The Journal of the Society for Art and Science, Vol. 23, No. 2, pp. 4:1 – 4:17 (2024)

A New Matching Algorithm for Stone Tool Reassembly Based on Contour Points of Flake Surface

Amgalan Altansukh[1)] **Mengbo You**[1)] **Enkhbayar Altantsetseg** [2)]
Oyundolgor Khorloo [2)] **Fumito Chiba** [3)] **Kouichi Konno** [1)]

1)Graduate School of Science and Engineering, Iwate University
2) National University of Mongolia
3) LANG CO.,LTD.

amgaa0809(at)gmail.com, {ymb, konno21}(at)iwate-u.ac.jp

Abstract
Reassembling fragmented stone tools from the Paleolithic era remains a significant archaeological challenge. Our study addresses this challenge by introducing a matching algorithm with a primary emphasis on enhancing computational efficiency and refining the flake surface matching process. Our method builds upon previous study, specifically targeting reductions in the computation time. The algorithm was tested on a dataset comprising 43 stone models. A critical aspect of this study is flake surface matching which is a fundamental aspect of stone tool reassembly. By optimizing the computational cost, we aim to provide archaeologists with a more efficient and accurate tool for reconstructing archaeological artifacts.

Keywords: Flake surface, Matching algorithm, Point cloud, Reassembly, Stone tool restoration

1 Introduction

The Paleolithic era witnessed the production of a wide range of stone tools, including cutting implements and weapons. Several archaeological sites have been excavated and analyzed. These sites often contain remnants of debris generated during the production and utilization of stone tools. Considering their durability compared to organic materials, such as bones, antlers, and wood, stone artifacts provide significant evidence of the locations and periods of early human activities because of their geographical distribution, and adaptive capacities in various environments [1].

Stone tools offer valuable insights for early human toolmakers, including their production techniques, life patterns, and transfer of things [2]. For instance, archaeologists analyze the assembly order for investigating creation of stone tools. Figure 1 shows an example of excavated stones, which were restored. The restored mother rock is referred to as the "Joining Material." Joining materials have played a significant role in the analysis of human activities during this period. Moreover, these joining materials have a significant educational value and can be displayed in historical museums and educational institutions [1].

However, the completion and analysis of relics [1] is challenging. Even though relics are subject to measurement through scanning methods [3] and several issues can be solved swiftly and accurately by computational algorithms, the task of reassembling stone tools remains intricate and demanding.

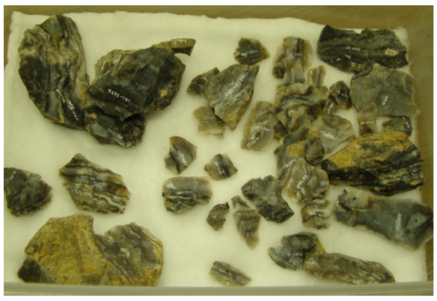


Figure 1: Example of excavated stones.

Archaeologists manually reassembled stone tools using traditional restoration techniques, as shown in Figure 2(a). However, this approach has two major limitations. First, the assembly process is time-consuming because it requires the identification of correctly matched surfaces by trial and error. Second, the assembly and disassembly of stone tools require special archaeological knowledge. These are challenging tasks, because not all stone tools constructed from the mother rock can be excavated.

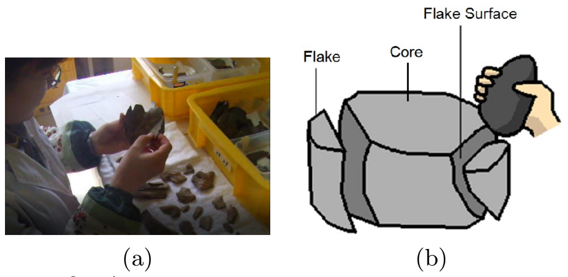


Figure 2: a) Traditional reassembly.
b) Making stone tools.

In recent years, several reassembly approaches have been developed for the reconstruction of fragmented archaeological artifacts such as pottery and fresco wall paintings. However, the successful application of these methods to reassemble stone tools is limited for owing to the following reasons: (1) Irregular shapes of stone fragments: stone tools exhibit highly irregular shapes, making it challenging for descriptor-based methods to extract crucial features. The lack of distinct regional features on flake surfaces further complicates the feature extraction process. (2) Global matching algorithms have proven inadequate, considering the requirement for partial matching between pairs of flake surfaces. The previous study [16] introduced a method of stone matching. [16] extracts the boundary of the flake surface to determine the initial position. However, [16] the excessive computation time and difficulties with partial matching.

Therefore, our study introduces a new flake surface matching algorithm designed to significantly improve the computational time while accurately accommodating partial matches. This study aims to address these issues. The proposed method involves aligning two flake surfaces based on the maximum number of matched contour (edge) points. By introducing a tunable matching parameter, a user can control the balance between the computation time and accuracy, thus addressing a critical bottleneck in stone tool reassembly.

The contribution of this study is a new fine-tuned approach to flake surface matching that utilizes the contour points of flake surfaces. This approach not only enhances the efficiency of the matching process but also proves effective in handling the unique challenges posed by partial matching scenarios. We confirmed that our method was effective for reassembling stone tools.

2 Related Work

2.1 Stone Tools

Numerous stone tools have been manufactured prehistorically through rock striking process. Crafting a stone tool involves the repetitive action of striking the edge of a rock with a pebble, resulting in the peeling of flakes to modify and refine the tool shape, as shown in Figure 2(b). Within this archaeological context, an assembly of stone tools, known as joining materials, represents a reconstruction in which the core and flakes originate from the same rock. In addition, the joining materials are dispersed across the excavation site. Restoration involves reassembling these fragments to recreate the original composition and structure of a stone tool [1].

2.2 Reassembly Methods

Extensive studies have been conducted on the reassembly fractured objects. Huang et al. [4] introduced a feature-based alignment method that effectively addressed these scenarios and achieved remarkable results. However, their method is complex and, comprises various specialized algorithms for tasks such as segmentation, multi-scale feature extraction, correspondence determination, registration, collision detection, and supervised learning. Consequently, implementation and adoption of these methods pose significant challenges. Brown et al. [5] proposed a method specifically designed to reassemble fragments of wall paintings. Willis et al. [6] presented a system tailored to pottery shard reassembly.

In [7], an innovative method was proposed for reassembling axially symmetric ceramic pots from three-dimensional (3D) scanned fragments. Their method addressed specific challenges with ceramic artifacts, particularly axial symmetry. Their approach is noteworthy because it incrementally adds fragments using a beam search technique. This significantly mitigates the false positive matches. Their method not only improves the match accuracy through geometric descriptors but also facilitates the simultaneous reassembly of multiple pots from mixed collections. However, the application of their technique to the reassembly of stone tools is challenging. Reassembly stone tools lack the axial symmetry present in ceramic pots and require consideration of the flake removal sequence from the core stone.

In [8], a new methodology for generating synthetic 3D fragmented data was proposed to facilitate the evaluation of object restoration, particularly in the context of cultural heritage. Their approach was designed to overcome the challenges of limited availability of real test data. Their method produces artificial fracturing from an input object without physical simulation. In addition, their method uses a “cutter object” to create new breaking edges and thereafter fragments the object. It generates a large-scale fragment test dataset from existing cultural heritage models. These datasets have advantage of ground truth (the input object before fracturing), which is often missed. However, it does not directly relate to the reassembly stone tool task or reassembly task. It focuses on the reverse process of reassembling models.

In [9], a mesh-based approach to create restorations integrated with broken objects was introduced. Utilizing the 3D scanned meshes of both broken objects and their intact counterparts, their method generates a restoration piece by smooth. It focuses on objects for which the missing parts are predictable, and its goal is to restore the original form of the objects. However, its application to the reassembly of stone tools poses several challenges. First, it relies on the predictability of the shape of the missing part. This condition is not satisfied in stone tool reassembly in which the shape and size of the flakes are unpredictable. Second, the methodology is mesh-based, whereas stone data were represented in a point cloud. Finally, their proposed method does not address partial matching or the sequential order of assembly, both of which are crucial for stone tool reassembly tasks.

The "Fantastic Breaks" dataset [10] was designed to facilitate machine learning study in automated reassembly by providing a comprehensive collection of 3D scans of real-world broken objects and their counterparts. It focuses on a complete object reconstruction using predefined models. The use of predefined models may not align with the stone tool reassembly tasks. This often requires addressing the unpredictable fragment shapes and sizes. Machine learning approaches for stone tool reassembly may face challenges owing to the requirement for extensive, well-prepared datasets and the computational demands for processing large datasets. Stone tool datasets are often characterized by a lack of extensive pretested data. A highly detailed point cloud representation can limit the applicability of machine learning methods. Specifically, our datasets with 280,000 points per stone can strain the GPU resources [11]. This renders the learning process more challenging.

2.3 Matching Methods for Flake Surface

Iterative Closest Point (ICP) [12], Super4PCS [13], fast point feature histograms (FPFH) [14] and random sample consensus (RANSAC) [15] algorithms are commonly employed in point cloud registration tasks. Furthermore, these methods have been adapted and applied to surface matching scenarios such as flake surface matching.

The ICP algorithm [12], which is one of the cornerstones of point cloud registration, has demonstrated wide-ranging effectiveness in aligning 3D surfaces. However, it may be difficult to obtain an accurate initial alignment, particularly in cases involving partial flake surfaces in stone tools, where achieving a precise alignment can be particularly challenging.

Super4PCS [13], recognized for its proficiency in matching partial 3D shapes, efficiently identifies local surface correspondences. Considering this capacity, it is a compelling candidate for adapting to the task of matching partial flake surfaces. The inherent ability of the algorithm to handle partial matching aligns well with the result of the present study.

The FPFH [14] calculates the correspondences and correspondence probabilities, providing valuable guidance for subsequent RANSAC [15] matching algorithms. In [15], a robust method for estimating transformation models between data points is presented, particularly in the presence of outliers, and is a crucial step in aligning partial flake surfaces. Their approach, which utilizes the FPFH [14] for the initial correlation, addresses challenges related to partial matching and aligning surfaces with potentially irregular shapes.

Our previous study [16] focused on the same task, considering the shape of the flake surface by utilizing contour points, which is similar to our approach. However, it was characterized by prolonged computation times and limitations in effectively matching partial flake surfaces.

3 Proposed Method

3.1 Overview

The input of our method comprised a collection of point cloud of stone tools, acquired through measurements, as described in [3]. Our proposed reassembly method builds upon the pipeline outlined in [16], with a primary focus on reducing the computation time and addressing limitations. The pipeline is executed using the following procedure:

1. Extracting flake surface and identifying contour points.

2. Finding the best flake surface among the candidate flakes. For each flake surface of the core stone, a matching algorithm is run to determine the optimal matching flake surface. This operation calculates a transformation matrix and iterates the process until

the best matching surface is identified within the candidates.

3. Transformation of matched flake. All flake surfaces associated with an unique flake stone of the matched surface are transformed by the transformation matrix.

4. Reconstruction. After integrating the matched flake stone with the core stone, the flake surfaces of the core stone are thereafter reconstructed.

5. Iterative matching. The matching process is iterated until all stone tools are successfully matched.

In contrast to a previous study [16], our study introduces several pivotal enhancements. Initially, the region growing algorithm [17] is employed to extract flake surfaces. However, the algorithm challenges in capturing the boundary points of flake surfaces owing to its sensitivity of normal vector variation and curvatures near sharp edges. To address this shortcoming, we subsequently perform a boundary correction step that has not been used in the previous study. In addition, our matching algorithm features a tunable matching parameter that enables the optimization of flake surface matching for enhanced efficiency. This parameterized approach differs significantly from that used in the previous study. Although the D2 algorithm [18] is employed to select candidate part in the previous study, we decide to employ our matching algorithm by tuning the parameter for candidate ordering, as our candidate ordering improved without D2. Finally, our method performs matching and reconstruction operations directly on the point cloud, as opposed to the mesh-based approach used in the previous study. This shift enhances the computational efficiency.

3.2 Extracting flake surface and identifying its contour points

The region growing algorithm segments a point cloud starting from a seed point and adding neighboring points that satisfy the criteria of geometric features such as differences in point normals and curvatures. Near sharp edges, the large amount of shape variation in these local geometric features can result in over-segmentation near them. This is largely owing to the algorithm's sensitivity to abrupt changes in normal vectors and curvatures.

In the region growing segmentation, the angle threshold α limits the addition of point to those with similar normal vectors in a region, ensuring the smoothness of the region. The curvature threshold c excludes points from the region with abrupt changes in curvatures. The minimum number of points l is the required number of points for a region to be considered valid.

The algorithm categorizes segments as valid or invalid based on the number of points they contain. Regions that are over-segmented, particularly near sharp edges, may be classified as invalid owing to their small size. If the number of points belonging to a segment is less than l, points in these invalid segments are considered unsegmented points, as shown in the gray points in Figure 3. These unsegmented points are crucial, because they contain essential information regarding the true boundaries of the flake surfaces. Therefore, our method corrects the boundary of the region by merging unsegmented points as a re-segmented operation.

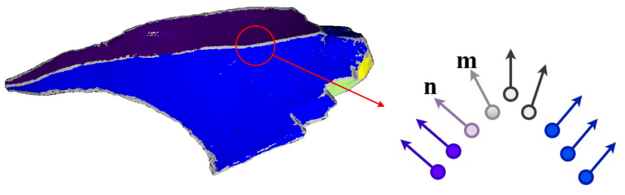


Figure 3: a) Result of the region growing with an angle threshold α set to 1.5°, a curvature threshold c to 0.1 and the minimum number of point in a valid segment l to 500.
b) A case of the unsegmented point with normal vector.

To correct the boundaries of the flake surfaces, The points were re-segmented using five steps. The first two steps are introduced in Point Cloud Library [17]. Steps from 3 to 5 are additional steps to correct boundary points of the segmentation process. The detail of steps is as follows:

1. Running the region growing segmentation with the angle threshold α and curvature threshold c.

2. The segments are categorized into two groups based on the number of points. Regions with a point count exceeding l are considered as valid segments, represented by the purple, blue, green, and yellow points in Figure 3(a). Conversely, those with fewer points are identified as over-segmented regions that included unsegmented points, which are depicted as gray points in Figure 3(a).

3. Find the k nearest neighbors of each unsegmented point. The k is determined that at least one point included in the valid segment is selected as the nearest neighbor point.

4. For each unsegmented point:

   (a) Determine if the neighboring points belong to a valid segment. If a neighboring point belongs to a valid segment, that segment is considered a bordered segment with the unsegmented point. Thus, the unsegmented point is a candidate of merging to bordered segment. This scenario is visualized with the unsegmented point as a light gray point and the neighboring point as a light purple point in Figure 3(b). The neighboring point is considered a border point. When two or more neighboring points are found in the same valid segment, select the nearest neighboring point as the border point.

   (b) Calculate the angle ϕ between the normal vectors of border point invalid segment and the unsegmented point, illustrated as $\mathbf{m}$ and $\mathbf{n}$ in Figure 3(b), respectively. This calculation is computed by Equation (1).

$$\phi = \arccos\left(\frac{\mathbf{n}\cdot\mathbf{m}}{|\mathbf{n}||\mathbf{m}|}\right) \tag{1}$$

   (c) Merge the unsegmented point into the bordering valid segment with the minimum degree of ϕ.

5. Repeat step 4 until all unsegmented point have been merged into valid segments.

After extracting the flake surface using the region growing segmentation with the boundary correction step, the next step involves identifying the contour points of the flake surface. First, a fitting plane is computed for the flake surface using Principal Component Analysis [19] to ensure an accurate representation. Subsequently, all the points within the flake are mapped onto the fitting plane, thereby effectively aligning them with the surface morphology. Finally, a two-dimensional concave hull algorithm [20] is executed. This crucial step aids in the precise identification of contour points in the clockwise direction and offers valuable insights into the characteristics of the flake surface.

3.3 Reassembly

Lithic materials exhibits distinct characteristics representing the joining surface shape and separation from the reassembly of other fractured objects [16], as shown in Figure 4. The first distinguishing property is the sequence of flake generation from a single core [21]. Unlike arbitrary flake matching, stone tool reassembly requires a sequential order. For instance, if flake A precedes flake B in the peeling process, then, considering the sequential extraction of flakes from the core stone, the matching procedure entails a reverse order.

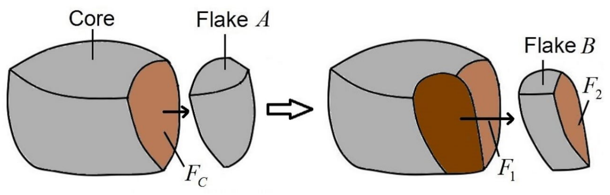


Figure 4: Case of making a stone tool (flake A is peeled first followed by flake B).

The second property pertains to the flat and smooth of most flake surfaces in the matching. This is because of the capacity of the mother stone to divide sharply [1]. Consequently, conventional matching algorithms that rely on surface features may not be suitable.

The third property arises when a single flake surface can fracture into several fragments. As shown in Figure 4, the matching of flake A requires prior matching of flake B with the core stone to create flake surface F_c, which is a composite derived from flake surfaces F_1 and F_2. Consequently, the matched flake surfaces required a reconstruction process to identify the subsequent surface.

In our approach, stone cores are designated manually, initiating the matching process on the flake surfaces of the core stones. As our dataset comprised multiple core stones, they are reassembled in succession. Each flake surface of the core stone is matched with every flake surface of the flakes to identify the best match. To enhance the efficiency and reduce the number of matching tasks, the candidate order for each flake surface is calculated. This order is sorted based on the corresponding candidate score.

3.3.1 New fine-tuned flake surface matching algorithm

A fine-tuned matching algorithm is developed for two purposes during reassembly. It is designed to calculate both the candidate and matching scores in pairwise matching. The matching algorithm calculates a matrix $\mathbf{M}_{st}$, by which a source surface in the candidate stone is mapped onto a target surface in the core stone. The algorithm calculates both the candidate and matching scores by tuning the matching parameter n, which determines the level of detail during the matching process.

Figure 5 shows the difference between the candidate and matching scores for different values of n. For instance, when the matching parameter n is set to 6, the algorithm works on all combinations between every 6 points in the target surface and source contours. When it is set to 12 as an example, the algorithm works on all combinations between every 12 points in the target and source contours. In this case, the number of combinations is reduced, and the working process faster than n is set to 6. In contrast, the accuracy tends to decrease depending on the contour shape. Therefore, n is determined by considering the balance of speed and accuracy.

Our reassembly method requires less computational time to calculate the candidate score. However, the matching accuracy must be high when refitting a candidate for the core stone and calculating the matching score.

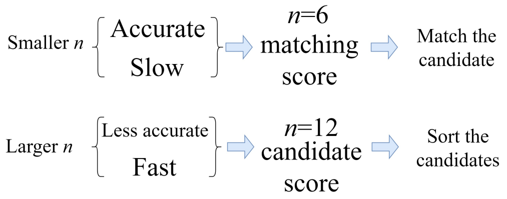


Figure 5: Tuning the precision parameter.

A five-point set $\{\mathbf{p}_i, \mathbf{p}_{i+n}, \mathbf{p}_{i-n}, \mathbf{d}_i, \mathbf{c}_t\}$ is constructed, where $\mathbf{p}_i$ is i-th contour point of the target surface t, n is the matching parameter, $\mathbf{p}_{i+n}$ and $\mathbf{p}_{i-n}$ are separated from $\mathbf{p}_i$ by n, $\mathbf{c}_t$ denotes the center point of the target surface, and $\mathbf{d}_i$ denotes the centroid point of the triangle whose vertices are $\mathbf{p}_{i+n}$, $\mathbf{p}_{i+n}$, and $\mathbf{c}_t$. Similarly, another five-point set $\{\mathbf{p}_j, \mathbf{p}_{j+n}, \mathbf{p}_{j-n}, \mathbf{d}_j, \mathbf{c}_s\}$ is constructed in the same manner on the source surface s, as shown in Figure 6.

The normal vector $\mathbf{k}_i$ of the triangle, control vector $\mathbf{m}_i$, and centroid $\mathbf{d}_i$ of the triangle are calculated using Equations (2), (3), and (4), respectively. All the points and vectors are constructed in the same manner on the source surface. The 3D rotation matrix $\mathbf{R}_{ij}$ is determined by satisfying Equation (6) owing to $\mathbf{k}_i \perp \mathbf{m}_i$ and $\mathbf{k}_j \perp \mathbf{m}_j$. The 3D translation matrix $\mathbf{T}_{ij}$ is calculated using Equation (5). The overall transformation matrix $\mathbf{M}_{ij}$ is thereafter computed using Equation (7), integrating both rotation and translation to facilitate transformation in a 3D space.

$$\mathbf{k}_i = (\mathbf{c}_t - \mathbf{p}_{i-n}) \times (\mathbf{p}_{i+n} - \mathbf{p}_{i-n}) \tag{2}$$

$$\mathbf{d}_i = (\mathbf{p}_{i+n} + \mathbf{p}_{i-n} + \mathbf{c}_t)/3 \tag{3}$$

$$\mathbf{m}_i = \mathbf{d}_i - \mathbf{p}_i \tag{4}$$

$$\mathbf{d}_i = \mathbf{T}_{ij} \cdot \mathbf{d}_j \tag{5}$$

$$\begin{cases} \mathbf{k}_i = \mathbf{R}_{ij} \cdot \mathbf{k}_j \\ \mathbf{m}_i = \mathbf{R}_{ij} \cdot \mathbf{m}_j \end{cases} \tag{6}$$

$$\mathbf{M}_{ij} = \mathbf{T}_{ij} \cdot \mathbf{R}_{ij} \tag{7}$$

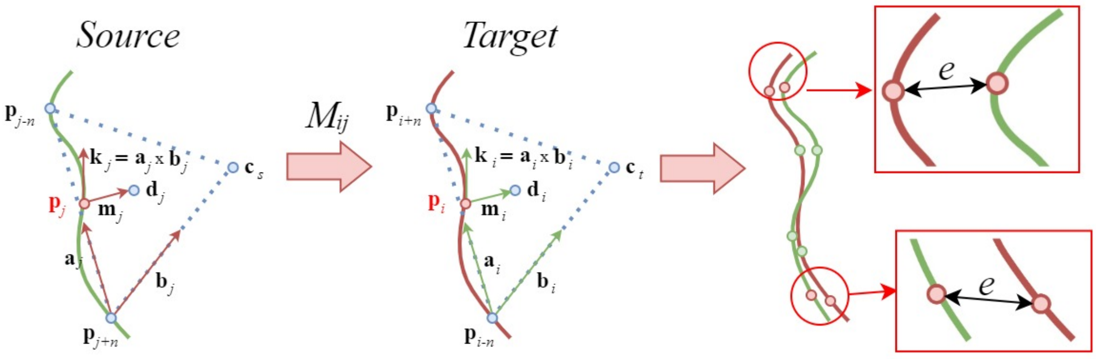


Figure 6: Mapping a source into a target surface.

Equation (8) computes error metric e at point $\mathbf{p}_i$, which is the contour point of the target surface and determines the minimum distance between $\mathbf{p}_i$ and any contour point $\mathbf{p}'_j$ on the transformed source surface by $\mathbf{M}_{ij}$. The number of matched target contour points r_{ij} is calculated using Equation (9), where E_s denotes the maximum edge length in the concave hull [20]. The matching transformation matrix $\mathbf{M}_{st}$, which aligns the source surface s to the target surface t, is selected based on max_r_{st}, which is the highest value of r_{ij} as shown in Equation (10). Algorithm 1 outlines our flake surface matching algorithm; n_t and n_s denote the respective numbers of contour points; $\mathbf{c}_t$ and $\mathbf{c}_s$ denote the center points; n denotes the matching parameter in the input; $\mathbf{M}_{st}$ denotes the matching transformation; max_r_{st} denotes the highest number of matched contour points in the output.

$$e(\mathbf{p}_i) = \min(|\mathbf{p}_i, \mathbf{p}'_j|); \quad j, i+ = n \tag{8}$$

$$r_{ij} = \sum_{i+=n} \begin{cases} 1, & \text{if } e(\mathbf{p}_i) < n \cdot E_s, \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

$$max\_r_{st} = \max(r_{ij}) \tag{10}$$

The uniform point density of the dataset enables a straightforward representation of the matched area using the number of corresponding points. In this context, $\mathbf{q}_i$ denotes any point on the target surface t, whereas $\mathbf{q}_j$ is a point on the source surface s that is nearest point to the tangent plane of $\mathbf{q}_i$.

The two orthogonal distances can be calculated by surface s and t. One is the distance from the tangent plane of $\mathbf{q}_i$ to $\mathbf{q}_j$, and the other is the distance from the tangent plane of $\mathbf{q}_j$ to $\mathbf{q}_i$. In our method, if both distances are less than d_c and the distance between $\mathbf{q}_i$ and $\mathbf{q}_j$ is also less than d_c, $\mathbf{q}_i$ is identified as a correspondence point of $\mathbf{q}_j$. d_c is determined 1.5 by experiment.

Subsequently, matched point pairs of surface s and t are derived and the matched surface percentages are determined using Equation (11). In this equation, P_u denotes the matched area percentage of the surface u, N_u denotes the number of corresponding points of the surface u, T_u denotes the total number of points on surface u. Specifically, P_t and P_s are the instances of P_u for the target surface t and s, respectively. P_t and P_s are indicated the percentage of each matched surface. Moreover, the matching score, denoted as S_{st}, or candidate score, is computed using Equation (12), where max_r_{st} denotes the number of matched contour points, and P_t and P_s represent the matched surface percentage of the target and source surfaces, respectively.

A higher S_{st} value indicates a greater level of matching between the target t and source s surfaces, implying that there are more matched contour points and a larger matched area between them. Conversely, a lower S_{st} value implies a lower matching level, indicating fewer matched contour points and a smaller matched area between the surfaces.

$$P_u = \frac{N_u}{T_u} \tag{11}$$

$$S_{st} = max\_r_{st}\sqrt{P_t P_s} \tag{12}$$

Algorithm 1: Flake Surface Matching Algorithm

```
Input  : contour points of the target, $n_t$, $\mathbf{c}_t$, contour points of the
         source, $n_s$, $\mathbf{c}_s$, $n$
Output: $\mathbf{M}_{st}$, $max\_r_{st}$
$\mathbf{M}' \leftarrow \emptyset$;
for $i \leftarrow 0$ to $n_t$ do
    $\mathbf{p}_i \leftarrow i$-th contour point of the target;
    for $j \leftarrow 0$ to $n_s$ do
        $\mathbf{p}_j \leftarrow j$-th point of the source;
        $\mathbf{M}_{ij} \leftarrow$
          getTransformMatrix($\mathbf{p}_i, \mathbf{c}_t, \mathbf{p}_j, \mathbf{c}_s, n$);
        // by Equation 7
        $\mathbf{M}' \leftarrow \mathbf{M}' \bigcup \mathbf{M}_{ij}$;
        $j += n$;
    end
    $i += n$;
end
$r_{st} \leftarrow 0$;
foreach $M_{ij}$ in $M'$ do
    $S' \leftarrow \{\mathbf{p}'_j \mid \mathbf{p}'_j = \mathbf{M}_{ij} \cdot \mathbf{p}_j,\ \mathbf{p}_j$ is any
      contour point of the source $\}$;
    $r_{ij} \leftarrow 0$;
    for $i \leftarrow 0$ to $n_t$ do
        $\mathbf{p}_i \leftarrow i$-th contour point of the
          target;
        $e = \min \|\mathbf{p}_i - \mathbf{p}'_j\|,\quad \mathbf{p}'_j \in S'$;
        if $e < E_s \cdot n$ then
            $r_{ij} += 1$;
        end
        $i += n$;
    end
    if $r_{ij} > max\_r_{st}$ then
        $max\_r_{st} \leftarrow r_{ij}$;
        $\mathbf{M}_{st} \leftarrow \mathbf{M}_{ij}$;
    end
end
```

3.3.2 Flake surface reconstruction

The process of reconstructing the original flake surface involves the detection and integration of divided flake surfaces. Figure 7 shows a visual representation of the reconstruction, with the flake surfaces F_a in blue and F_b in green. The reconstruction proceeds through the following steps:

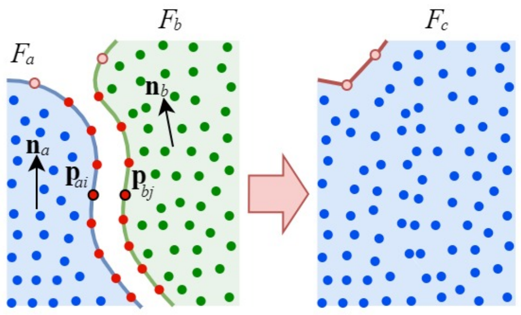


Figure 7: Flake surface reconstruction.

1. Each pair of flake surfaces F_a and F_b from matched flakes undergoes a search for the closest contour point $\mathbf{p}_{bj}$ of F_b for each contour point $\mathbf{p}_{ai}$. If the distance between $\mathbf{p}_{ai}$ and $\mathbf{p}_{bj}$ is less than the threshold named d_r, the pair is designated as a corresponding contour point pair, red points in Figure 7.
2. Calculate the angle between $\mathbf{n}_a$ and $\mathbf{n}_b$, average normal vector of F_a and F_b, respectively. In addition, the number of corresponding pair is counted.
3. The corresponding pair derived by Step 2 is higher than the threshold w and the angle between n_a and n_b is less than the threshold w_θ, then the flake surface pair F_a and F_b is merged into a single flake surface F_c.

4 Results and limitation

The implementation was performed on a PC with an Intel Core i7-10700 CPU and 16 GB memory. The experiment involved testing data from

43 stone models, as shown in Figure 8. The data could be reassembled into three groups. $No.01$ and $No.20$, shown in thick red box are the core stones and Group 1 and 2, respectively. The stone tools belongs to Group 3 are role of dummy data for assembling Group 1 and Group 2. There were 12,300,222 points, with an average of 286,052 points per stone.

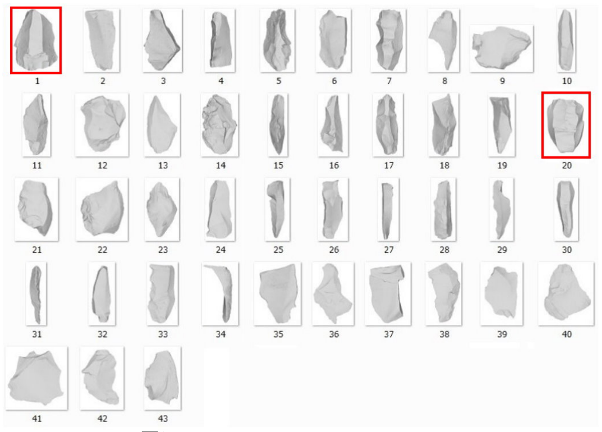


Figure 8: Experiment data.

4.1 Experimental Results

This section describes the evaluation and experimental results. First, we examined an evaluation score and execution time according to the tunable matching parameter n, when n is changed from 1 to 12. Second, we compared our method with related methods. Thereafter, we presented experimental results. Finally, we compared our results with the previous study [16].

To assess the accuracy of the resulting shape, we calculated the evaluation score which is how the matching surface shape coincides. To achieve this, Equation (12) which indicates the surface coincident equation between surface s and surface t, was used as follows:

$$S_e = S_{st}\big|_{n=1} \tag{13}$$

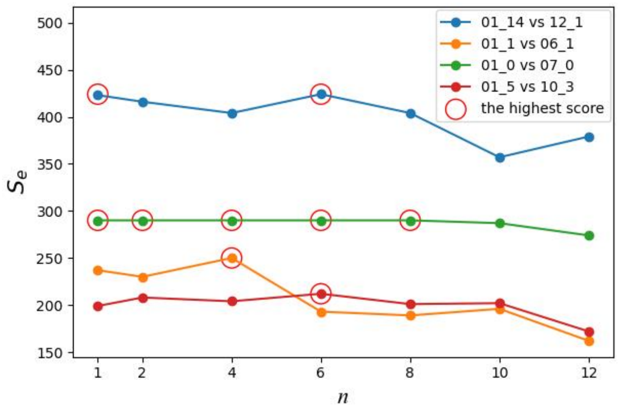
where S_e means score of the surface coincident. To evaluate how to match the surface shape Equation (13) was applied to compare the resulting shape of other methods. In Equation (13), we set n to 1 to be fair to the shape evaluation when we evaluate the resulting shape. In other words, all points belonging to surfaces s and t are used to evaluate the resulting shape. If S_e is high, it means that the shape is more match.

Figure 9(a) illustrates that the evaluation score S_e for the four matching pairs when the matching parameter n varied from 1 to 12. Figure 9(b) shows the computation time for the same parameter variations. Figure 9(a) shows the highest scores for each pair, indicated by red circles. As shown in Figure 9(a), three of the four had highest scores when parameter n was set to 6. In contrast, Figure 9(b) shows the computation time of the four matching pairs. The average computation time of each matching pair for parameter n =6 was 1.52, and that for n =4 was 6.87 s. This shows that when n =6, the computation time is approximately 4.5 times faster than when n =4. Therefore, to balance computation time with accuracy—as reflected by the evaluation score—in the performance of the algorithm, we set n =6 for the comparative analysis.

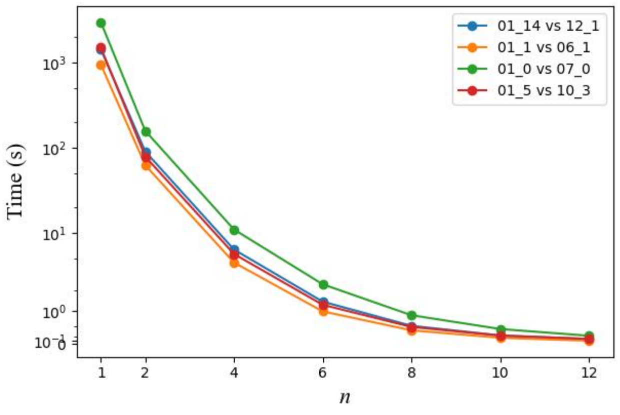
We compare our method and previous works. The comparison involved benchmarking our matching algorithm against Super4PCS [13] and FPFH-SACIA [14] with 100 iterations employed for partial matching tasks. For a fair comparison across all methods, we applied S_e, the evaluation score, calculated by Equation (13).

The ground truth was manually aligned to imitate the real reassembly results achieved by archaeologists, thereby providing a benchmark for evaluation. Table 1 summarizes the results of this comparison. In this table, column "t" denotes the ID of the target surface, formed by combining the stone identification and flake surface ID(e.g.,"01_0" indicates core stone ID "01" and flake surface ID "0"). Similarly, column "s" refers to the source flake surface ID, following the same identification pattern. "S4PCS" represents the absolute difference between the evaluation score of Super4PCS and the evaluation score of the ground truth. "FPFH" and "Our" indicate the same manner. These differences are calculated using Equation(14), where D denotes the absolute difference, S_{gt} denotes the evaluation score of the ground truth, and S_e denotes the evaluation score of the respective algorithm. In this table, the proposed flake surface matching algorithm yields the closest evaluation score to the ground truth across all the target and source pairs. The performance is graphically presented in Figure 10. These comparison results highlight that our algorithm performed better than existing methods for flake surface matching, particularly partial matching.

$$D = |S_{gt} - S_e| \tag{14}$$



(a) Relationship between n and the evaluation score S_e.

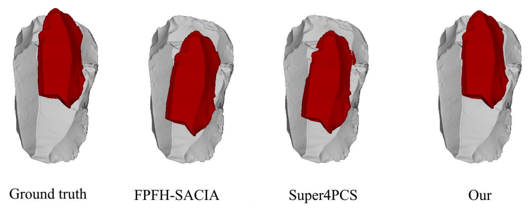


(b) Relationship between n and the computation time.

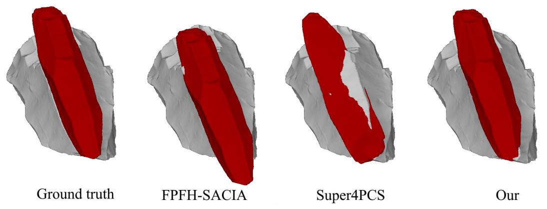
Figure 9: Parameter sensitivity test.

Table 1: The absolute difference between the ground truth and the evaluation score.

t	s	S4PCS	FPFH	Our
01_14	12_1	331.4	374.6	1.7
01_1	6_1	142.9	115.8	19.1
01_0	07_0	248.3	157.6	21.5
01_5	10_3	162.8	157.2	6.5



(a) 01_1 vs 06_1.



(b) 01_5 vs 10_3.

Figure 10: Visual representation of comparison result.

We describe the experimental results of our method. The segmentation process for extracting the flake surfaces requires the estimation of point normal vectors and point curvatures. Previous studies [22] and [19] are employed for normal vector estimation and point curvature calculation, respectively. During segmentation described by Section 3.2, the parameters are tuned as follows: the angle threshold α is set to 1.5°, curvature threshold c is set to 0.1, minimum number of points for a valid segment l is set to 500. The number of nearest neighbor k is set to 4 in the boundary correction step. The dataset in Figure 8 is segmented, extracting 311 flake surfaces and averaging 7 surfaces per model for the matching process. Figure 11 shows the segmentation results for point cloud $No.10$. Figure 11(a) displays the point cloud with point normal vectors, while Figure 11(b) shows the outcome of region growing segmentation using [17], where unsegmented points are shown in gray color, resulting in 67962 unsegmented points of 276336. In contrast, Figure 11(c) shows our boundary correction result, which extracts 8 flake surfaces, and all points are segmented. In Figure 11(b), (c), and (d), the various colors represent different flake surfaces, and the contour points are highlighted in red. Figure 12 (a) and (b) show the results of the segmentation for stones $No.06$ and $No.07$, respectively. The left side of each figure shows the result of the region growing, and the right shows the result of the boundary correction step. As shown in Figure 11 and 12, our method enables to correct boundary.

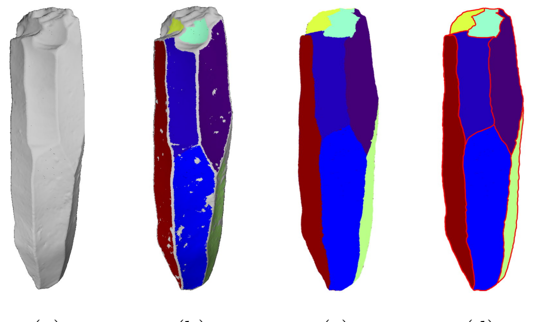


Figure 11: Result of the boundary correction.
a) Point cloud.
b) Result of region growing segmentation.
c) Result of our boundary correction step.
d) Result of extracting contour points.

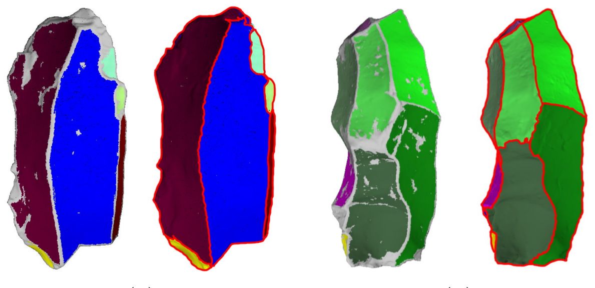


Figure 12: Result of the segmentation.
a) $No.06$.
b) $No.07$.

Figure 13 shows the matching results between the two flake surfaces 01_1 and 06_1. The correlation between the two surfaces is shown in Figure 13(b). Green points represent correlation points, red points represent unmatched surface points and yellow points represents matched contour points, and blue points indicate unmatched contour points on the target surface.

Figure 14 shows a matching scenario in which flake surfaces require reconstruction for alignment. Three flake surfaces of stone tools $No.10$, $No.06$ and $No.16$ were reconstructed to achieve alignment with the surface of stone $No.17$.

Tables 2 and 3 summarize the specific details of matching Groups 1 and 2. In these tables,

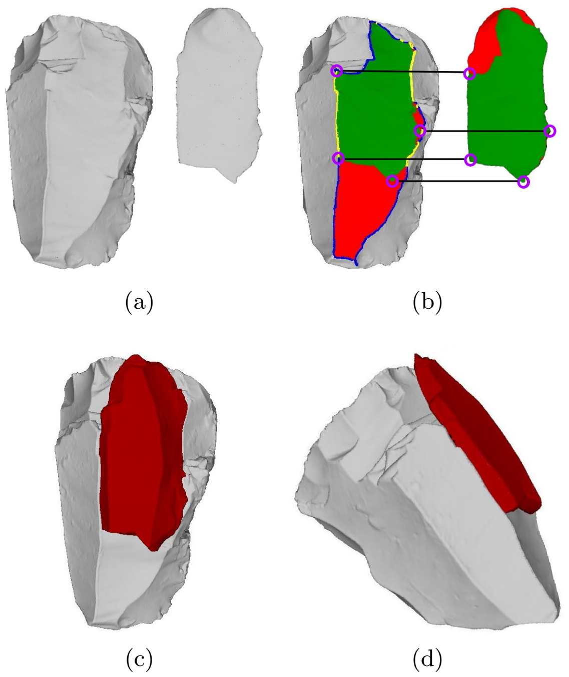


Figure 13: Result of flake surface matching.
a) Stone models $No.01$ and $No.06$.
b) Correspondence between flake surfaces 01_1 and 06_1.
c), d) Result of matching in different views.

columns "t" and "s" maintain the same format as that in Table 1, denoting the respective target and source flake surfaces. The "Order" column signifies the order of flake surface for the best matching, determined by the candidate score. "Time" represents the duration required for computing the fitting transformation matrix for each pair. For instance, finding the best match for flake surface 01_1 in Group 1 requires 44.7 s. On average, each matching process uses approximately 7.45 s per pair, considering 6 (candidate order) matching times. All flake surfaces initiated matching from stone core $No.01$ and $No.20$. During the experiment, the maximum edge length in concave hull E_s was set to 0.5 and the matching tune parameter n was set to 6 for matching, 16 for ordering the candidates. Additionally, we defined two distance thresholds, d_c and d_r, both set to 1.5. The threshold w is set to 30, and the angular threshold w_θ is set to 15 degrees.

Group 1 resulted in the reassembly of 16 stone

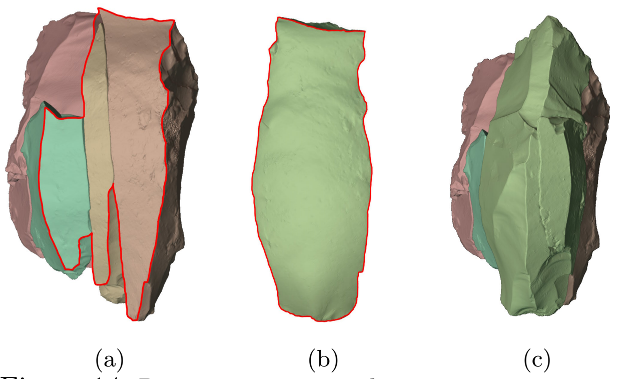


Figure 14: Reconstruction result.
a) Reconstructed flake surfaces of $No.6$, $No.10$ and $No.16$.
b) Source flake $No.17$.
c) Result of matching.

tools, whereas Group 2 was reassembled of 16 stone tools. Flake stone $No.14$ remained unmatched using our method, as explained in detail in Section 4.2. Figure 15 shows the final matching results for both groups, accompanied by images of the manually matched imitations.

Table 4 summarizes the comparison result of our method against our previous study [16] for two groups of stones. In the table, "N.Dis" indicates the normalized distance measurement in millimeters, introduced in previous studies [16], [23]. This measurement method is utilized to identify the most compatible flake surfaces in their studies. It provides a standard to measure the difference between two flake surfaces on a unit area. Essentially, a smaller normalized distance between the surfaces indicates a superior match, signifying a closer similarity or better alignment between two flake surfaces. The column "Time" records the computation time for each method in seconds. The computation time for the previous method is normalized to account for the difference in CPU. Our study was tested on the Intel Core i7-10700, while the previous study utilized an Intel Core i7-4790. The normalization factor of 2.68, derived from the relative floating-point math speed of the two CPUs [24], is applied to the computation times of the previous method to utilize a fair comparison.

In Group 1, our method completed the task in 400 seconds, whereas the previous study required 5912 seconds. In Group 2, our method required 290 seconds, compared with the previous 1641 seconds. In the normalized distance measurements, our method demonstrated an average of 0.012 mm for Group 1, in contrast to 0.033 mm reported in the previous study. Similarly, for Group 2, our method achieved 0.016 mm, compared with 0.027 mm in the previous study. Moreover, we successfully addressed the limitation related to partial matching, allowing us to match 4 more stones in Group 2 and ultimately reassemble all the stones in that group.

Table 2: Reassembly of Group 1

t	s	Order	Score	Time(s)
01_14	12_0	1/260	417.71	27.22
01_1	06_0	6/250	188.40	44.70
01_4	10_6	2/243	306.41	20.96
01_0	07_0	1/234	459.82	37.23
01_49	05_1	1/225	378.01	10.37
01_43	16_0	2/218	219.58	29.09
01_55	11_3	2/209	175.32	19.08
01_39	17_0	1/205	578.64	29.75
01_57	02_0	4/203	140.52	15.20
01_53	18_1	1/193	281.97	37.58
01_59	13_1	1/187	435.26	25.39
01_66	08_3	1/185	554.52	11.05
01_63	19_0	4/178	346.89	57.80
01_64	15_0	3/172	315.48	23.83
01_68	04_1	6/166	147.44	11.11
			Total time:	400.36

4.2 Limitation

Figure 16 shows a scenario in which proposed method could not correctly match for stone $No.14$. Red circles indicate the correspondence of the correct matching. Matched contour points of the two flake surfaces are indicated in yellow, whereas the unmatched contour points are presented in blue. Our matching method is designed for alignment based on the maximum number of matched contour points max_r_{st}. However, in this case, max_r_{st} did not achieved the correct matching. Therefore, our method encountered

Table 3: Reassembly Group 2

t	s	Order	Score	Time(s)
20_10	34_1	1/162	442.66	7.01
20_9	27_2	1/153	360.03	14.29
20_10	36_0	2/146	229.89	19.12
20_5	33_6	1/138	331.59	6.72
20_2	25_2	1/132	219.45	18.61
20_7	31_1	2/127	217.92	18.76
20_12	26_0	4/117	196.15	51.69
20_18	29_0	2/108	234.27	29.87
20_33	31_0	5/102	302.02	33.77
20_5	30_2	3/99	278.34	17.40
20_38	28_1	2/91	305.22	13.58
20_43	35_2	1/83	408.63	7.76
20_0	22_0	3/79	166.33	18.52
20_44	23_3	2/74	60.67	26.32
20_54	24_1	1/64	60.12	6.57
			Total time:	289.98

Table 4: Comparison table with previous study

		N.Dis (mm)	Time (s)
Group 1	Our	0.012	400
	Previous	0.033	5912
Group 2	Our	0.016	290
	Previous	0.027	1641

difficulties in this scenario. In the experiments, this case occurred only once.

5 Conclusion and future work

In this study, we introduced a novel matching algorithm for stone tool reassembly based on contour points derived from flake surfaces. Our approach addressed the critical challenges encountered in a previous study [16], resulting in significant reductions in computation time and improvements in partial matching.

However, it is essential to acknowledge the limitations of our method, particularly in cases in which maximum contour point matching does not yield a correct match. Although these instances are rare, they should be highlighted in future studies.

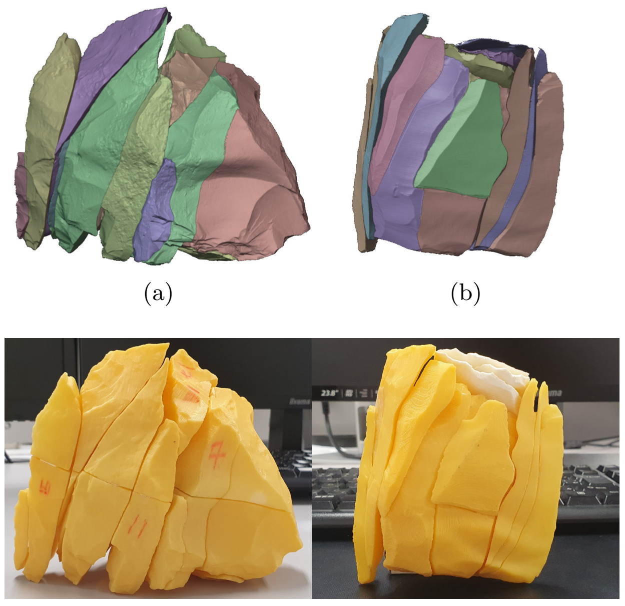

(a) (b) (c) (d)

Figure 15: Result of reassembly.
a), b) Reassembled results by our method.
c), d) Reassembled flakes of two groups.

Acknowledgments

A part of this study was supported by JSPS KAKENHI (Grant Number JP22K00998).

References

[1] K. Matsufuji, S. Monta, Yoku wakaru koukogaku (Understand archaeology), Minerva Shobo, Kyoto, pp. 18, 2010.

[2] K. Suzuki, Koukogaku Nyuumon (Archaeology Introduction), University of Tokyo Press, Kyoto, JP, 1988.

[3] E. Altantsetseg, Y. Muraki, F. Chiba, K. Konno, 3D Surface Reconstruction of Stone Tools by Using Four-Directional Measurement Machine, International Journal of Virtual Reality, Vol. 10, No. 1, pp. 37-43, 2011.

[4] Q. Huang, S. Flory, N. Gelfand, M. Hofer, Reassembling Fractured Objects by Geometric Matching, ACM Transactions on Graphics(TOG), vol. 25, Issue 3, pp. 569-578, 2006.

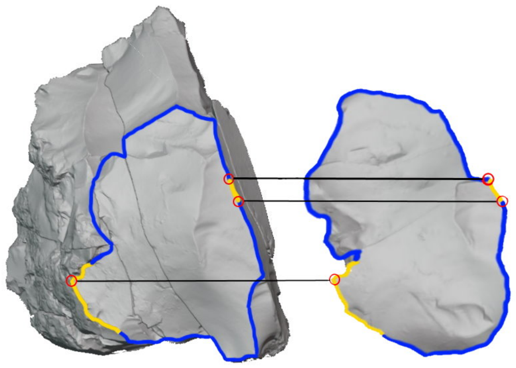


Figure 16: Limitation of our method: Failed matching scenario for stone $No.14$.

[5] B. Bronn, C. Toler-Franklin, A System for High-Volume Acquisition and Matching of Fresco Fragments: Reassembling Theran Wall Paintings, ACM Transactions on Graphics(TOG), vol. 27, Issue 3, pp. 1-9, 2008.

[6] A. Wills, Stochastic 3D Geometric Models for Classification, Deformation, and Estimation. Ph.D. thesis, Brown Univ. Press., 2004.

[7] J. H. Hong, S. J. Yoo, M. A. Zeeshan, Y. M. Kim and J. Kim, Structure-from-Sherds: Incremental 3D Reassembly of Axially Symmetric Pots from Unordered and Mixed Fragment Collections, 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, pp. 5423-5431, 2021.

[8] R. Gregor, D. Bauer, I. Sipiran, P. Perakis, T. Schreck, Automatic 3D Object Fracturing for Evaluation of Partial Retrieval and Object Restoration Tasks - Benchmark and Application to 3D Cultural Heritage Data, The Eurographics Association, 2015.

[9] N. Lamb, S. Banerjee, N. Banerjee, Automated reconstruction of smoothly joining 3D printed restorations to fix broken objects, In Proceedings of the 3rd Annual ACM Symposium on Computational Fabrication (SCF '19), Association for Computing Machinery, New York, NY, USA, Article 3, pp. 1 – 12, 2019.

[10] N. Lamb, C. Palmer, B. Molloy, S. Banerjee, N. Banerjee, Fantastic Breaks: A Dataset of Paired 3D Scans of Real-World Broken Objects and Their Complete Counterparts, 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, pp. 4681-4691, 2023.

[11] M. Zhang, H. You, P. Kadam, S. Liu, C. Kuo, PointHop: An Explainable Machine Learning Method for Point Cloud Classification, IEEE Transactions on Multimedia, vol. 22, no. 7, pp. 1744-1755, 2020.

[12] P. Besl, N. McKay, A method for Registration of 3-d Shapes, IEEE Trans Pattern Anal Mach Intel, vol. 14, no. 2, pp. 239 – 256, 1992.

[13] N. Mellodo, N. Mitra, D. Aiger, Super 4PCS Fast Global Point cloud Registration via Smart Indexing, Computer Graphics Forum, vol. 33, no. 5, pp. 205 – 2015, 2014.

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

[15] M.A. Fischler, R.C. Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography, Readings in Computer Vision, pp. 726 – 740, 1987.

[16] X. Yang, K. Matsuyama, K. Konno, A New Method of Refitting Mixture Lithic Materials by Geometric Matching of Flake Surfaces, The Journal of the Society for Art and Science, Vol. 15, No. 4, pp. 167-176, 2016.

[17] Point Cloud Library, Region growing segmentation.

[18] R. Osade, T. Funkhouser, B. Chazelle, D. Dobkin, Matching 3D models with shape distributions, Shape Modeling and Applications, SMI 2001 International Conference, pp. 154-166, 2001.

[19] Mark Pauly, Richard Keiser, Markus Gross, Multi-scale Feature Extraction on Point-Sampled Surfaces, Eurographics, Vol. 22, No. 3, 2003.

[20] H. Edelsbrunner, D. Kirkpatrick, R. Seidel, On The Shape of a Set of Points in The Plane, Information Theory, IEEE Transactions on Information Theory, vol. 29, no. 4, pp. 551- 559, 1983.

[21] A. Igarashi, Hakuhen hakuri genri: Seisei no zengo kankei (Flakes peeling principle: The context of generation), Sekki zukuri no ziken koukogaku (Experimental archaeology of the Stone tool Making), Lithic Technology Research Society, Gakuseisha, Tokyo, JP, pp. 22-35, 2004.

[22] A. Altansukh, M. You, E. Altantsetseg, O. Khorloo, K. Konno, A Study on Automatic Flake Surface Segmentation of Stone Tools by Calculating Shape Features, IWAIT, SPIE Digital Library, Vol. 12592, pp. 123-128, 2023.

[23] K. Yamahara, K. Konno, F. Chiba, M. Satoh, A Method of Detecting Adjacent Flakes in Stone Tool Restoration by Extracting Peeling Surfaces, Japan Society for Archaeological Information., Vol. 17, No. 1-2, pp.23-31, 2011.

[24] Inter i7-10700 vs. Intel i7-4790, CPU Benchmark. Accessed on: Mar.17, 2024, Available: https://www.cpubenchmark.net/compare/3747vs2226/Intel-i7-10700-vs-Intel-i7-4790.

Amgalan Altansukh

is a Ph.D candidate of Graduate School of Science and Engineering at Iwate University. He received B.S. and M.S degrees from the National University of Mongolia in 2018 and 2019. His research interest include geometric modeling, virtual reality, and simulation.

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.

Enkhbayar Altantsetseg

received the B.S. and M.S. in mathematics from National University of Mongolia in 1995 and 1997, respectively. He earned his Dr. Eng. in design and media technology from Iwate University in 2013. He is currently a professor of the School of Information Technology and Electronics of National University of Mongolia. His research interests include geometric modeling, virtual reality, augmented reality, and simulation.

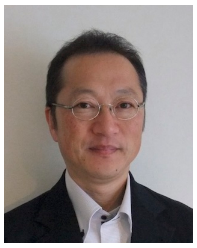
Oyundolgor Khorloo

is currently an associate professor in the Department of Information & Computer Science at School of Information Technology and Electronics of National University of Mongolia. Her research interests include computer graphics, virtual reality, noise-based animation, and simulation of natural phenomena. She received a B.S. degree in applied mathematics from National University of Mongolia in 1995 and an M.S. degree in computer science from University of Colorado at Denver in 2001. She received her Dr. Eng. in design and media technology from Iwate University in 2012.

Fumito Chiba

received the Dr.Eng. degree from the Department of Electronic Engineering and Computer Science, Graduate School of Engineering, Iwate University. He worked as a research associate in the Department of Computer Science, Faculty of Engineering, Iwate University, and now he is the Managing Director of LANG CO., LTD. He is engaged in research and development of 3D shape measurement and processing of archaeological artifacts. He is a member of Japan Association for Archaeoinformatics.

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, 3D surface 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.