A Study on Junction Region Recognition for Shape Similarity Metric of Stone Tools Using 3D Measured Point Cloud

Altansoyombo Enkhbaatar[1] , Mengbo You[2] , Kouichi Konno[3]

[1]Iwate University, Iwate, Japan, s3321004@iwate-u.ac.jp
[2]Iwate University, Iwate, Japan, ymb@iwate-u.ac.jp
[3]Iwate University, Iwate, Japan, konno21@iwate-u.ac.jp

Corresponding author: Altansoyombo Enkhbaatar, s3321004@iwate-u.ac.jp

Abstract. In the realm of archaeological research, numerous stone tools are unearthed in archaeological excavations providing essential evidence for studying human activity of different periods, cultural backgrounds, and customs. Traditional methods of stone tool analysis often rely on manual inspection, a time-consuming and labor-intensive process. 3D point clouds, which represent surface shapes through data points in 3D space, have become an essential representation in fields such as archaeology, architecture, and art. The fusion of real-world measurements, edge points extraction, and junction regions presents a compelling avenue for improving stone tool identification. Some methods, such as selecting a candidate point of feature lines by its Mahalanobiss distance, and extracting ambiguous ridgelines, hardly extract edge features as intended and can not find junction regions. In other words, accurate evaluation of surface sharpness is a critical challenge in distinguishing junction regions. Previous selecting candidate points of feature lines have limitations due to their complexity, the integration of edge features and junction regions enhances the efficiency of feature extraction and reduces calculation costs. Therefore, this paper presents a method of feature extraction, edge features, and junction region for shape similarity metric of stone tools by 3D measured point cloud.

Keywords: point cloud, feature points, edge points, junction region, stone Tool, flake surfaces
DOI: https://doi.org/10.14733/cadaps.2026.358-369

1 INTRODUCTION

The advancement of 3D scanning technologies has significantly contributed to reverse engineering and digital reconstruction of physical objects. In particular, the analysis of point clouds has become a crucial step in transforming real-world objects into digital models for analysis, visualization, and classification. Reverse engineering typically involves capturing dense point cloud data, segmenting it into meaningful regions, and reconstructing the objects with structural features such as edges, surfaces, and corners. While substantial progress has been made in Computer-Aided Design (CAD) model reconstruction from point clouds[10], much of the existing research focuses on complex free form surfaces generated through surface fitting[10], without incorporating semantic information for topology construction [15]. While modern products are produced with high precision and standard geometric forms, stone tools are handmade, irregular, and lack consistent surface patterns. Therefore, industrial CAD reconstruction methods cannot be directly applied to stone tools.

Unfortunately, the organic matter underground is decomposed because of Japan's hot and humid weather and high soil acidity. Therefore, numerous stone tools instead of plants and wood, are unearthed in archaeological excavations, providing essential evidence for studying human activity. Stone tools, considered cutting tools or weapons, were crafted by hammering and polishing stones, representing significant artifacts of prehistoric societies. Archaeologists gain insights into the characteristics of different periods, cultural backgrounds, and customs by analyzing these buried cultural assets. 3D point clouds, which represent surface shape through data points in 3D space, have become an essential representation in fields such as archaeology[13], architecture[1], and art[9].

Point clouds have gained significant popularity as a method for representing 3D objects, one field that has greatly benefited from the utilization of point clouds is cultural heritage research, where they play a vital role in preserving valuable artifacts and historical sites for future generations. Shurentsetseg et al.[8] proposed a method for automatically extracting feature lines from stone tools represented by a point cloud, aiming to reduce the time and expertise required for scale drawing. The method uses the Mahalanobis distance metric to extract points on the outline of stone tools from a point cloud. Surface variation is then calculated using different numbers of neighbors to detect potential feature points based on their surface characteristics. The initial set of potential feature points undergoes refinement through thinning, utilizing Laplacian smoothing to arrive at a final selection. Feature lines are obtained by linking these points according to the Mahalanobis distance field. Their method[8] extracts only edge shapes, which is insufficient information for shape matching. Edge relations such as junctions may be required for accurate shape matching.

Edge points represent the points on the boundary edges bounded by the flake surface of the stone tool. Detecting junction regions, defined as regions where edges share a common vertex, continues to be a challenging aspect of 3D point cloud analysis. Junction regions are crucial for capturing the fundamental structure of a stone tool, however they are frequently hard to extract due to issues like noise and ambiguous shapes. If junction regions are not accurately recognized, the resulting features may fail to reflect the geometry of the stone tool, leading to incorrect shape similarity assessments. In previous work[2], edge points and junction regions are extracted from 3D point cloud.

Therefore, this paper presents a novel method to recognize junction regions, which are essential for capturing the distinctive characteristics of stone tools. Utilizing feature line extraction[8] analysis and edge detection methods to pinpoint these junction regions with high accuracy. To evaluate the effectiveness of our technique, the extracted features are compared with peakit[7] images of the stone tools.

2 RELATED WORKS

Feature extraction techniques can generally be divided into two main types: point-based methods and mesh-based methods[4]. Our current method adopts a point-based approach, with a particular focus on detecting junction regions where multiple sharp edges converge within the point cloud. While significant advancements have been made in curvature approximation methods over the past two decades, accurately identifying and extracting junction regions remains a challenging aspect of feature extraction, especially on sharp edges and complex geometric structures.

Sawada et al.[11] introduced a stone tool matching method using contour similarity HuMoment algorithm[12] and ICP algorithm for better accuracy. While effective for both thin and thick stone tools, its precision decreases significantly for thicker stones, underscoring the need for more sophisticated methods or additional information sources to enhance recognition precision for thicker stone tools.

Amgalan et al.[5] developed a technique for extracting flake surfaces for the assembly of stones for making joining material, utilizing unorganized point clouds obtained from a 3D laser scanner. The curvature of each point in the point clouds is approximated with a local surface fitting algorithm, which helps in identifying potential feature points. Feature lines are extracted through a directionally growing algorithm, making it easier to detect feature lines from unorganized point data. This method[5] focuses on extracting feature lines using the principal curvatures and principal directions of potential feature points along the axis directions, which are then merged to form complete feature lines. In contrast, semantic segmentation classifies each point into meaningful categories using structural information. For example, it can easily distinguish parts of tables and chairs because of their regular shapes and standard components. However, applying semantic segmentation to the stone tools is hard, as their handmade and irregular geometry lacks consistent features or repeating patterns, making it difficult for specific shape characteristics to appear reliably across different stone tools. To ensure the accuracy of the illustration, the detected features are modified based on specific knowledge of illustrating stone tools. The approach takes advantage of the computational efficiency of data-parallel computation on the GPU for faster processing.

The major steps of the reverse engineering process, contain steps: 1) capturing and pre-processing of original data; 2) segmentation of the point clouds; 3) regions of interest recognition in the segmentation; 4) classification of the regions identified in the segmentation step; 5) generation of analytical surfaces and features; 6) finishing operations and CAD model reconstruction. In the stone tool point cloud, we use steps 1-4, but it is hard to use analytical features and CAD model reconstruction because the flake surface is difficult to represent using analytical surfaces such as spheres, cylinders, cones, or elliptic surfaces, and the boundaries of the flake surface are often ambiguous..

Z. Yu et al.[14] proposed a method for extracting and visualizing structural information from point clouds using immersive interfaces. They introduced a method to improve human perception of the foundational structures and interactive tools for features such as faces, edges, vertices, and connectivity information from unorganized point clouds. The method[14] focused on extracting faces from unorganized point clouds and extraction of edge points based on which face they are adjacent to. Their method extracting vertices is not suitable in stone tool, because of many small surfaces that give noise. In our method, prepare a point cloud with resampling to reduce noise before the region-growing algorithm, which improves the accuracy of edge point extraction.

L. Yujia et. al.[10] In reverse engineering and CAD model reconstruction, various studies have tackled the challenge of extracting structural information from point clouds. Traditional methods segment and fit basic geometric primitives to reconstruct boundary representations (B-reps), but struggle with complex free-form surfaces. Recent advancements, such as Point2CAD[10], have introduced learning-based segmentation combined with geometric surface fitting to recover complete CAD models, including edges, junctions, and freeform surfaces. Their approaches are primarily optimized for industrial CAD applications with an F-score[10] of surface 0.947, edges 0.816, and corners 0.736. However, the irregular surfaces of stone tools are different from industrial objects, which makes it difficult to apply their methods directly.

F. Buonamici et al.[6] present a comprehensive overview of reverse engineering methodologies and tools, with a focus on the reconstruction of CAD models from measured data. Categorize reverse engineering approaches into three main types: surface-based modeling the direct reconstruction of surfaces from point cloud data using methods such as NURBS (Non-Uniform Rational B-Splines), feature-based modeling the identification and modeling of geometric features (e.g., planes, cylinders) from the scanned data, facilitating the creation of parametric CAD models, hybrid modeling integrates surface and feature-based techniques, leveraging the strengths of both methodologies. Evaluates various reverse engineering software systems, highlighting their capabilities and limitations in handling complex geometries and integrating with CAD platforms.

Z. Zhang et. al.[15] address the challenges in reverse engineering CAD models from original geometric data.

Conventional methodologies frequently require considerable manual intervention, which limits the efficiency of the reconstruction process. To reduce these constraints, propose a comprehensive CAD reconstruction pipeline that guarantees high-precision reconstruction while markedly diminishing the requirement for manual operations. Notable innovations within their framework include the automatic determination of extrusion heights the algorithm that autonomously computes the height of extruded features, thereby reducing errors during the construction of the model, efficient primitive loop fitting a method to rapidly fit the cutting lines of models to primitive loops, enhancing the accuracy and speed of the reconstruction process.

Given these limitations in existing works, our study proposes a junction region recognition method that enhances the structural analysis of stone tools by identifying and utilizing junction points to improve shape similarity metrics. Unlike previous approaches, our method provides a more accurate representation of edge connectivity, ensuring reliable matching and classification of stone tools based on their geometric features.

The methods [14], [11], and [5] all use feature extraction to work with and extract faces, edges, and contour points. Unlike previous shape similarity methods, which are slower, leveraging the number of junction regions can improve the matching process's efficiency and accuracy. Our novel approach using junction regions for stone tool identification has never been used before.

3 PROPOSED METHOD

3.1 Overview

Junction areas are points in a point cloud that indicate the intersections where three or more segmented regions or edge point pairs converge. Edge points represent the convex shape of an object, and refer to points that lie on the edge of a flake surface. These points are an essential representation for junction region extraction, which represents areas where several boundaries come together, and for stone tool identification.

Our proposed method identifies edge points and junction regions. Concretely, our method involves several key steps: 1) re-sampling a point cloud by the VoxelGrid filter for calculating cost, 2) applying a region-growing algorithm, 3) extracting edge features, and 4) extracting junction regions, 5) constructing topological structure. Our method assumes that a point cloud of the stone tool is separated front and back sides.

Before extracting edge features, we need to prepare the point cloud using a re-sample and simplify using the VoxelGrid filter to speed up computation, maintaining the overall shape. Then, the back side of the stone tool is removed based on the point cloud z-axis to reduce noise for the next step.

A region-growing algorithm of the method[5] works with full stone tool point cloud and makes segmentation by the looped ridge lines. The algorithm has been modified to function with half stone and recognize the stone tool's outline as a limit of the point cloud, creating a loop with ridge lines. A region-growing algorithm is used to identify and segment the smooth areas in the point cloud that represent the flake surface. However, due to the variation in the normal vector angle around the ridgeline, the boundary of the segmented region is not accurately aligned with the ridgeline. To ensure the accuracy of the ridgeline extraction, it is necessary to identify points along the boundary that are close to the ridgeline. To extract an accurate ridgeline location, more new steps of region-growing algorithm are introduced to find the edge points which are the closest points to the ridgeline.

Fig.1(a) shows a stone tool, and Fig.1(b) shows the stone tool point cloud located on the coordinate system x, y, z. Our method assumes that the half stone tool point cloud is located on a flat surface indicated by the x, y plane as shown in Fig.1(b). The different flake surface RGB colors show the result of our region-growing algorithm.

3.2 Data Structure of the Points in Point Cloud

In this section, data structure for efficient extraction of junction regions are explained. The data structure of the point $\mathbf{p}_i$ is as shown in Fig.2:

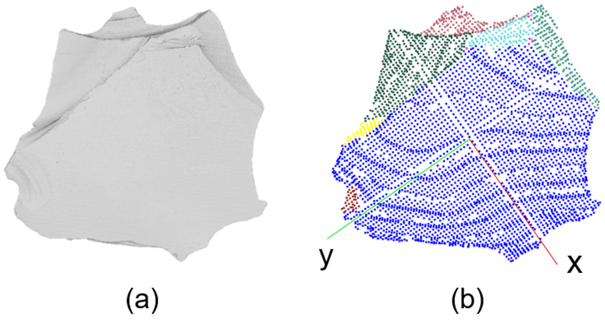

Figure 1: Stone tool point cloud in the coordinate system.

1. x, y, z,
2. R, G, B, - color of the region,
3. $\mathbf{n}_x$, $\mathbf{n}_y$, $\mathbf{n}_z$ - normal vector information,
4. 0/1 - inside point/Boundary point,
5. 0/1 - not Edge point/Edge point,
6. EPP_id: if it is Edge point, add Edge point pair id
7. 0/1 - not junction point/Junction point,
8. JP_ids: if it is Junction point, add Junction point ids of Junction area.

Point set	
p_i	(x, y, z), (RGB), (n_x, n_y, n_z), InP/BP, notEP/EP, EPP_id, notJP/JP, JP_ids

Figure 2: Data structure of point set.

The data structure of the region R_i is as shown in Fig.3:

1. R, G, B, - color of the region,
2. Boundary_Point_List - boundary points of the region,
3. Edge_Point_Pair_List - edge point pair with nearest region.

Region set	
R_i	(R, G, B), Boundary Point List, Edge Point Pair List

Figure 3: Data structure of regions.

In Fig.4, the region list is denoted. Before the region is generated, edge point pair list is generated by the method described in Sec.3.3. When the region denoted by by the method described in Sec.3.4 is detected, it is added to the region list. The region list is the base for constructing topological information using the boundary point list and edge point pair list.

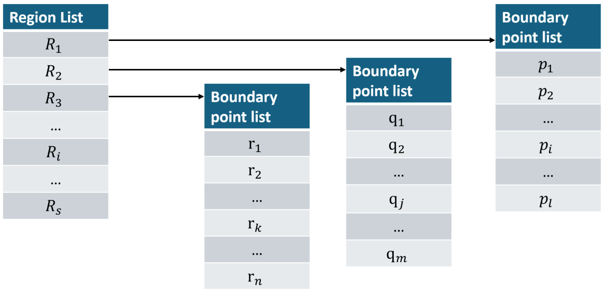

Figure 4: Data structure of region set.

3.3 Extract Edge Features

To extract edge points: 1) use a region growing algorithm with a correction process of region boundaries[5] and extract edge points by choosing points where the angle difference between the normal vectors of adjacent points exceeds a specified threshold, defined as the average smooth angle difference within the regions, as described in method[2]. 2) arrange boundary points of each region in the counterclockwise direction. Fig.5(a) shows the boundary points of R_1 and R_2 represented by black color as an example.

After the boundary points are extracted, edge points osculated two regions are determined. For example, the red points shown in Fig.5(b) represent edge points between R_1 and R_2. In point cloud representation, edge points and corresponding edge point pairs are used for shape representation.

Pairs of edge points are used for extracting junction region. Edge point pair is determined as following steps:

1. Use region-growing algorithm to extract regions.
2. Extract boundary points for each region and make a boundary point list into the region structure. In addition, store boundary points flag into a point set structure. For example all boundary points of R_1 are shown in Fig.5(a)
3. Traverse boundary point list of R_1 and find edge point for each boundary point. Candidate of edge point is boundary point lists without R_1's list. For example, edge points of R_1, denoted $\mathbf{p}_i$, and edge points of R_2, denoted $\mathbf{q}_j$ is shown in Fig.5(c).
4. Extract edge points of R_1 and R_2 and make edge point list for intermediate information.
5. Calculate the average distance between $\mathbf{p}_i$ and $\mathbf{q}_j$, derived by step 3 denoted by ϵ.
6. If the distance is less than the average distance ϵ, make a point pair $(\mathbf{p}_i, \mathbf{q}_j)$ as shown in Fig.5(b).
7. After making point pairs $(\mathbf{p}_i, \mathbf{q}_j)$ add them to the edge point pair list as shown in Fig.3

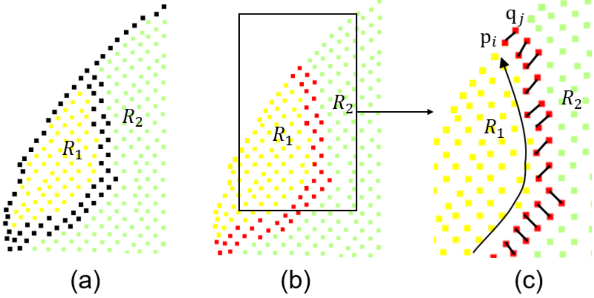

Figure 5: Red points are edge points between R_1 and R_2.

3.4 Junction Region Recognition

Junction regions play a crucial role in reconstructing the geometry of objects from point clouds. Their accurate classification helps create a more detailed and accurate representation of the stone tool topology. The junction region and edge points provide a structure of the flake surface connection of the stone tool. No vertex is used because the junction region is an area and is not a point position.

Junction regions are identified within the point cloud, representing areas where three or more edge point segments meet. Since all flake surfaces have convex shapes and no concave shapes appear, these junctions naturally form at the intersections of outward-facing surface boundaries. These regions are crucial for understanding the overall structure and shape of the stone tool. In stone tool analysis, junction regions provide valuable information about the connectivity and relationships between flake surfaces, contributing to shape similarity metrics.

How to find the junction is as following steps:

1. Traverse edge point pair list that belongs to region R_i as shown in Fig.3. For example, the edge point pair list of R_1 is traversed as shown in Fig.6.

2. Find the edge point pair where the region of the pair is changed. For example, $(\mathbf{p}_3, \mathbf{q}_4)$ to $(\mathbf{p}_4, \mathbf{r}_3)$ as shown in Fig.6 is a place of changing the region pair. If the changing pair is detected, the edge point pairs are added to the edge point pair of junction list.

3. Get junction region by ordering junction points in a clockwise direction, such as $(\mathbf{p}_i, \mathbf{p}_{i+1})$, $(\mathbf{p}_{i+1}, \mathbf{r}_k)$, $(\mathbf{r}_k, \mathbf{r}_{k+1})$, $(\mathbf{r}_{k+1}, \mathbf{q}_j)$, $(\mathbf{q}_j, \mathbf{q}_{j+1})$, $(\mathbf{q}_{j+1}, \mathbf{p}_i)$. For example, Fig.6 shows the junction region which start point is p3, represented by $(\mathbf{p}_3, \mathbf{p}_4)$, $(\mathbf{p}_4, \mathbf{r}_3)$, $(\mathbf{r}_3, \mathbf{r}_4)$, $(\mathbf{r}_4, \mathbf{q}_3)$, $(\mathbf{q}_3, \mathbf{q}_4)$, $(\mathbf{q}_4, \mathbf{p}_3)$

4. Make all junction list to traverse all edge point pair list in R_i.

Junction regions are key markers, providing valuable information for precisely characterising the object's surface features. After finding the junction region, repeat the process for every region edge point pair to find all junction regions.

3.5 Topology Map Using Junction Regions

A topology map represents the structural connectivity of an object by identifying and linking key feature points.

The steps of extracting junction region order for each region are as following steps:

1. Traverse the junction region list which is generated by the method described in Sec.3.4, apply the id number to each junction region. In Fig.7(a), for example, $\mathbf{n}_1$ to $\mathbf{n}_4$ are the id number of junction regions.

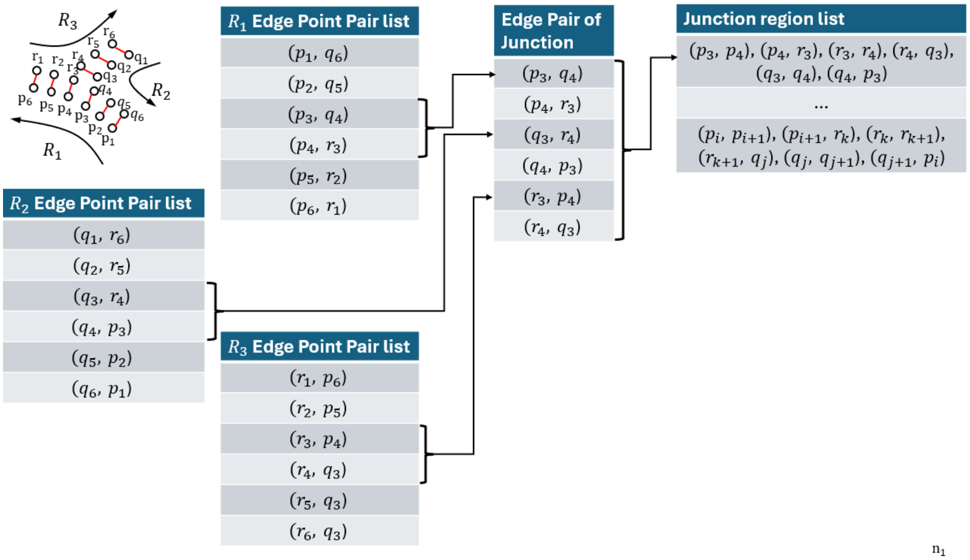

Figure 6: Data structure of junction region list and example how to extract junction region.

2. Using edge point pair list, find the adjacent region next to the junction regions as shown in Fig.7(b)
3. By traversing edge points of each region starting from R_1 in counterclockwise direction, make junction region connections $((\mathbf{n}_1, \mathbf{n}_2), (\mathbf{n}_2, \mathbf{n}_3), (\mathbf{n}_3, \mathbf{n}_1))$, $((\mathbf{n}_4, \mathbf{n}_2), (\mathbf{n}_2, \mathbf{n}_1), (\mathbf{n}_1, \mathbf{n}_4))$, $((\mathbf{n}_3, \mathbf{n}_2), (\mathbf{n}_2, \mathbf{n}_4), (\mathbf{n}_4, \mathbf{n}_2))$ as shown in Fig.7(c).

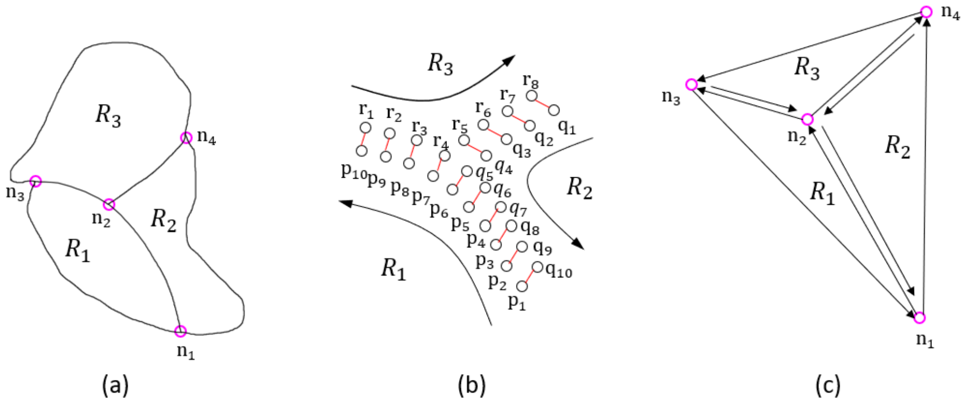

Figure 7: Stone topology map structure.

For stone tools analyzed through 3D measured point clouds, junction regions serve as critical markers that define the connections between surfaces. The following steps outline the process of constructing a topology map using junction regions.

How to construct topology map:

1. Visualize the junction regions connections using a graph representation.

2. Nodes represent junction regions $\mathbf{n}_i$ and regions R_i, while edge points represent adjacency and shared boundaries as shown in Fig.8(b).

3. The resulting topology map provides a clear structure of how regions and junctions are connected as shown in Fig.8(c).

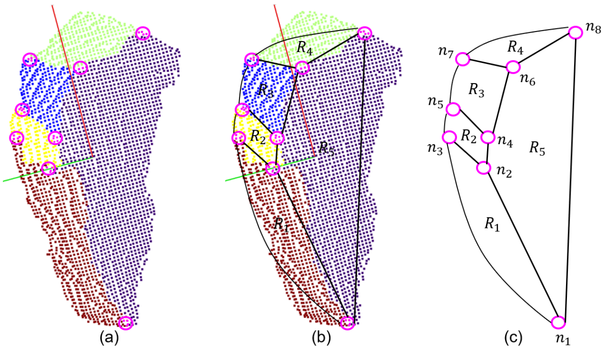

Figure 8: Topology map.

4 RESULTS

The proposed method includes junction region and edge point extraction. The proposed method was implemented in C++ programming language used for the recognition and visualization of the stone tools with Visual Studio 2019 and Windows 10 Pro Education. The experiment was performed on an Intel Core i5-11400 2.60 GHz machine with 16 GB of RAM.

The 3D scan data utilized in this study were acquired through a 3D surface reconstruction technique employing a four-directional measurement machine [3].

The proposed method was implemented on all 3D stone tool flakes with id numbers 39-46, resulting in the extraction of their junction points. Following this, top and bottom surfaces of the flakes with id numbers 39-46 were extracted and the results of the edge points and junction regions. The results were verified by comparing them with peakit[7] image of the stone tools.

As shown in Fig.9, Fig.9(a) is the peakit[7] image of the stone tool, and Fig.9 (b) is a point cloud of the stone tool with extracted junction regions. Fig.9(b) shows a more detailed result with the same quantity of junction region as Fig.9(a). For every stone tool side in the database, there is an optimal number of points from 5000-15000 in a point cloud. Simplifying the point cloud into a regular point cloud improves the recognition of flake surfaces and produces more accurate results.

Table 1 shows the result of the number of junction regions in peakit[7] image and stone tool point clouds. Also, the number of points in the stone tool point cloud and execution time. By comparing number of junction regions of all stone tools with peakit[7] image from 27 junction regions, our method found 29 junction regions. We have observed the position of the junction regions. Thus, 27 junction regions are the same as the peakit[7] image. However, the two junction regions are different. As the result, 93% of the junction regions are extracted. Therefore, our method is useful for extracting junction regions.

5 CONCLUSIONS

This paper proposed a method for extracting edge features and junction regions from a 3D point cloud of a stone tool and constructing its topological structure. Our method introduces a novel use of junction regions,

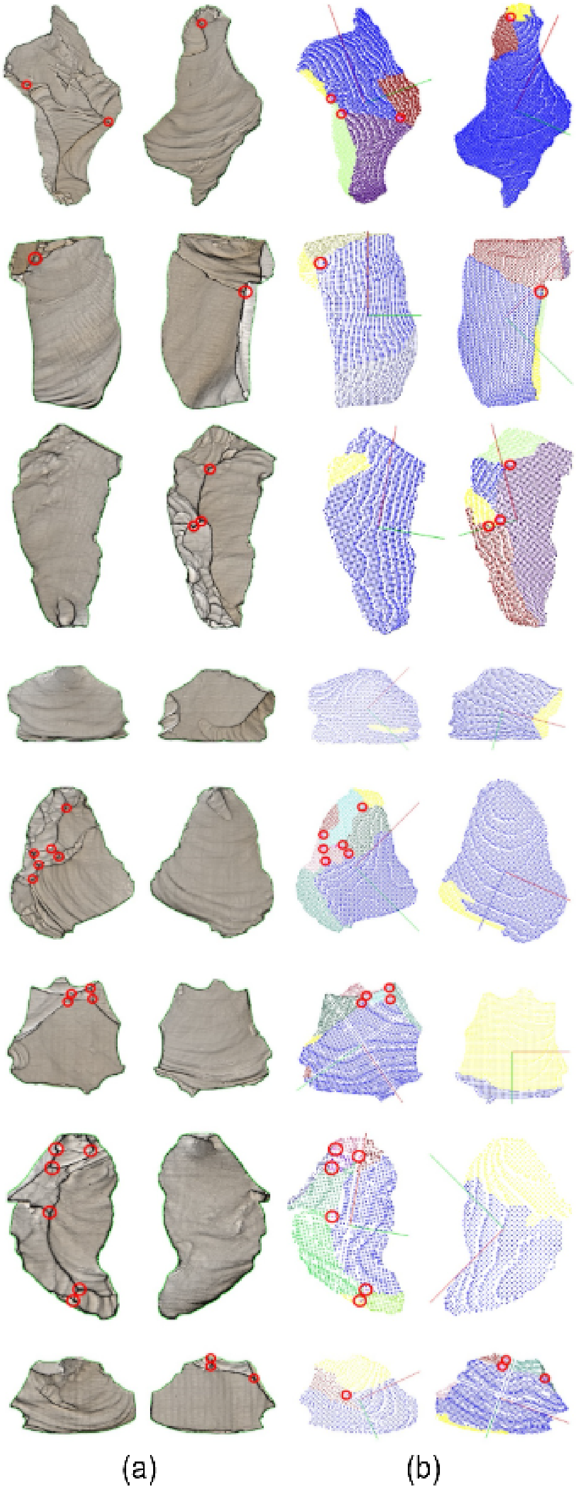

Figure 9: Stone tools from Group 3 with extracted junction regions.

Stone Id	peakit[7]	Junction regions number	Number of points	Execution time
39	3	4	11899	0.35s
40	2	2	7426	0.25s
41	3	3	6097	0.23s
42	0	0	5941	0.19s
43	6	6	7490	0.26s
44	4	4	5386	0.15s
45	6	6	2676	0.09s
46	3	4	3051	0.10s

Table 1: Results of extracted junction regions comparing with those in peakit[7] images.

derived from edge point sequences, to form a topological structural representation of the stone tool. We verify the result of edge points and junction regions by observation. In future work, we aim to extend this method to combine the front and back parts of a point cloud and make a topological structure of the full stone tool. Extracted edge points and junction areas can be used to identify matching joint surfaces between fragments, supporting automated or semi-automated reassembly of stone tools. Furthermore, we plan to apply this method for the reconstruction of stone tool fragments, joining them together to form a complete topological structure of the mother rock.

ACKNOWLEDGEMENTS

A part of this work was supported by JSPS Grant Number JP22K00998.

REFERENCES

[1] Abreu, N.; Pinto, A.; Matos, A.; Pires, M.: Procedural point cloud modelling in scan-to-bim and scan-vs-bim applications: A review. ISPRS International Journal of Geo-Information, 12(7), 2023. ISSN 2220-9964. http://doi.org/10.3390/ijgi12070260.

[2] Altansoyombo, E.; Mengbo, Y.; Kouichi, K.: Edge feature extraction for shape similarity metric of stone tool by 3d measured point cloud. In Nicograph International 2024, vol. 1, 32–35, 2024. http://doi.org/10.1109/NICOInt62634.2024.00015.

[3] Altantsetseg, E.; Muraki, Y.; Chiba, F.; Konno, K.: 3d surface reconstruction of stone tools by using four-directional measurement machine. International Journal of Virtual Reality, 10, 37–43, 2011. http://doi.org/10.20870/IJVR.2011.10.1.2800.

[4] Altantsetseg, E.; Muraki, Y.; Matsuyama, K.; Chiba, F.; Konno, K.: Feature extraction and modification for illustrating 3d stone tools from unorganized point clouds. The Journal of the Society for Art and Science, 12(1), 36–47, 2013. http://doi.org/10.3756/artsci.12.36.

[5] Amgalan, A.; You, M.; Konno, K.: A study on automatic flake surface segmentation of stone tools by calculating shape features. In International Workshop on Advanced Imaging Technology (IWAIT) 2023, vol. 12592 of Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, 2023. http://doi.org/10.1117/12.2666684.

[6] Buonamici, F.; Carfagni, M.; Furferi, R.; Governi, L.; Lapini, A.; Volpe, Y.: Reverse engineering modeling methods and tools: a survey. Computer-Aided Design and Applications, 15, 1–22, 2017. http://doi.org/10.1080/16864360.2017.1397894.

[7] Chiba, F.; Yokokoyama, S.; Kaneda, A.; Konno, K.: Development of network-type archaeological investigation system. ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XL-5/W7, 99–103, 2015. http://doi.org/10.5194/isprsarchives-XL-5-W7-99-2015.

[8] Erdenebayar, S.; Konno, K.: Feature line extraction of stone tool based on mahalanobis distance metric. The Journal of the Society for Art and Science, 18(1), 51–62, 2019. http://doi.org/10.3756/artsci.18.51.

[9] Ivsic, L.; Rajcic, N.; McCormack, J.; Dziekan, V.: The art of point clouds: 3d lidar scanning and photogrammetry in science & art. In ACM Association for Computing Machinery, vol. 1, 404–411, 2021. http://doi.org/10.1145/3452853.3452855.

[10] Liu, Y.; Obukhov, A.; Wegner, J.; Schindler, K.: Point2cad: Reverse engineering cad models from 3d point clouds. In Conference on Computer Vision and Pattern Recognition (CVPR). Seattle, WA, USA, 2024. http://doi.org/10.1109/CVPR52733.2024.00361.

[11] Sawada, Y.; Kinoshita, T.; Renchin-Ochir, A.; Chiba, F.; Konno, K.: 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(4), 213–224, 2022. http://doi.org/10.3756/artsci.21.213.

[12] Seto, A.; Seto, M.; Kinoshita, T.; Chiba, F.; Konno, K.: A method to improve the accuracy of stone tools identification with d2 distribution. In International Workshop on Advanced Imaging Technology (IWAIT) 2024, vol. 13164, 110, 2024. http://doi.org/10.1117/12.3019564.

[13] Yang, S.; Hou, M.; Li, S.: Three-dimensional point cloud semantic segmentation for cultural heritage: A comprehensive review. Remote Sensing, 15(3), 548, 2023. http://doi.org/10.3390/rs15030548.

[14] Yu, Z.; Lin, T.; Gumhold, S.; McGinity, M.: An immersive method for extracting structural information from unorganized point clouds. In Nicograph International 2024, vol. 1, 24–31, 2024. http://doi.org/10.1109/NICOInt62634.2024.00014.

[15] Zhang, Z.; Zhao, M.; Shen, Z.; wang, Y.; Jia, X.; Yan, D.: Interactive reverse engineering of cad models. Computer Aided Geometric Design, 111, 2024. http://doi.org/10.1016/j.cagd.2024.102339.