the point cloud density or minor noise as shown in black circles in Figure 5 (a). To address this, a post-processing step is applied to merge small regions with their nearest neighboring region.

Regions below a specified threshold size, ϵ_b, are identified and checked for neighboring regions. The neighboring region is selected based on the closest point belonging to a different region, and the entire small region is merged into this target region. If no valid neighboring region is found, the small region is excluded from the segmentation process and marked with a distinct color.

The merging process iterates over the region list until no further small regions remain. Additionally, a second merging pass ensures that any remaining small regions are processed effectively.

After merging, the total number of segmented regions is reduced, resulting in a more coherent segmentation. The updated segmentation results are shown in Figure 5 (b), where the previously fragmented segments have been successfully merged.



(a) (b)

Figure 5: (a) Region-growing result. (b) Merging result.

3.2 Initial alignment

Since the initial alignment of the point clouds heavily affects the matching result of the ICP algorithm [16], the point clouds are aligned through the following steps.

Step 1: Shifting to the Global Origin

The centroids of both the stone and the segmented trench point clouds are calculated. These centroids are then shifted to the global origin $O(0, 0, 0)$.

Step 2: Aligning the Local Axes Using PCA

The local axes for both point clouds are calculated using PCA [24]. The three vectors obtained from PCA are assigned to the coordinate axes based on their lengths: the longest vector is assigned to the x-axis, the second longest to the y-axis, and the shortest to the z-axis. The segmented point set is then transformed to align with the principal axes of the stone accordingly.

For open point clouds like the segmented points from the trench investigating site, where the front and back sides may not be distinguishable, the direction of the z-axis derived from PCA can be opposite to the normal vectors of the points. In such cases, the z-axis is adjusted to align with the normal vectors.

Step 3: Downsampling the Stone Point Cloud

To enhance the efficiency of the ICP algorithm, the stone point cloud is downsampled to match the density of the segmented points of the trench investigating site.

Step 4: Bounding Box Translation

A bounding box around the stone is derived, and the segmented points from the trench are translated to align with the bounding box. The centroid of the segmented points is shifted to align with the centroid of the bounding box's surface, and the local axes of the segmented points are

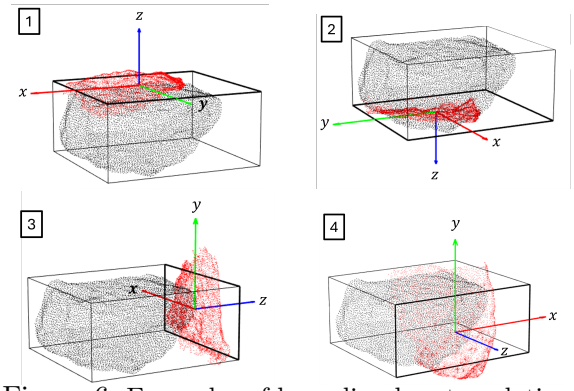


Figure 6: Examples of bounding box translation.

rotated to align with the normal of the bounding box surface. After z-axis aligns with the normal of the bounding box surface, there are 4 possible rotations for the x and y axes including their negations and 6 surfaces, resulting in 24 possible rotations for this alignment.

In Figure 6, the black-colored point clouds represent the stone, while the red-colored point clouds represent the segmented points. The local axes of the segmented points are depicted with the x-axis in red, the y-axis in green, and the z-axis in blue. The target surface for the alignment is highlighted in bold. For instance, Translation 1 shows shifting the segmented points to the top of the bounding box without rotation.

Step 5: Applying the ICP Algorithm

After each bounding box translation, the ICP algorithm is applied to finalize the registration, and the transformation with the lowest ICP score is selected as the matching result. The ICP algorithm performs alignment based on point pairs of the point clouds. Sawada *et al.* [13] introduced the evaluation value E, referred to as the ICP score in this paper:

$$E = D^2 \times (1 - R) \tag{12}$$

where D is the average distance between point pairs and R is the ratio of point pair numbers and point numbers.

4 Experimental results

The implementation and experimentation of this study were performed on a computer equipped with a 12th Gen Intel Core i5-12400 processor,

16.0 GB of RAM, and operating on Windows 11 Pro Education.

The trench and stone point clouds were obtained from the Osato site located in Mishima, Kagoshima, Japan [3]. Trench 1 point cloud, shown in Figure 7 (a), contains 7,716,454 points and includes 10 stones. Trench 2 point cloud in Figure 8 (a), contains 3,596,498 points and has 15 stones. Both point clouds include color information used for the segmentation process. However, they do not include normal vectors.

In denoising, k is set to 30, based on the point density, and ϵ_a is set to 0.05 as the minimum distance for the noise removal. To merge the regions of the region-growing result, ϵ_b is set to 10,000.

Following the steps in Section 3.2, the results of the initial alignment are summarized in Table 1 and Table 2, detailing the point count of the stone and its segmented parts, the number of ICP iterations, the ICP score, and the execution time in seconds.

Table 1 presents results where the stone and segmented point clouds were downsampled to 10,000 points, except for Stone No.671. This downsampling significantly reduced execution time, with all registrations completed in under 10 seconds.

From Table 2, the execution time is dependent on the point count; with fewer points, the execution is faster. Stone No.557, 558, and 559, which have the fewest points, required approximately 3 seconds. In contrast, stones with larger point counts, such as Stone No.560 and 660, required between 50 to 100 seconds.

Based on multiple experimental results, an ICP score of $E \leq 1.5$ is considered a successful registration. For Trench 1, all six stones were successfully registered, verifying the effectiveness of the proposed method. Compared to Trench 2, Trench 1 had a higher point count and density, which contributed to the success of all registrations without failure.

For Trench 2, among the nine stones, seven were successfully registered to the trench. However, Stone No.562 and 563 failed to register. Although Stone No.563 had an ICP score of 1.4, it still failed due to additional factors beyond the ICP score. This issue occurs when the stone has a similar color to the ground (No.563) or when the stone is buried deep in the trench with only a few parts recorded in the trench point cloud (No.562). Nevertheless, as shown in Figure 8 (b) and (c), the registration of the remaining stones was successful.

Figure 7 (b) and Figure 8 (c) show the result of our method, where the registered stones are circled in red. The white line represents the boundary between the soil and the underground. As seen in the figures, the point cloud of the stone visible on the surface is aligned with the point cloud of the stone after excavation.

5 Conclusion

In this paper, we proposed the automated registration of 3D point clouds obtained from archaeological trench investigations. A customized region-growing algorithm utilizing average distance and normal vector angle was used for segmentation. PCA-based transformation and pattern-driven bounding box translation were applied before the ICP algorithm for refined alignment results. Based on the experimental results, the practical use of the proposed method was verified using two types of datasets.

Future works include improving the region-growing algorithm to better segment the stones from the trench, considering multiple-colored or ground-colored stones. Additionally, refining the initial alignment is important to ensure effectiveness for cases like Stone No.562 and 563.

This paper is an updated version of the one presented at NICOGRAPH2024 [25]. We sincerely appreciate the NICOGRAPH 2024 program committee members for their valuable reviews and suggestions on the submitted paper [25]. A part of this work was supported by JSPS KAKENHI Grant Number JP25K00537.

References

[1] Historic England. *Graphical and Plane Table Survey of Archaeological Earthworks.* Historic England, Swindon, Second edition, 2018.

Table 1: The result of ICP algorithm for Trench 1.

Stone No.	Stone point	Segmented trench point	ICP iteration	ICP score, E	Execution time (s)
660	1,000,002	66,228	167	1.3625	9.585
663	500,002	25,801	80	0.9532	2.985
664	400,002	48,801	232	1.1939	8.035
667	399,998	99,490	63	0.6839	3.160
669	400,002	111,811	90	0.7622	3.333
671	250,002	17,010	136	0.7589	16.177

Table 2: The result of ICP algorithm for Trench 2.

Stone No.	Stone point	Segmented trench point	ICP iteration	ICP score, E	Execution time (s)
557	500,002	2,825	121	1.2232	0.71
558	400,002	4,982	118	0.8433	2.223
559	400,002	5,088	201	1.1406	3.202
560	400,002	23,660	315	0.6326	99.907
562	400,002	3,774	132	2.7317	1.218
563	500,000	18,333	403	1.4108	31.973
660	1,000,002	18,000	202	0.9296	53.107
663	500,002	6,420	116	0.987	3.348
664	400,002	6,176	353	1.3824	9.15

(a) (b)

Figure 7: (a) Trench 1 point cloud.
(b) The registration result.

[2] The Geospatial Information Authority of Japan. What is Total Station? `https://www.gsi.go.jp/kidou/TS.html`. Accessed: 2024-8-5.

[3] S. Nakazono, M. Tarora, H. Hirakawa, and T. Toya. An Attempt of "Consistent 3D Excavation" and "Paperless Excavation": Osato Site, Mishima, Kagoshima, Japan. *Papers and Proceedings of the Japan Association for Archaeoinformatics*, 25:1–5, 2022.

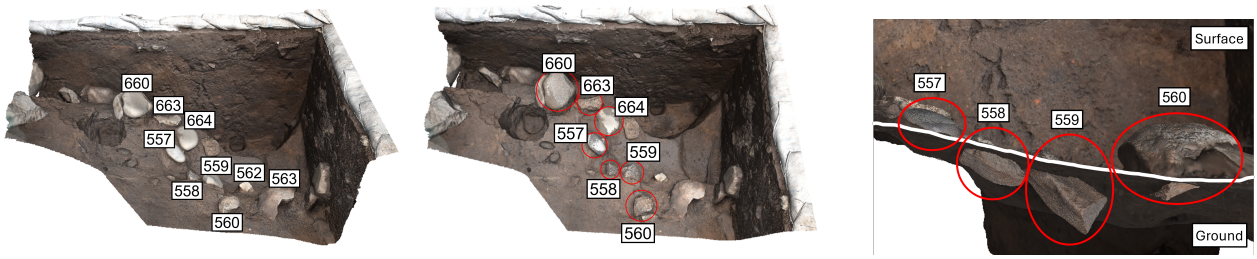
[4] E. Grilli, F. Menna, and F. Remondino. A Review of Point Clouds Segmentation and Classification Algorithms. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 42W3:339–344, 2017.

[5] X. Ruan and B. Liu. Review of 3D Point Cloud Data Segmentation Methods. *International Journal of Advanced Network, Monitoring and Controls*, 5:66–71, 2020.

[6] W. Wang, X.L Ren, and X.Y. Chen. Improved Region Growing Segmentation Algorithm Based on Color 3D Point Cloud. *Foreign Electronic Measurement Technology*, 11:10–14, 2018.

[7] C.L. Kang, F. Wang, M.M. Zong, Y. Cheng, and T.N. Lu. Research on Improved Region Growing Point Cloud Algorithm. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 4210:153–157, 2020.

[8] J. Zeng, D. Wang, and P. Chen. Improved Color Region Growing Point Cloud Segmentation Algorithm based on Octree. In *2022*



(a) Trench 2 point cloud. (b) The registration result. (c) The registration result.

Figure 8: The registration results of Trench 2.

3rd International Conference on Information Science, Parallel and Distributed Systems (ISPDS), pages 424–429, 2022.

[9] W. Liang, P. Xu, L. Guo, H. Bai, Y. Zhou, and F. Chen. A Survey of 3D Object Detection. *Multimedia Tools and Applications*, 80:1–25, 2021.

[10] C.R. Qi, L. Yi, H. Su, and L.J. Guibas. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In *Neural Information Processing Systems*, 2017.

[11] S. Shi, X. Wang, and H. Li. PointRCNN: 3D Object Proposal Generation and Detection From Point Cloud. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–779, 2019.

[12] S. Shi, C. Guo, L. Jiang, Z. Wang, J. Shi, X. Wang, and H. Li. PV-RCNN: Point-Voxel Feature Set Abstraction for 3D Object Detection. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10526–10535, 2020.

[13] Y. Sawada, T. Kinoshita, A. Renchin-Ochir, F. Chiba, and K. Konno. Stone Tool Identification Method Based on Measured Points by RGB-D Camera and Points of Stone Tool Database. *The Journal of The Society for Art and Science*, 21:213–224, 2022.

[14] M. Chang and B. B. Kimia. Measuring 3D Shape Similarity by Matching the Medial Scaffolds. In *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*, pages 1473–1480, 2009.

[15] X. Liu, Z. Han, Y. Liu, and M. Zwicker. Fine-Grained 3D Shape Classification With Hierarchical Part-View Attention. *IEEE Transactions on Image Processing*, 30:1744–1758, 2020.

[16] T. Takahashi, M. You, and K. Konno. A Study on Partial Shape Matching between Flake Surface and Surface of Joining Material using Measured Point Cloud. *The Journal of the Society for Art and Science*, 22:1–10, 2023.

[17] F. Tombari, S. Salti, and L.D. Stefano. Unique Signatures of Histograms for Local Surface Description. In *Computer Vision – ECCV 2010*, pages 356–369, 2010.

[18] R.B. Rusu, N. Blodow, and M. Beetz. Fast Point Feature Histograms (FPFH) for 3D Registration. In *2009 IEEE International Conference on Robotics and Automation*, pages 3212–3217, 2009.

[19] N. Mellado, D. Aiger, and N. Mitra. SUPER 4PCS Fast Global Pointcloud Registration via Smart Indexing. *Eurographics Symposium on Geometry Processing 2014*, 33:205–215, 2014.

[20] M. Furukawa, A. Enkhbayar, and K. Konno. A Study on Hole Filling Algorithm by Synthetic Technique of Three-dimensional Measurement Point Cloud by laser Scanner and Photogrammetry. *Journal of the Japanese Society for Information Archaeology: Information Archaeology*, 26:1–8, 2021.

[21] H.Q. Tian, X.Q. Dang, J.H. Wang, and D.M. Wu. Registration Method for Three-

dimensional Point Cloud in Rough and Fine Registrations based on Principal Component Analysis and Iterative Closest Point Algorithm. *Traitement du Signal*, 34:57–75, 2017.

[22] R. Bala and R. Eschbach. Spatial color-to-grayscale transform preserving chrominance edge information. *Proceedings of the Society for Imaging Science and Technology (IS&T) Twelfth Color Imaging Conference*, pages 82–86, 2004.

[23] R.B. Rusu, Z.C. Marton, N. Blodow, M. Dolha, and M. Beetz. Towards 3D Point Cloud Based Object Maps for Household Environments. *Robotics and Autonomous Systems*, 56:927–941, 2008.

[24] A. Altansukh, M. You, E. Altantsetseg, O. Khorloo, F. Chiba, and K. Konno. A New Matching Algorithm for Stone Tool Reassembly Based on Contour Points of Flake Surface. *The Journal of the Society for Art and Science*, 23:4:1–4:17, 2024.

[25] M. Enkhtugs, M. You, M. Tarora, H. Hirakawa, S. Nakazono, and K. Konno. A Study on Semi-automatic Registration of 3D Point Clouds Obtained from Archaeological Trench Investigations. *NICOGRAPH2024*, pages F–1:1–F–1:8, 2024.

Maral Enkhtugs

is a master course student at the Graduate School of Art and Science, Iwate University. She received a B.S. degree in 2017 from the National University of Mongolia. Her research interests include image processing, automation and visualization.

Mengbo You

is an assistant professor in the Faculty of Science & Engineering, Iwate University. He received the B.S. degree in 2012 from the Computer Science Department, Northwest A&F University, the M.S. degree in 2015 and the Dr.Eng. in 2018, both from Iwate University, Japan. His research interests include machine learning, image processing, object detection and deep learning. He is a member of the Society for Art and Science, and the Institute of Image Electronics Engineers of Japan.

Maki Tarora

is a research assistant at the Environmental Conservation Section of the Museum Science Department at the Curatorial Division of Kyushu National Museum. She has been a research fellow at the International University of Kagoshima since 2019 and joined the Kyushu National Museum as a research assistant in 2025. Her research focuses on 3D documentation and analysis of archaeological materials and sites, with additional interests in archaeological science and digital archaeology.

Hiromi Hirakawa

is a visiting researcher at the Nara National Research Institute for Cultural Properties, Japan. She is also a part-time lecturer at the International University of Kagoshima, where she received her Ph.D. in 2017. Her research interests are “technology and the body” in the ethnographic fields of pottery production and cognitive archaeology, and human cognitive evolution.

Satoru Nakazono

is a professor of archaeology in the Faculty of Intercultural Studies at the International University of Kagoshima, Japan, and received his Ph.D. from Kyushu University in 2002. His research interests include archaeological science, encompassing technological and morphological studies of pottery, as well as XRF analysis and cognitive archaeology. He is also engaged in the exhaustive 3D documentation of archaeological excavations and the practice of public archaeology.

Kouichi Konno

is a professor of Faculty of Science and Engineering at Iwate University. He received a BS in Information Science in 1985 from the University of Tsukuba. He earned his Dr.Eng. in precision machinery engineering from the University of Tokyo in 1996. He joined the solid modeling project at RICOH from 1985 to 1999, and the XVL project at Lattice Technology in 2000. He worked on an associate professor of Faculty of Engineering at Iwate University from 2001 to 2009. He has written a book Introduction to 3D shape processing. His research interests include 3D modeling, 3Dsurface data compression, archaeological relics restoration. He is a member of The Society for Art and Science, The Institute of Image Information and Television Engineers, Japan Association for Archaeoinformatics, Information Processing Society of Japan, and EuroGraphics.