

PuzzleFixer: A Visual Reassembly System for Immersive Fragments Restoration

Shuainan Ye, Chen Zhu-Tian, Xiangtong Chu, Kang Li, Juntong Luo, Yi Li, Guohua Geng, and Yingcai Wu

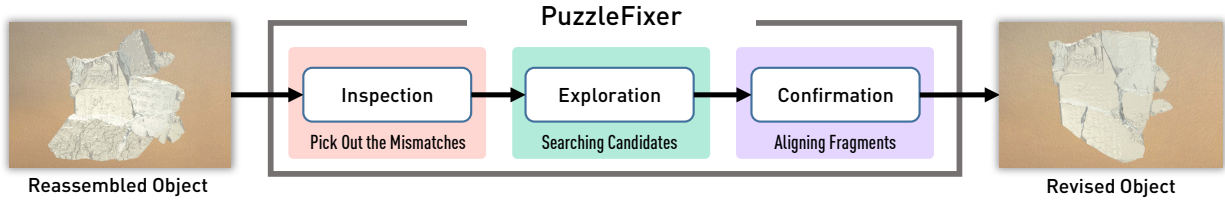


Fig. 1. Reassemble a defect object via PuzzleFixer. Given a set of fragments with matching errors, PuzzleFixer successively provides three stages to solve the puzzle: 1) inspection stage that allows users to find and pick out the mismatches, 2) exploration stage that supports searching for possible solutions from a large number of matching candidates, and 3) confirmation stage that ensures the accuracy and efficiency of manual alignment of fragments.

Abstract—We present PuzzleFixer, an immersive interactive system for experts to rectify defective reassembled 3D objects. Reassembling the fragments of a broken object to restore its original state is the prerequisite of many analytical tasks such as cultural relics analysis and forensics reasoning. While existing computer-aided methods can automatically reassemble fragments, they often derive incorrect objects due to the complex and ambiguous fragment shapes. Thus, experts usually need to refine the object manually. Prior advances in immersive technologies provide benefits for realistic perception and direct interactions to visualize and interact with 3D fragments. However, few studies have investigated the reassembled object refinement. The specific challenges include: 1) the fragment combination set is too large to determine the correct matches, and 2) the geometry of the fragments is too complex to align them properly. To tackle the first challenge, PuzzleFixer leverages dimensionality reduction and clustering techniques, allowing users to review possible match categories, select the matches with reasonable shapes, and drill down to shapes to correct the corresponding faces. For the second challenge, PuzzleFixer embeds the object with node-link networks to augment the perception of match relations. Specifically, it instantly visualizes matches with graph edges and provides force feedback to facilitate the efficiency of alignment interactions. To demonstrate the effectiveness of PuzzleFixer, we conducted an expert evaluation based on two cases on real-world artifacts and collected feedback through post-study interviews. The results suggest that our system is suitable and efficient for experts to refine incorrect reassembled objects.

Index Terms—Immersive visualization, interactive exploration, fragment reassembly, cultural heritage

1 INTRODUCTION

Reassembly, which reconstructs the fragments of a fractured object to restore its initial state, is an imperative task to recover the function and value of the object. It has proven benefits across various application domains, such as artifact reassembly in archaeology [43], fracture restoration in surgery [58], and physical evidence reconstruction in forensics [26]. Prior research on reassembly is built upon the advances in 3D scanning technology and geometry processing, mainly developing automatic approaches that match the adjacent fragment pairs [28, 39] and gradually merge the pairs until the fractured object is restored

[53, 68]. While these methods significantly improve the reassembly efficiency, their success rates are still far from satisfying as the broken pieces and missing fragments cause large noises and ambiguities [22, 24]. Therefore, experts need to manually fix the errors after object is automatic reassembled. For example, analysts usually observe the semantic features (e.g. textures and global appearance) to identify the mismatches and adjust the position and rotation of the fragments when manually reassembling an object [9, 20].

Various techniques have been proposed to facilitate manual reassembly of 3D fragments, including direct interaction [33, 48], automatic alignment assistance [36] and immersive reassembly tools [20, 50]. However, most of these techniques assemble the fragment from the scratch, requiring for many efforts from users to find and align the fragments. Recently, Lima-Hernandez et al. [9] has largely simplified such laborious work by introducing automatic techniques into manual reassembly. They pre-assembled the fragment pairs to derive coarse results before manual correction. However, these previous methods do not provide visualization of the internal match relation between multiple reassembled fragments, which is essential for experts to identify the mismatches during reassembly correction.

We collaborate with six experts from cultural relics protection to develop our system that facilitates object correction. Considering that the fragments and the reassembling interactions in our situation are inherently three-dimensional, we adopt virtual reality (VR) techniques which enable realistic perception of the 3D objects and direct control over the movement and rotation of the fragments. We identify two challenges in developing the system. First, finding the optimal

- S. Ye, X. Chu, and Y. Wu are with the State Key Lab of CAD&CG, Zhejiang University. They are also with the Laboratory of Art and Archaeology Image (Zhejiang University), Ministry of Education, China. E-mail: {sn_ye, xiangtongchuu, ycwu}@zju.edu.cn.
- C. Zhu-Tian is with John A. Paulson School of Engineering and Applied Sciences, Harvard University, Cambridge, MA. E-mail: ztchen@seas.harvard.edu.
- K. Li and G. Geng are with Northwest University. E-mail: {ltkang, ghgeng}@nwu.edu.cn.
- J. Luo and Y. Li are with Xi'an Beilin Museum. E-mail: ljuntong@foxmail.com, liyi_beilin@163.com.
- Y. Wu and K. Li are the co-corresponding authors.

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxx/TVCG.201x.xxxxxxx

solution to correct the mismatches is laborious since it requires frequent observation and comparison of the shape and match relations among multiple reassembly solutions. The massive reassembly possibilities are overwhelming for experts as they need to determine the reassemblies correctness based on excessive fragment relations and detailed information on the surface. Second, it is difficult to rectify the alignment between multiple fragments. Direct mid-air interaction provided by the immersive system allows users to intuitively transform the fragments [22] and specify which part of the fragment surface should be matched and aligned. However, when matching multiple fragments, users need to simultaneously control the transformation of each fragment and justify the alignment accuracy of fragments with several factors (e.g. the alignment of detailed surface and the overall appearance). This reduces the direct interaction efficiency as the scale and shape complexity of the fragments increase.

To overcome these challenges, we propose PuzzleFixer, an semi-automatic interactive system for experts to better visualize and correct the defective reassembled object in the immersive environment. Based on automatic matching techniques, PuzzleFixer provides multi-level exploration and instant feedback, which enables detection of reassembly errors, identification of possible matches, and refinement of detailed alignments. To address the first challenge, PuzzleFixer automatically generates reassembly categories according to matching similarity. Each category provides coarse matching results (e.g. which fragments are matched and their coarse transformation), allowing experts to exclude unreasonable solutions and explore the remaining ones in depth. To ease matches finding, we place the solutions in front of users and order the solutions by their matching similarities. For the second challenge, we augment reassembled objects with 3D node-link network where a node represents a fragment and an edge connects a pair of matched fragments. Based on the network, we design visual and force feedback to help users align the fragments. Specifically, when users are interacting with fragments, PuzzleFixer instantly generates edges between the matching faces on the fragments and guides the fragment transformation by introducing resistance along the edges of the matched faces to prevent these faces from being separated. We demonstrate the usability and effectiveness of our system by conducting two case studies based on a real-world cultural relic restoration. We further collected expert feedback from a semi-structured interview and reported insights and suggestions about our system.

To summarize, our contributions are listed as follows:

- We developed PuzzleFixer, an interactive system for defective reassembled object correction, featuring reassembly cluster visualization for solution exploration and instant feedback guidance for efficient alignment.
- We conducted case studies on the reassembly of a set of real-world culture relics fragments and collected feedback from six domain experts, which provided insights into reassembly correction.

2 RELATED WORK

Automatic Reassembly Process for Virtual Fractured Object. Automatic reassembly of fractured objects is a vibrant and complex topic in graphics and cultural heritage research. It aims to calculate the relative position of each fragment model (e.g., 3D meshes converted from physical fragments) and construct the virtual entirety shape in a fully automatic way. Due to the great advantages in terms of efficiency and avoidance of secondary damage, the automatic reassembly process has been extensively studied for various applications [4, 40, 58, 65]. Papaioannou et al. [39] systematically presented the pipeline for automatic fragment reassembly, which classified the process into two phases: pairwise matching and multi-part optimization.

The pairwise matching phase generates a set of potentially correct matches that align pairs of fragments through fracture surfaces. It extracts various geometry features from the fragment models, searches all possible fragment pairs, and provides each pair a score indicating its match quality [12, 16, 21]. For example, Huang et al. [21] used a forward search algorithm to find and measure the fragment pairs based on the sharpness and roughness similarity of faces. Later, Zhang et al. [68] improved the matching robustness for thin-shell objects

by matching fragments against predefined templates before matching search. Recently, a coarse-precise matching strategy [28, 65] was proposed, which rapidly excludes a large number of impossible matches based on salient features before searching and aligning fragment pairs precisely. In this work, we leverage various state-of-the-art methods [28, 32, 63] depending on the fragment type used in our case studies to generate potential fragment pairs with scores and alignments for manual reassembly correction.

The multi-part optimization is responsible for determining correct matches and constructing the entire object. Current methods for multi-part optimization are generally based on graph optimization. In particular, a reassembly graph is generated where a node represents a fragment and an edge with weight represents a pairwise match and its matching score. Reassembly can be built by finding the maximum cumulative score on edges using the shortest path algorithm [40, 41]. To increase the quality, some of the common constraints, such as loop closure [24] and non-intersection constraint [21], are introduced to ensure stable transformation of fragments without penetration. Building upon the constraints, Zhang et al. [68] iteratively selected the edge to maximize the score using the greedy-based method. Sizikova et al. [53] proposed a genetic algorithm to increase the robustness of noisy and incorrect matches. Although these methods are promising, the results are not always reliable due to low quality of fragments caused by inevitable field damage [39] as well as local optimum problem induced from optimization algorithms. We leverage multi-part optimization approaches to pre-reassemble the object before manual refinement and obtain match relations between fragments. Based on the result, we design efficient visualizations and interactions to help experts identify the problem region and refine the reassembly in the immersive environment.

Immersive Interface for Manual Puzzle Solving. Solving the virtual fragment puzzle manually requires users to understand the process of reassembly and accurately control the fragment model in 6 degrees of freedom (DOF). Given the benefits of visualizing and interacting 3D data [66, 69, 72], many immersive systems have been developed for 3D reassembly tasks [6, 7, 20]. One of the most common approaches for enhancing interaction accuracy is to augment the object with visual cues. For example, Sreng et al. [54] designed sphere and arrow glyphs with visual effects to indicate real-time distances and angles between objects. Weiß et al. [60] overlaid the target region with a crosshair glyph and pie chart to show the splicing position and transforming angle while operating objects. Besides visual guidance, some approaches propose the precise mode to improve the accuracy of manual operation [22, 46, 50]. In contrast to direct interaction with free-hand movement, the precise mode only allows objects to be transformed in one DOF at a time. Apart from the fully manual method, some studies designed the snap effect [20, 33] that automatically aligned a fragment pair using geometry processing such as iterative closest point (ICP) [49].

However, these methods do not consider aligning multiple fragments with complex shapes and match relations. As the number of regions on the fragments increases, visualizing and specifying the correct regions to align becomes a challenge. We design instant visual and force feedback according to the relative position of the current fragments to improve the efficiency of manual alignment.

Integrating Human Knowledge with Automatic Reassembly techniques. Due to the limitations in the automatic reassemble process and manual matching in practice, semi-automatic tools that integrate human knowledge with matching algorithms have been developed to empower the reassembly performance. One of the integrating strategies is to precompute pairwise matching candidates for users to identify correct matches. For example, Brown et al. [5] allowed users to iteratively compare and identify matches through small multiples after all candidates were automatically found. Similarly, Deever and Gallagher [10] visualized the automatically assembled document pieces in order of similarity to facilitate evaluation. Besides, to improve the candidate quality, Palmas et al. [37] enabled users to define alignment constraint points directly on fragments to instruct the automatic matching process. Yet these approaches are worked in the traditional desktop environment. Recently, 3D Puzzling Engine [9] visualized precomputed rough alignments of fragment pairs in immersive space. These pairs were

augmented with matching lines that served as cues for users to estimate and manually refine the matches.

While most systems enable users to assemble the unaligned fragments, only a few studies have investigated refining the reassembled object. Refining the reassembly introduces extra tasks, including identifying mismatches in objects and finding the correct reassembly from numerous candidates. We propose PuzzleFixer, which visualizes match relations with an embedded node-link diagram. Our system further enables level-of-detail explorations that visualize and filter the potential reassemblies from coarse match relations to fine shape details.

3 BACKGROUND

Automatic reassembly approaches generate reassembled objects for expert review and correction. This section introduces the concept of automatic reassembly and describes the data produced by automatic reassembly techniques.

3.1 Automatic Reassembly

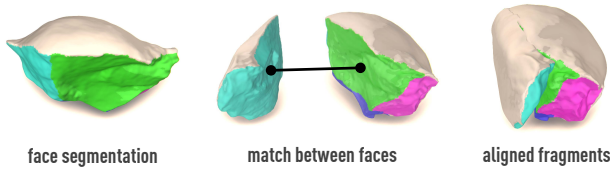


Fig. 2. A demonstration of the automatic reassembly procedure (models from huang et al. [21]). Left: The surface of a fragment is segmented into fractured faces (colored) and intact faces (white). Middle: A pairwise match between two fragments. Right: The aligned fragments.

Automatic reassembly identifies the fractured face of each fragment, and determines fragment relations (*which fragments are matched*) and their transformation (*how the matched fragments align with each other*). The final result of automatic reassembly is a reassembled object comprising all fragments with their respective transformations determined. Specifically, there are three steps in automatic reassembly, as shown in the following:

- **Step 1: Face Segmentation.** The fragment surface is categorized into 1) **fractured faces** caused by object damage, which are to be matched with other fractured faces; and 2) **intact faces**, which are used to construct the profile of the original object (Fig. 2 Left).
- **Step 2: Pairwise Matching.** With the fractured faces, the reassembly procedure finds the potential matches between all pairwise fragments and generates alignment for each match. Here we define a **match** is an adjacent relation, which refers to a pair of fractured faces of two fragments that can be adjoined against each other (Fig. 2 Middle). An **alignment** is the transform of the matched fragments (Fig. 2 Right). It comprises the transformation of the matched fragments to make the fractured faces merge with each other. For each match, a **score** is recorded to represent the alignment accuracy.
- **Step 3: Multi-part Optimization.** Finally, a reassembly is generated from a set of pairwise alignments by the multi-part optimization approach. Specifically, the procedure gradually selects a pairwise match from the set that maximizes the accumulated matching score, merges the match to the current reassembly while optimizing all the reassembled alignments under certain constraints (e.g. prevent interpenetration). All possible selections construct a **reassembly space**, which includes all combinations of the fragments with different alignments. It has been proven that search for the optimal reassembly from a reassembling space is a NP-hard problem [11].

In summary, automatic reassembly provides pairwise matches and alignments with accuracy scores and constructs an entire object from the reassembly space.

3.2 Data Description

The data produced by automatic reassembly consists of shape and relation data. To visualize and interact with the reassembly from pairwise matches to global alignments, we further differentiate the two kinds of data from different levels.

Data Type from reassembled objects includes the 3D shape and its corresponding match relations.

- **Shape Data** contains information of 3D fragment models that illustrates what the reassembly look like. The shape of a fragment is defined by: 1) a triangle mesh that constructs the surface of the fragment, and 2) the transformation (i.e., position and rotation) of the fragment. Shape data describes fragment faces and alignments, and provides virtual models for experts to visualize and interact with.
- **Relation Data** includes the fragment matches and their corresponding scores, explaining how fragments are reassembled. Previous studies have introduced the concept of graph structures to represent fragment relations, including weighted graphs [21], directed multi-graphs [24] and reassembly graphs [68]. In our study, we describe fragment relations with a matching graph $G = (F, E)$. Specifically, each node $f_i \in F$ in the graph denotes a fragment that contains M segmented faces $s_{i_m} \in f_i, m = 1, \dots, M$. An edge $e_{i_m, j_n} \in E$ represents a match between the m -th face on f_i to the n -th face on f_j . Each edge e_{i_m, j_n} includes a score p_{i_m, j_n} that measures the alignment accuracy between s_{i_m} and s_{j_n} . Particularly, if a s_{i_m} have not yet found its matched fractured face, we denote its edge as $e_{i_m, -1}$ and assign it a zero score. Therefore, relations in a reassembly can be illustrated by an undirected graph weighted by scores.

Data Level decides the level of details that visualizations and interactions can access. The data produced by automatic reassembly ranges from the transformation between fragment pairs in one reassembly to a set of reassemblies. Therefore, we further categorize the data into three levels (Fig. 3):

- **Candidate Level** data consists of a set of possible reassemblies from the reassembly space. Each reassembly is a reassembled object with different matches and alignments. Hence, candidate level data mainly provides an overview of all possible solutions to reassemble the given set of fragments.
- **Fragment Level** data is obtained from a set of reassembled fragments. The relation between a fragment pair f_i, f_j is represented by a match together with the average score on all aligned fractured faces, which is denoted as $e_{i, j}$ and $p_{i, j}$. Data at this level provides an important basis for correctness judgment of reassembly, such as texture continuity across adjacent surfaces and the score from the inside aligned fractured faces. Therefore, the fragment level data is mainly used for correctness check of a reassembly.
- **Face Level** data comprises both intact and fractured faces on each fragment. Each fractured face is characterized by one score. Unlike automatic matching techniques generally based on geometrical features of fractured faces, human experts further use semantic features (e.g. textures and inscriptions) on intact faces to visualize the reassembly. Reassembling at this level focuses on refining the matches between faces.

4 REQUIREMENT ANALYSIS

To guide the system design, we first abstract the reassembly tasks into inspection, exploration, and confirmation stages. We then derive experts' requirements in each stage.

4.1 Workflow for Reassembly Optimization

To clarify the tasks experts need to perform in the correcting process, we conducted a semi-structured interview with two experts in the digital restoration domain. One expert is well experienced in automatic reassembly and digital protection of cultural heritage for over ten years; the other is a professor who specializes in geometry processing and visualization for 3D cultural heritage artifacts.

In the interview, the experts revealed that optimizing the reassembly relies on the selection of correct matches and interactions for refining alignments. To ensure the efficiency in such process, they conduct analytics tasks on both reassembled object observation and possible solutions searching. *Observation task* aims to judge the correctness of the reassembly and identify mismatches. Based on the observation, *exploration task* is conducted to learn the reassembly space and determine potential matches that can be aligned correctly. Subsequently,

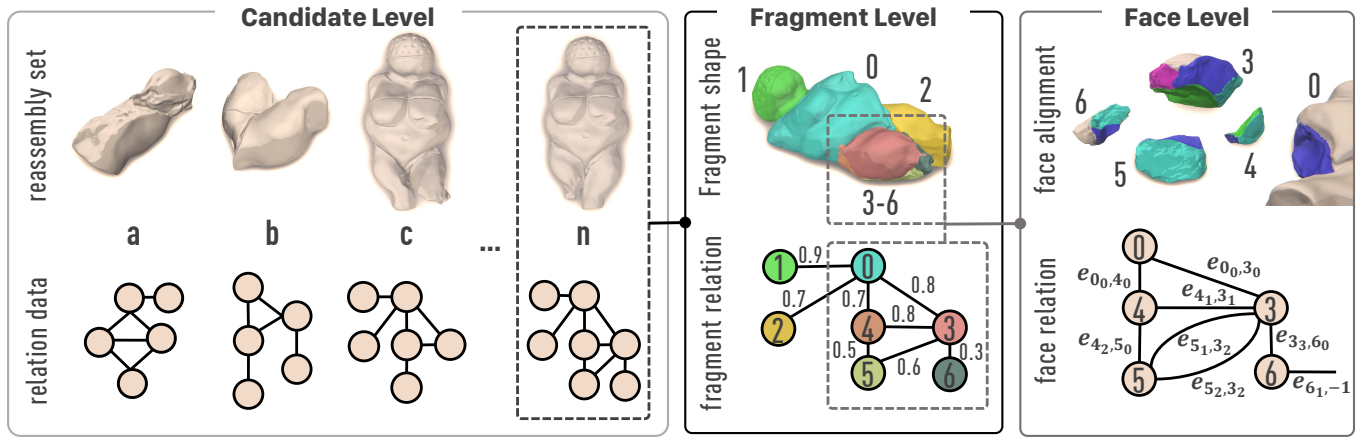


Fig. 3. Three levels for characterizing the reassembling data (models from huang et al. [21]). The candidate level includes all possible combinations of fragments. The fragment level data serves a fragment as a unit and reveals the quality through fragment relation scores (numbers on the edges). The face level data serves one face relation as a unit, shows which faces are used to align the fragments (e.g. f_5 aligns with f_3 with 2 faces, and f_6 has one mismatched face).

they conduct *alignment task* to carefully align the matched fragments by adjusting their transformation through direct manipulation. *Completion task* is finally performed by automatic reassembly that restores the entire object based on the refined alignments. In addition, they perform all the tasks iteratively since matches changed in one region may lead to changes and even errors in other regions, thus causing a knock-on effect.

Based on interview results, we propose a three-stage iterative workflow that integrates the tasks with automatic reassembly techniques for reassembly refinement (Fig. 4):

Inspection Stage: At the beginning of optimization, experts conduct the *observation task* to identify mismatches from the reassembled object, which is generated by automatic reassembly approaches. The mismatches are unveiled by several features, such as the overall discordant shape appearance, discontinuous surface texture between adjacent fragments, and low matching score measured by feature similarity on matched fractured faces. When mismatches are identified, the expert needs to disconnect the corresponding edges and divides the object into multiple reassembly groups.

Exploration Stage: In this stage, the expert performs *exploration task* to determine possible solutions for mismatches. These solutions can be obtained by traversing all possible alignments between matches. However, the reassembly space might be too large and presenting all matches would be overwhelming. Therefore, this stage allows the expert to efficiently exclude impossible matches and identify a subset of interests from the large reassembly space.

Confirmation Stage: With a set of potential matches determined, the expert then needs to confirm the correct matches, refine their alignments, and enter the next reassembly iteration through the *alignment* and *completion tasks*. Precisely aligning multiple fragments by direct interaction is a challenge task, given that merging multiple irregular faces simultaneously without destroying other aligned faces. Thus, this stage provides interaction aids to reduce the difficulty and improve alignment accuracy. When experts are satisfied with the current solution, automatic reassembly approaches will precisely align the mismatches that do not covered in the selected candidate and generate the entire object based on the interaction results.

4.2 Design Requirement Analysis

Based on the workflow, we further communicated with experts, demonstrated several prototypes of our reassembly system, and collected feedback regarding detailed concerns of each stage. After several rounds of refinement, we distilled six design requirements centered around the three stages to guide the design of our system.

Inspection Stage:

- **I1: Visualize fragment shape with relation data.** The shape and

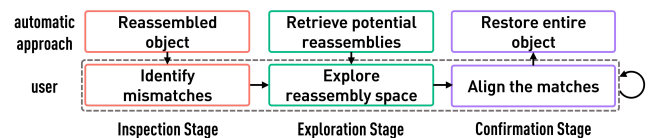


Fig. 4. The workflow of PuzzleFixer consists of inspection, exploration, and confirmation stages. In the first two stages, automatic approaches provide the data for users to inspect mismatches and explore the re-assembly space. When the correct matches are aligned and confirmed in the last stage, the automatic approach restores the entire object.

relation data are crucial for mismatch observation as they provide match evidence at the fragment level (e.g. how the reassembled object looks like) to justify the reassembly correctness and the face level (e.g. similarity between aligned faces) to find mismatched fractured faces. Thus, the system should integrate fragment shapes with relation data to facilitate the identification of mismatches.

- **I2: Provide intuitive interaction for modifying the matching graph.** Interacting with the matching graph is essential to specify and remove the mismatches between fractured faces. However, prior direct interactions on fragment shapes [22, 48, 50] are not enough to select and remove relation data at face level. Thus, the system should provide intuitive interactions for match relation modification.

Exploration Stage:

- **E1: Present potential reassemblies based on match evidence.** To help experts quickly identify candidates with high matching potential, reassemblies should be organized in a certain order. Since candidates are selected by experts based on match evidence, reassemblies should be ordered by evidence similarity.
- **E2: Reveal matching patterns of possible reassemblies.** To exclude a large number of unreasonable candidates from the reassembling space, experts need to understand overall patterns of the space, such as what types of matches are available and the approximate appearance of each type. Hence, the system should cluster candidates based on similar match evidence, provide a visual summary of each cluster to assist experts with correctness justification and focus on the interested subset of candidates.

Confirmation Stage:

- **C1: Enhance awareness of face relations based on instant feedback.** Aligning fragments manually is challenging since face correspondence will change in real-time with variations in the relative positions of fragments. Experts need to constantly check the face relations to perform appropriate fragment transformation, such as “Which faces are currently facing?” and “Which direction should not be moved to retain aligned faces?” Instant feedback on the face relations can guide experts in fragment alignments and interaction

efficiency enhancement [29, 54, 56, 73].

- **C2: Generate precise reassembly according to alignment interactions.** Considering the imprecision nature of the direct interactions [34], the system should instantly align manually transformed fragments precisely to improve the alignment quality. Moreover, after the expert finishes the confirmation, the system needs to match the mismatches not included in the chosen candidate and update the relation data accordingly to form a complete object.

5 PUZZLEFIXER: IMPLEMENTATION AND DESIGN

Based on the requirements, we design and implement PuzzleFixer, a visual reassembly system that enables experts to refine the reassembled object in VR. In this section, we first present a usage scenario of the system to demonstrate the workflow of PuzzleFixer and then introduce the details in each stage.

5.1 Usage Scenario

We demonstrate how PuzzleFixer can support an expert in fragment puzzle reassembly on the gargoyle dataset from prior work [21]. The dataset contains 30 3D segments with 98 fracture surfaces and is described by a point cloud with over 3.5 million data points.

Inspecting erroneous matches. When the expert is wearing the VR headset and controllers, PuzzleFixer immerses the expert in a museum environment and presents each fragment according to the position generated by automatic reassembly approaches (Fig. 5a). The expert quickly notices that the body (highlighted in yellow dotted box) is not well connected to the head and bottom of the gargoyle statue. To help experts find the specific mismatches, PuzzleFixer presents matches with a node-link diagram that shows matches between faces (I1). The match quality is encoded with a blue-to-green colormap, where a color closer to green indicates better quality. The expert then navigates to the matches between the body and other parts (Fig. 5 b1, b2) and identifies false matches by touching the corresponding edges (I2).

Exploring potential reassemblies. All fragment combinations that can be reassembled to the body are generated and clustered into several groups (Fig. 5c) (E2). Each group is represented by an averaged diagram that indicates the coarse positions and matches of fragments within the group. Averaged diagrams are scattered in front of users' view, where adjacent graphs exhibit similarities in positions and matches (E1). Shapes of the body are presented to visualize match positions of the closest diagram. The expert gradually shifts the averaged diagrams to the enlarged body shapes and rotates the shapes to observe the match position and angles from different perspectives. When averaged diagram is not ideal as expected, the expert further shifts the clusters in different directions to search for the others. Through scanning the diagrams, the expert focuses on the upper-right diagram since it exhibits a high match quality and matches all faces on the upper side of the body with regular positions. Thus, he selects the group by clicking the "trackpad" on the controller. Accordingly, PuzzleFixer presents all reassemblies in this group (Fig. 5d). Each reassembly contains fragments that directly match the body.

Confirming correct matches. After observing the reassemblies shown in Fig. 5d, the expert chooses one reassembly which contains head-recognizable fragments. Next, he grabs the fragments and attempts to align their matches (Fig. 5e). When the expert navigates to the faces, PuzzleFixer displays the overall shapes through a reassembly-in-miniature. The facing faces are instantly connected with hint edges to guide angle and position adjustment for fragment alignment (C1). Once the expert complete an alignment attempt, PuzzleFixer precisely aligns the interacted faces. Satisfied with the results, the expert ends the alignment operation. PuzzleFixer automatically reassembles the left fragments and matches and shows the final result (Fig. 5f) (C2).

5.2 Inspecting the reassembled object

At the beginning of the workflow, PuzzleFixer visualizes the reassembled object with relation data to help users identify mismatches.

Reassembly generation. Prior research on automatic reassembly has proposed different techniques for resembling objects in various shapes

and materials. According to the characteristics of the fragments we used in our case study, we adopt multiple computer graphics approaches to generate potential reassemble solutions.

- We obtain meshes from the 3D point cloud of fragments by screened poisson reconstruction [23]. Based on the meshes, we adopt a robust face and boundary extraction approach [63] to get the intact and fractured faces of each fragment.
- We apply pairwise matching algorithm [28] to calculate matching scores between all pairwise fragment matches.
- With the scores, we adopt multi-part reassembly [39] to automatically select the matches and use openGR [31] for alignment and construct a reassembly result.

Visualizing and identifying the mismatches. We develop *Embed Component* to visualize the reassembly with two types of data (I1). As shown in Fig. 5a, the match between a face pair is presented by a 3D tube that links between the centers of corresponding fragments. In particular, we represent an unmatched face by a tube that links between the face center to its fragment center. The score of each match is encoded by the color of its face, where the color closer to green indicates a higher score. Consequently, *Embed Component* integrates match relations with fragment shapes with a skeleton metaphor, where a node represents a fragment, and an edge denotes a match between a fragment pair. To reduce the visual clutter of edges, we support users to filter the edges by their scores or direct manipulations. The edge scores are normalized within 0 and 1. The users can define a threshold to filter the edges and focus on unqualified matches. In addition, users can filter edges by direct manipulations through the virtual hands, which allows the system to only show the interesting matches to reduce edge clutter and ease mismatch searching.

Based on the skeleton metaphor, PuzzleFixer provides three kinds of interactions to support mismatch identification and modification:

- **Object navigation.** PuzzleFixer provides grab and transform gestures that allow users to transform the entire object. The object and its edges will be rotated and shifted synchronously with hand transformation. Users can also freely move in the virtual space to observe the object from different perspectives.
- **Match modification.** *Face Component* leverages touch gesture to enable users to identify mismatches (I2). The touched edge will be replaced by two disconnected edges that represent the corresponding mismatched faces (Fig. 5b1, b2). Moreover, the user can touch an intact face to switch the material between translucent or opaque to focus on the edges or surface texture.
- **Target group selection.** When mismatches are specified, the object is divided into several interconnected fragment groups. PuzzleFixer allows users to select a target group via touching gesture. Next, the user can explore how the remaining groups match with it.

5.3 Exploring reassembly candidates

Based on the mismatches, PuzzleFixer first generates all possible candidates, then features *Layout Component* and *Cluster Component* to support overall reassembly pattern understanding and help users progressively find the correct reassembly, respectively.

Candidate generation. Given the groups $G = \{G_1, G_2, \dots, G_n\}$, PuzzleFixer generates matching candidates by permutating the mismatches between the target and other groups: $\binom{n_{G_i}}{k} \binom{n_{G'}}{k} k!$, where G_i is the target group and G' is the set of remaining groups other than G_i . n_{G_i} and $n_{G'}$ are the number of mismatches in G_i and G' respectively, $k = \min(n_{G_i}, n_{G'})$. We use the mean of pairwise matching scores to indicate the match quality of a candidate. Since the number of candidates grows combinatorially with the increase in the number of mismatches, we set up two rules to exclude impossible matches to ensure generation efficiency: 1) alignments of the permutation matches cannot be achieved by breaking down the other alignments within that group, and 2) fragments should not penetrate into another after alignment. These rules help to eliminate meaningless candidates before user exploration.

Layout of candidates. PuzzleFixer applies *Layout Component* to distribute candidates based on the similarity of their skeleton to facilitate match identification (E1). Specifically, *Layout Component* 1) calculates

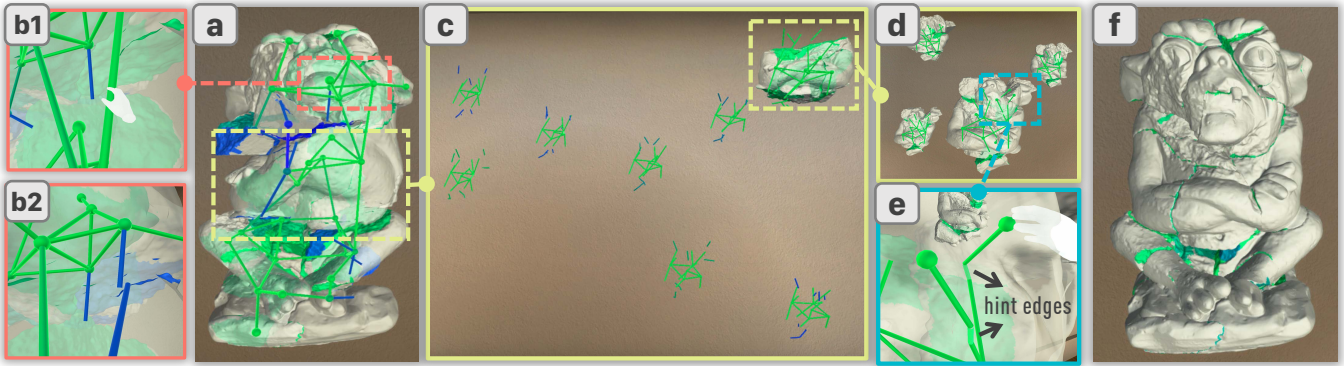


Fig. 5. Reassembling the gargoyle statue. (a) The fragments are automatically reassembled and visualized with the embedded node-link network. (b1, b2) The user modifies the error matches with hand gestures. Based on the modification, the possible matches are clustered and presented through summarized skeleton diagrams (c). Shapes of the reassemblies from the selected cluster are visualized in (d). In (e), the user manually aligns the faces. The other fragments are reassembled automatically, as shown in (f).

Chamfer distance [3, 47] between the nodes on the skeletons to represent similarities between candidates, and 2) performs dimensionality reduction using t-SNE [18, 57] to obtain candidate distribution on a 2D canvas. Previous studies [8, 30] have confirmed the efficiency of exploring 3D small multiples using a flat layout in an immersive setting. Following this method, we place canvas vertically in front of users. We leverage t-SNE to ensure the similarity between neighboring candidates and reveals patterns of the matched shapes.

Supported by *Layout Component*, PuzzleFixer uses the grab and transform gestures to achieve the following interactions at candidate level:

- *Candidate navigation*. After users grab the controller in the mid-air, all candidates are panned along with the canvas and rotated synchronously according to hand transform.
- *Candidate selection*. *Layout Component* zooms in on candidate to be selected to perform a focus by context layout. Users can select the enlarged candidate for further in-depth analysis.

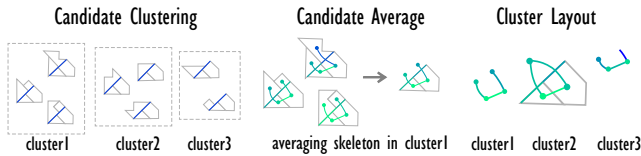


Fig. 6. The visual summary of candidates. Clusters are first identified by the layout position of each candidate and then visualized through the average skeleton. The cluster layout follows the rules from *Layout Component* and presents the target group shape of the cluster at the canvas center for focused observation.

Visual summary of candidates. After dimensionality reduction, candidate are represented by a scatterplot on a 2D plane. *Cluster Component* groups candidates and provide a level-of-detail searching approach to prevent the exploration from getting overwhelming (E2). Specifically, *Cluster Component* uses DBSCAN [13] to group candidates into clusters. We choose DBSCAN since it has been widely used to show the patterns of points after dimensionality reduction [15, 19, 45]. We perform two steps to generate visual summary of candidates (Fig. 6): 1) generate an average skeleton by averaging each fragment node in each candidate skeleton; 2) place the average skeleton at the mean position of the nodes it refers to. The score of each cluster is averaged and encoded by the corresponding skeleton color. Based on the cluster, *Cluster Component* supports candidate navigation by 1) categorizing all candidates based on the skeleton similarity to avoid visual overwhelming and 2) providing level-of-detail searching that enables users gradually find candidates according to coarse match relations and specific shape alignments. When users complete cluster selection, both the shape and skeleton of candidates in the clusters are visualized (Fig. 5d).

The visual summary of a cluster mainly incorporates three considerations: 1) *expressive* so that it shows the essential match evidence for

experts to determine the potentially correct ones, 2) *succinct* so that the presented evidence does not require excessive time for judgment, and 3) *representative* so that it presents the thumbnail of all the candidates in the cluster. To fulfill the expressive and representative criteria, we leverage the average skeleton to present the relation data and show the general form of the clustered candidates. The relative positions between the matching edges (i.e., edges which match the fragments to the target group) provide feedback on which faces are matched. The edge color indicates the quality of the alignment. For the succinctness, we avoid showing the surface of each cluster because it contains many detailed matching evidence (e.g., specific shape alignments and texture continuity) that requires much time for in-depth analysis of each candidate individually.

5.4 Confirming the matches

We design *Matching Component* and *Snapping Component* to assist with manual alignment (C1) and obtain precise reassemble results (C2).

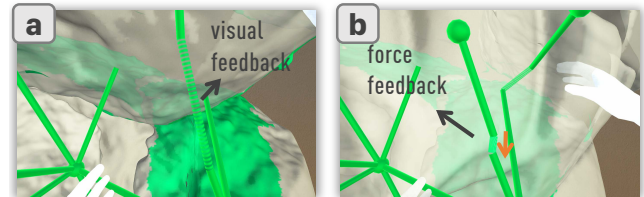


Fig. 7. The demonstration of two instant feedback. (a) visual feedback instantly shows a facing match. (b) force feedback adds the resistance (red arrow) along with the edge of a confirmed match.

Aligning fragments with instant feedback. PuzzleFixer allows users to align face pairs by grabbing inside the fragment shape and transforming the fragment with the controller. To guide the alignment, *Matching Component* provides two kinds of feedback (Figure 7): 1) visual feedback that instantly shows the aligning matches with a 3D dotted tube, which links faces that are currently close and facing each other and matches confirmed by previous alignment attempts; 2) the force feedback that exerts a resistance along with the edges of confirmed matches when these aligned faces are dragged away or the directions in which they are facing are changed. The visual feedback presents instant relations between the aligning faces, help users explore whether the matches are in agreement with their expectation. Instead of providing force feedback from the controller [2, 25], we use the change in the fragment transformation relative to the controller to create a force feedback cue. Specifically, as the user holds a fragment by grabbing the controller, the controller acts as a realistic reference to map its transformation directly onto the fragment without scaling. However, when the fragment is partially matched, those fragment transformations that lengthen the confirmed edges will be applied with a constant decreasing factor. Consequently, the user needs a larger transformation magnitude to move away from the prior matched faces until the distance between

the matched faces exceeds a threshold. This decreasing transformation mapping reduces the impact of misoperations (e.g., matches new faces but breaks previous ones), while creating a perception that the fragments are dragged by the edges.

Reassemble fragments with precise alignment. After the match is confirmed, *Snapping Component* uses a magnetic metaphor to automatically align the corresponding faces. We use the global registration technique [31] to perform a precise and fast alignment according to the meshes of matched face pairs. After all matches are confirmed, PuzzleFixer calls the automatic method to reassemble left mismatches and starts next iteration.

5.5 Implementation

The PuzzleFixer is implemented in a server-client architecture. The client provides an immersive environment for users to perform the reassembly workflow. We use Unity and HTC VIVE VR system with two hand controllers to set up the immersive environment and express the hand gestures. The server is built with Python for dimensionality reduction, clustering, and automatic matching methods. Specifically, we use scikit-learn [42] to generate clusters. These tasks are performed asynchronously to ensure interaction fluency. ZeroMQ [1] is applied to send and receive messages between the server and client.

6 EXPERT EVALUATION

We invited six domain experts to evaluate PuzzleFixer on a real-world cultural relics dataset. We report the study settings, two cases of reassembling process conducted by the experts, and the qualitative feedback and insights from the post-study expert interviews. We didn't perform quantitative analysis due to the limited sample size.

6.1 Study Settings

Dataset and preprocessing. The dataset is digitized from Ma Lin Broken Stele, engraved with calligraphy on its surface, and has high artistic and historical value. The dataset consists of ten broken stones of various sizes and in irregular shapes, with a total of 33 fracture surfaces. All the stone fragments are scanned and converted to Wavefront format files for 3D model presentation. We apply surface segmentation [63] and pairwise matching [28] to obtain an initial reassemble result with a matching structure and matching score.

Participants. We invited six experts (P1-P6; age: 23-52) from the university and museum to participate in our interview. One of the experts is a professor engaging in the field of restoration of digital artifacts. Another expert is a researcher who has experience in archaeology and cultural heritage protection for over 20 years. Other experts major in digital artifact modeling and restoration. All the experts have experience in manual reassembly in virtual reality. Only the professor participated in our previous workflow discussion.

Procedure. We conducted one-on-one interviews with the six experts. The interview included four parts: 1) We introduced the concepts of three-stage workflow and six components of PuzzleFixer, following a system demonstration to present the process of reassembly refinement (20 minutes). 2) After the introduction, we asked them to set up the VR headset and repeat the process in the demonstration to get familiar with the system (15 minutes). 3) When experts were confident in operating the system, we changed the dataset to the Ma Lin Broken Stele and encouraged the experts to reassemble the fragments (20 minutes). 4) The procedure concluded with a semi-structured interview, including the subjective perception, evaluation of workflow and each system component, and suggestions for improvement (30 minutes).

6.2 Case Studies

Case I: Finding Correct Matches from Series of Errors. This case produced by P1 focuses on identifying mismatches and searching for correct solutions.

Inspecting the reassembly fragments. After importing the dataset, the expert glanced at the reassembled object and quickly noticed the discordant fragment in the upper right area through the mismatched edges and the blue fractured faces (Fig. 8a). The skeleton showed that

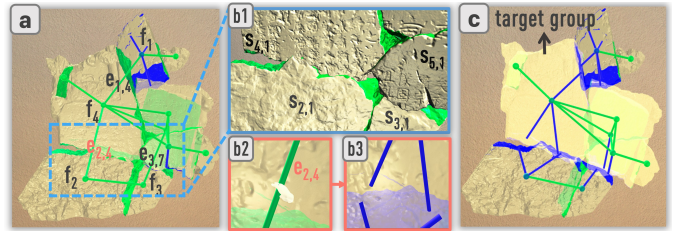


Fig. 8. Identify the mismatches. (a) The skeleton reveals the mismatches from f_1 . (b1) The faces show $s_{4,1}$ and $s_{5,1}$ are inscribed faces, which are not matched with $s_{2,1}$ and $s_{3,1}$. (b2-b3) the mismatch $e_{2,4}$ is disconnected through touch gestures. (c) a target group is selected.

there were several mismatched faces on f_1 . Thus, he dragged the object closer to disconnect $e_{1,4}$ and navigated to the bottom half of the stele (highlighted in the blue dotted box) for detailed observation. Through the inscription letters and the decorations on the surfaces (Fig. 8 b1), the expert found that their faces ($s_{2,1}$ and $s_{3,1}$) were discontinuous with their adjacencies ($s_{4,1}$ and $s_{5,1}$). Thus, he considered edges $e_{2,4}$ and $e_{3,7}$ were incorrect and disconnected them accordingly (Fig. 8 b2, b3). After identifying these error matches, he chose the largest group (Fig. 8c) as the target group and entered the exploration phase to see the possible solutions.

Exploring the candidates. As shown in Fig. 9 a1, the system presented the clusters of reassembly results in a plane canvas. The expert noticed that the overall results showed two forms: all the fragments were concentrated below the target (upper area of the canvas) or matched at the lower and the upper right area, respectively. Since the initial result was presented in the latter form, the expert was curious about what all the other fragments matched below would look like. Through scanning those clusters, he noticed that all the mismatched edges at the bottom of the skeleton sk_1 had corresponding edges with reasonable matching angles. Hence he selected it for further exploration.

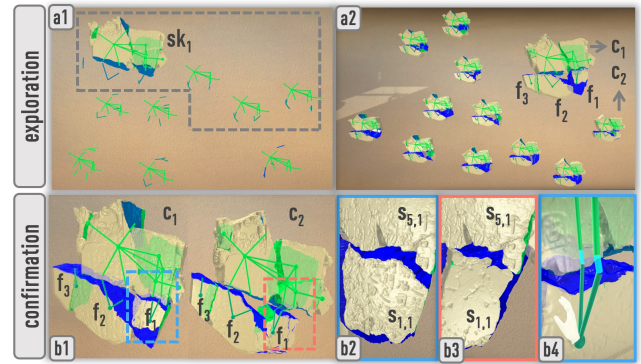


Fig. 9. Explore and confirm the matches. (a1) Clusters show two patterns of the matching location. (a2) Candidates present the result with shapes. (b1) Two candidates selected by the expert. (b2-b3) Decorations on the surface show the continuity of the neighbor fragments. (b4) Align the matches by the grab and transform gesture.

As shown in Fig. 9 a2, the candidates in this category matched f_2 and f_3 to the same position while enumerating a variety of matching possibilities for f_1 . After panning and rotating the candidates, the expert quickly ruled out candidates whose appearances were obviously incorrect and selected candidates c_1 and c_2 to explore their details.

Confirming the matches. The filtered candidates were placed side-by-side (Fig. 9 b1). To identify the correct one, the expert hid the skeleton and focused on the surface around the matching regions. He recognized that the decoration on faces $s_{5,1}$ and $s_{1,1}$ was continuous in c_1 (Fig. 9 b2) but not in c_2 (Fig. 9 b3). Hence, he confirmed that f_1 and f_5 in c_1 were matched correctly and refined their alignment following the visual feedback (Fig. 9 b4). After the expert finished the alignment refinement, PuzzleFixer automatically aligned the remaining fragments and entered the inspection stage in the next iteration.

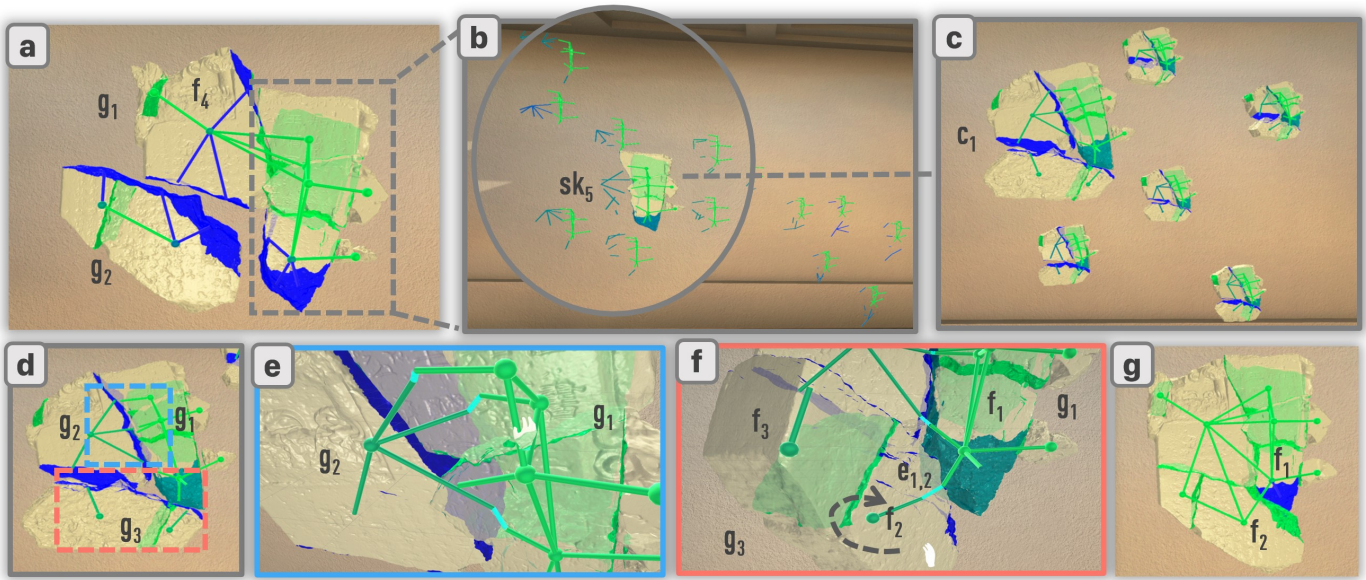


Fig. 10. Finishing the stele reassembly. (a) The skeleton and shapes shows two separate groups. (b) The clusters reveal that other fragments can be matched at bottom or left side of the target group. (c) The candidate in the selected cluster has the same appearance but different match relations. (d) The selected candidate has three groups. (e) Visual cues are shown in real-time when transforming the fragment groups. (f) Force cue is added on the $e_{1,2}$ to help rotate the g_3 to align f_3 . (g) The reassembled stele after finishing the workflow.

Case II: Assembling Fragments with Manual Matching. In this case, P3 completed the refinement mainly based on the exploration and confirmation stage.

Finding the potential matching shapes. At the beginning, the object was composed by two separate groups, g_1 and g_2 (Fig. 10a). The expert noted that the right part of g_1 seemed neat, while the fragment f_4 had several mismatched edges. Thus, he disconnected f_4 from g_1 and selected the remaining right part as the target to explore the matching opportunities. As shown in Fig. 10b, the *Layout Component* showed two general matching styles: the clusters inside the gray circle tended to concentrate the matching position on the left surface of the target group, while the rest matched more fragments at the bottom of the target group. The expert was interested in how to match the target from the left. Thus he focused on the candidates in the circle and chose cluster sk_5 , where all the left-mismatched edges in the target had corresponding matches. The selected candidates showed that their shapes were similar, and they only differed in the correspondence of the faces (Fig. 10c). Therefore, by default, he selected c_1 with the highest matching score to manually align the matches.

Correcting the matches with transform level interactions. As shown in Fig. 10d, the candidate contained three groups. The expert started with the match between g_1 and g_2 . By grabbing the group, the matching cues were visualized to inform the instant matching face and indicate how to transform the fragments (Fig. 10e). Thus, he aligned the matches by dragging g_1 according to the cues to refine the alignment. Next, to refine the alignment of g_3 , the expert observed the details on the surface and realized that g_3 was matched incorrectly. Hence, he decided to turn g_3 around for alignment. To this end, he first confirmed the match $e_{1,2}$ by dragging f_2 closer to f_1 along the matching cue line (Fig. 10f). To finish the alignment for f_3 , he rotated g_3 around the aligned edge ($e_{1,2}$). The force feedback was imposed along the matched edge to ease the rotation by preventing tilting or stretching the edge. When he finished fragment transforming and released the grab, *Snapping Component* immediately aligned f_1 and f_2 together. After alignment, the expert was satisfied with the match (Fig. 10g) and reviewed the result: the stele was relatively complete, while some fragments were missing between f_1 and f_2 in this stele dataset.

6.3 Observations and Expert Feedback

Time performance and usability. All participants could successfully correct the reassembly with a maximum of four iterations. The average

completion time of the participants was 14.3 minutes (1.1 min, 1.2 min, and 1.4 min in each stage, respectively). Note that we see these times as qualitative evidences due to the limited sample size. We found that the participants did not take much time during the exploration stage as they tended to roughly select several candidates rather than one for subsequent in-depth analysis. The confirmation stage is time-consuming as most of the participants needed to work back and forth to confirm the alignment correctness when transforming the fragments. Nevertheless, the participants were able to refine several mismatches within an object in one iteration within a few minutes. As for the usability, the participants thought that the system was easy to use and the interaction was intuitive, suggesting that “*the graph modification of VR operation is not difficult*” (P1) and “*these operations are just a matter of habit*” (P5). Some experts criticized that the system should be improved by adding, for example, “*more remarks to explain the operations*” (P2) and “*fallback function to reselect the candidates*” (P3). We take these suggestions for future improvements.

Refine the reassembly based on suggestions. We were interested in the comments about integrating automatic reassemble results into manual reassembly refinement. Overall, the experts confirmed the usefulness of showing potential reassemblies instead of trying to find these solutions manually by trial-and-error. This is because that automatically generated solutions “*can show coarse results for me to evaluate the matching correctness*” (P1, P2) and “*provide guidance for manually aligning fragments*” (P4-P6). We observed in our case study that although participants can freely align fragments in the confirmation stage, they prefer to make a minor modification to the generated reassembly in most cases. A main reason is that fragment alignment requires users to navigate and observe details while controlling fragment transformation with mid-air interactions. A low-quality candidate may cause users to make multiple alignment attempts, which is prone to arm fatigue [51, 55]. If satisfying candidates were not filtered out, the participants would tend to find and reselect good candidates in next iteration, instead of aligning a worse candidate. This finding suggests that experts’ interactions tend to rely on automatically generated results.

Assist relation understanding through network embedding. All experts spoke highly of the network design and showed enthusiasm for observing the matching results at the inspection stage. P4 thought the network was intuitive and helpful in enhancing understanding of the current matching structure. For example, “*It intuitively shows match relations*” (P3) and “*I can intuitively see whether there is a match*

relationship between key faces” (P2). All experts confirmed that they could find erroneous matches through visualization of reassembled objects. Nonetheless, we observed that users still spent much time on navigation. For example, our participants frequently walked around to observe an object from different perspectives and switched view between the global appearance and detailed faces. Despite that the experts believe this real-world-like navigation is intuitive and useful, its efficiency might be affected by structural complexity of objects. However, how to improve the navigation of 3D structures with complex fragment shapes and relations remains unclear.

Improve search efficiency with the level of detail selection. The experts highlighted the effectiveness of exploring solutions with two-level filtering. They thought the clusters showed the matching cues could eliminate a large number of mismatches and guide subsequent selections. P5 mentioned, “when I observed this cluster, I thought it would be correct if edges were rotated at the bottom a little bit. So I will focus on these intended matches in subsequent searches.” Although P6 thought that selecting the optimal reassembly takes time due to complexity of the shapes and the large number of reassembling possibilities, he thought the layout of each reassembly would improve the search efficiency since it “provides the trend of matches.” While adding more visual hints can further ease the searching process, we leave this for future improvements to avoid overwhelming users in the current implementation.

Reduce interaction laborious with instant feedback. When asked how instant feedback could assist manual alignment in the confirmation stage, experts reported that instant feedback immensely helped increase interaction efficiency because “I instantly know which faces are currently being aligned.” (P5) P1 praised the force feedback as he said “It limits incorrect fragment transformation” and “helps alleviate mis-operations.” P6 mentioned that the force feedback allowed him to align multiple faces sequentially instead of all of them at once, since the force added on aligned matches “guides me to align remaining faces.”

Suggestions. The experts in our case study pointed out several limitations of our system. First, some fractured surfaces were not detected in preprocessing. To this end, area selection tools should be provided, such as the lasso of mesh surfaces, to help users manually add or modify areas of fractured surfaces. Second, automatic alignment from *Snapting Component* was sometimes inaccurate, which affects subsequent matching of surrounding fragments. However, many state-of-the-art point cloud registration algorithms [17, 38] cannot perform the mesh alignment within response time or suffer from limitations in alignment of arbitrary and damaged surface pairs. To solve these problems, we further exploit a precise mode of direct interaction [50] for manual alignment improvement. This interaction allows users to gradually adjust the fragment transformation from one DOF in turn. Third, the system currently does not support fragment annotation, which easily leads to confusion between fragments with a similar appearance.

7 DISCUSSION

Significance—incorporating human insights into solving the NP-hard searching problem. Restoration of important objects is a prerequisite to manifest their value. It’s challenging to find a unique solution from numerous reassembly possibilities due to the huge reassembly space, which causes a NP-hard problem. While various automatic approaches have been proposed to solve the problem, the result fails when objects are diverse, incomplete, and low-quality [20]. We explore this direction by incorporating human insights in facilitating the reassembly searching. We further developed PuzzleFixer that visualizes 3D fragments with relational networks and provides instant feedback to facilitate manual reassembly. Our system is evaluated through case studies on real cultural relics, which shows the usefulness and efficiency in reassembly refinement in real-life scenarios.

Generalizability—from cultural relic reassembly to educational demonstration. Although PuzzleFixer is evaluated on thick stele blocks, the reassembled objects can be extended to types of data such as thin-shell pottery and shredded paper in manuscripts or images. Since our reassembly workflow is independent of automatic matching

method, various types of data can be imported to our system once corresponding method is well prepared. Our reassembly environment allows users to immerse themselves in a virtual setting and experience object reassembly, which can be used for cultural relic restoration and educational purposes. These adaptations allow our system to be used in other scenarios, such as experiencing cultural heritage protection [14,27] and understanding the complex anatomical structure [44,62,67].

Potentiality—towards more efficient physical reassembly. Reassembling fragments in virtual reality generates reassembly steps and offers guidance for restorers when conducting the physical reconstructions [7,35]. However, in practice, restorers need real-time guidance that provides in-situ aligning guidance while operating fragment reassembly. Guiding physical reassembly with a traditional 2D diagram cannot instantly show reassembling cues regarding the current perspective of restorers and fragment situation. While AR has prospects in crafting and objects assembly by embedding instructions into the physical context [59,60,70,71], AR-assisted physical reassembly in-situ raises new challenges such as reassembly simulation, planning, and designing of collaborative interaction [52,61]. We hope our exploration will inspire future work in this direction.

Study limitations. Although PuzzleFixer leverages levels of selection to help users find solutions efficiently with tens of fragments, identifying the optimal reassembly from tens of thousands of candidates is not supported due to the heavy burden of candidate generation and exploration. To avoid such a situation, we limit the number of mismatches and fragment groups specified at the inspection stage to no more than 10 and 5, respectively. The other wrong matches inside each group can be further corrected in next iteration. We also select the top- k clusters or candidates with the highest average matching score for presentation. A key observation is that the candidates with a low average score are usually incorrect. We set k to 15 in our case study, considering the complexity of the shape and structure. Besides, we haven’t considered the situation where the automatic approaches completely fail to reassemble the fragments into a single object. In this case, it is difficult for automatic approaches to provide potentially correct candidates. This makes it difficult for users to take advantage of automatically generated solutions, and fragment refinements can only be performed manually. Finally, our system has considered only two types of data, i.e., the shape data and the relation data. More rich contexts, such as the physical location of each fragment and classification information [64], can also promote experts’ judgment and objects reassembly. However, these new types of data introduce challenges for new visualization designs, which naturally require further studies. We plan to open source the manual interaction part of our system to the community¹ and involve more information to facilitate reassembly refinement in the future.

8 CONCLUSION

In this work, we present a virtual reassembly system, PuzzleFixer, to help experts refine the reassembled object in an immersive environment. Given a pre-assembled object, our system provides a three-stage workflow for users to inspect the mismatches, explore potential solutions for reassembly, and confirm the alignments. To facilitate mismatch inspection, PuzzleFixer visualizes 3D fragments with an embedded network to augment the fragment shapes with match relations. Once users identify the mismatches, PuzzleFixer generates and clusters potential solutions to provide level-of-detail explorations to facilitate reassembly searching. Moreover, we design visual and force feedback to improve manual alignment. We evaluate our system by conducting case studies on real cultural relics fragments. Insights and expert feedback illustrate the usefulness and effectiveness of the system and enlighten future research opportunities.

ACKNOWLEDGMENTS

The work was supported by NSFC (62072400), the National Key Research and Development Program of China (No.2019YFC1521102), and the Collaborative Innovation Center of Artificial Intelligence by MOE and Zhejiang Provincial Government (ZJU).

¹<https://github.com/PuzzleFixer/PuzzleFixer>

REFERENCES

- [1] ZeroMQ. <https://zeromq.org/>. Accessed: 2021-09-08.
- [2] A. Achberger, F. Heyen, K. Vidackovic, and M. Sedlmair. Touching data with PropellerHand. *Journal of Visualization*, pp. 1–16, 2022.
- [3] H. G. Barrow, J. M. Tenenbaum, R. C. Bolles, and H. C. Wolf. Parametric Correspondence and Chamfer Matching: Two New Techniques for Image Matching. In *Proceedings of the 5th International Joint Conference on Artificial Intelligence - Volume 2, IJCAI'77*, p. 659–663. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1977.
- [4] D. Bridger, D. Danon, and A. Tal. Solving Jigsaw Puzzles With Eroded Boundaries. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3523–3532. IEEE Computer Society, jun 2020.
- [5] B. Brown, L. Laken, P. Dutré, L. Van Gool, S. Rusinkiewicz, and T. Weyrich. Tools for virtual reassembly of fresco fragments. *International journal of heritage in the digital era*, 1(2):313–329, 2012.
- [6] N. Capece, U. Erra, M. Gruosso, and M. Anastasio. Archaeo Puzzle: An Educational Game Using Natural User Interface for Historical Artifacts. In M. Spagnuolo and F. J. Melero, eds., *Eurographics Workshop on Graphics and Cultural Heritage*. The Eurographics Association, 2020.
- [7] P. Carlson, A. Peters, S. B. Gilbert, J. M. Vance, and A. Luse. Virtual Training: Learning Transfer of Assembly Tasks. *IEEE Transactions on Visualization and Computer Graphics*, 21(6):770–782, 2015.
- [8] X. Chu, X. Xie, S. Ye, H. Lu, H. Xiao, Z. Yuan, C. Zhu-Tian, H. Zhang, and Y. Wu. TIVEE: Visual Exploration and Explanation of Badminton Tactics in Immersive Visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):118–128, 2022.
- [9] R. de Lima-Hernandez and M. Vergauwen. A Hybrid Approach to Reassemble Ancient Decorated Block Fragments through a 3D Puzzling Engine. *Remote Sensing*, 12(16), 2020.
- [10] A. Deever and A. Gallagher. Semi-automatic assembly of real cross-cut shredded documents. In *2012 19th IEEE International Conference on Image Processing*, pp. 233–236, 2012.
- [11] E. D. Demaine and M. L. Demaine. Jigsaw Puzzles, Edge Matching, and Polyomino Packing: Connections and Complexity. *Graphs and Combinatorics*, 23(1):195–208, feb 2007.
- [12] N. Derech, A. Tal, and I. Shimshoni. Solving archaeological puzzles. *Pattern Recognition*, 119:108065, 2021.
- [13] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, KDD'96*, p. 226–231. AAAI Press, 1996.
- [14] X. Fu, Y. Zhu, Z. Xiao, Y. Xu, and X. Ma. *RestoreVR: Generating Embodied Knowledge and Situated Experience of Dunhuang Mural Conservation via Interactive Virtual Reality*, p. 1–13. Association for Computing Machinery, 2020.
- [15] T. Fujiwara, O. Kwon, and K. Ma. Supporting Analysis of Dimensionality Reduction Results with Contrastive Learning. *IEEE Transactions on Visualization and Computer Graphics*, 26(01):45–55, jan 2020.
- [16] T. Funkhouser, H. Shin, C. Toler-Franklin, A. G. Castañeda, B. Brown, D. Dobkin, S. Rusinkiewicz, and T. Weyrich. Learning How to Match Fresco Fragments. *Journal on Computing and Cultural Heritage*, 4(2), nov 2011.
- [17] Z. Gojcic, C. Zhou, J. D. Wegner, and W. Andreas. The Perfect Match: 3D Point Cloud Matching with Smoothed Densities. In *International conference on computer vision and pattern recognition (CVPR)*, pp. 5545–5554, 2019.
- [18] R. Gove, L. Cadalzo, N. Leiby, J. M. Singer, and A. Zaitzeff. New guidance for using t-SNE: Alternative defaults, hyperparameter selection automation, and comparative evaluation. *Visual Informatics*, 2022.
- [19] J. Han, J. Tao, and C. Wang. FlowNet: A Deep Learning Framework for Clustering and Selection of Streamlines and Stream Surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 26(4):1732–1744, 2020.
- [20] R. D. L. Hernandez, S. Vincke, M. Bassier, L. Mattheuwsen, J. Derdeale, and M. Vergauwen. Puzzling Engine: a Digital Platform to Aid the Reassembling of Fractured Fragments. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 4215:563–570, Aug. 2019.
- [21] Q.-X. Huang, S. Flöry, N. Gelfand, M. Hofer, and H. Pottmann. Reassembling Fractured Objects by Geometric Matching. *ACM Transactions on Graphics*, 25(3):569–578, jul 2006.
- [22] M. Jurda, P. Urbanová, and J. Chmelík. Digital restoration of fragmentary human skeletal remains: Testing the feasibility of virtual reality. *Journal of forensic and legal medicine*, 66:50–57, 2019.
- [23] M. Kazhdan and H. Hoppe. Screened Poisson Surface Reconstruction. *ACM Transactions on Graphics*, 32(3), jul 2013.
- [24] C. Le and X. Li. JigsawNet: Shredded Image Reassembly Using Convolutional Neural Network and Loop-Based Composition. *IEEE Transactions on Image Processing*, 28(8):4000–4015, 2019.
- [25] A. Lécuyer, S. Coquillart, A. Kheddar, P. Richard, and P. Coiffet. Pseudo-haptic feedback: can isometric input devices simulate force feedback? In *Proceedings IEEE Virtual Reality 2000 (Cat. No.00CB37048)*, pp. 83–90, 2000.
- [26] H. C. Lee, T. Palmbach, and M. T. Miller. *Henry Lee's crime scene handbook*. Academic Press, 2001.
- [27] M. Li, Y. Wang, and Y.-Q. Xu. Computing for Chinese Cultural Heritage. *Visual Informatics*, 6(1):1–13, 2022.
- [28] Q. Li, G. Geng, and M. Zhou. Pairwise Matching for 3D Fragment Reassembly Based on Boundary Curves and Concave-Convex Patches. *IEEE Access*, 8:6153–6161, 2020.
- [29] T. Lin, R. Singh, Y. Yang, C. Nobre, J. Beyer, M. A. Smith, and H. Pfister. Towards an Understanding of Situated AR Visualization for Basketball Free-Throw Training. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems, CHI '21*. Association for Computing Machinery, 2021.
- [30] J. Liu, A. Prouzeau, B. Ens, and T. Dwyer. Design and Evaluation of Interactive Small Multiples Data Visualisation in Immersive Spaces. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 588–597, 2020.
- [31] N. Mellado. OpenGR: a 3D Global Registration Library. <https://github.com/STORM-IRIT/OpenGR>, 2018.
- [32] N. Mellado, D. Aiger, and N. J. Mitra. Super 4PCS Fast Global Pointcloud Registration via Smart Indexing. *Computer Graphics Forum*, 33(5):205–215, 2014.
- [33] N. Mellado, P. Reuter, and C. Schlick. Semi-Automatic Geometry-Driven Reassembly of Fractured Archeological Objects. In *Proceedings of the 11th International Conference on Virtual Reality, Archaeology and Cultural Heritage, VAST'10*, p. 33–38. Eurographics Association, Goslar, DEU, 2010.
- [34] D. Mendes, F. Relvas, A. Ferreira, and J. Jorge. The Benefits of DOF Separation in Mid-Air 3D Object Manipulation. In *Proceedings of the 22nd ACM Conference on Virtual Reality Software and Technology, VRST '16*, p. 261–268, 2016.
- [35] M. Murcia-López and A. Steed. A Comparison of Virtual and Physical Training Transfer of Bimanual Assembly Tasks. *IEEE Transactions on Visualization and Computer Graphics*, 24(4):1574–1583, 2018.
- [36] P. Olsson, F. Nysjö, J.-M. Hirsch, and I. Carlbom. Snap-to-fit, a haptic 6 DOF alignment tool for virtual assembly. In *2013 World Haptics Conference (WHC)*, pp. 205–210. IEEE, 2013.
- [37] G. Palmas, N. Pietroni, P. Cignoni, and R. Scopigno. A computer-assisted constraint-based system for assembling fragmented objects. In *2013 Digital Heritage International Congress (DigitalHeritage)*, vol. 1, pp. 529–536, 2013.
- [38] Y. Pan, B. Yang, F. Liang, and Z. Dong. Iterative Global Similarity Points : A robust coarse-to-fine integration solution for pairwise 3D point cloud registration. In *2018 International Conference on 3D Vision (3DV)*, pp. 180–189. IEEE, 2018.
- [39] G. Papaioannou, T. Schreck, A. Andreadis, P. Mavridis, R. Gregor, I. Sipi-ran, and K. Vardis. From Reassembly to Object Completion: A Complete Systems Pipeline. *Journal on Computing and Cultural Heritage*, 10(2), mar 2017.
- [40] M.-M. Paumard, D. Picard, and H. Tabia. Image reassembly combining deep learning and shortest path problem. In V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, eds., *Computer Vision – ECCV 2018*, pp. 155–169. Springer International Publishing, 2018.
- [41] M.-M. Paumard, D. Picard, and H. Tabia. Deepzzle: Solving Visual Jigsaw Puzzles With Deep Learning and Shortest Path Optimization. *IEEE Transactions on Image Processing*, 29:3569–3581, jan 2020.
- [42] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [43] R. Pintus, K. Pal, Y. Yang, T. Weyrich, E. Gobbetti, and H. Rushmeier. A

- Survey of Geometric Analysis in Cultural Heritage. In *Computer Graphics Forum*, vol. 35, pp. 4–31. Wiley Online Library, 2016.
- [44] D. Pohlandt, B. Preim, and P. Saalfeld. Supporting Anatomy Education with a 3D Puzzle in a VR Environment - Results from a Pilot Study. In *Proceedings of Mensch Und Computer 2019*, p. 91–102. Association for Computing Machinery, 2019.
- [45] J. Pu, H. Shao, B. Gao, Z. Zhu, Y. Zhu, Y. Rao, and Y. Xiang. matExplorer: Visual Exploration on Predicting Ionic Conductivity for Solid-state Electrolytes. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):65–75, 2022.
- [46] R. Radkowski and C. Stritzke. Interactive Hand Gesture-based Assembly for Augmented Reality Applications. In *Proceedings of the 2012 International Conference on Advances in Computer-Human Interactions*, pp. 303–308. Citeseer, 2012.
- [47] N. Ravi, J. Reizenstein, D. Novotny, T. Gordon, W.-Y. Lo, J. Johnson, and G. Gkioxari. Accelerating 3D Deep Learning with PyTorch3D. *arXiv:2007.08501*, 2020.
- [48] P. Reuter, G. Riviere, N. Couture, S. Mahut, and L. Espinasse. ArcheoTUI—Driving Virtual Reassemblies with Tangible 3D Interaction. *Journal on Computing and Cultural Heritage*, 3(2), oct 2010.
- [49] S. Rusinkiewicz and M. Levoy. Efficient Variants of the ICP Algorithm. In *Proceedings Third International Conference on 3-D Digital Imaging and Modeling*, pp. 145–152, 2001.
- [50] P. Saalfeld, C. Böttcher, F. Klink, and B. Preim. VR System for the Restoration of Broken Cultural Artifacts on the Example of a Funerary Monument. In *2021 IEEE Virtual Reality and 3D User Interfaces (VR)*, pp. 739–748. IEEE, 2021.
- [51] K. A. Satriadi, B. Ens, M. Cordeil, B. Jenny, T. Czauderna, and W. Willett. Augmented Reality Map Navigation with Freehand Gestures. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 593–603, 2019.
- [52] M. Sereno, X. Wang, L. Besancon, M. J. MCGuffin, and T. Isenberg. Collaborative Work in Augmented Reality: A Survey. *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–1, 2020.
- [53] E. Sizikova and T. Funkhouser. Wall Painting Reconstruction Using a Genetic Algorithm. *Journal on Computing and Cultural Heritage*, 11(1), dec 2017.
- [54] J. Sreng, A. Lecuyer, C. Megard, and C. Andriot. Using Visual Cues of Contact to Improve Interactive Manipulation of Virtual Objects in Industrial Assembly/Maintenance Simulations. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):1013–1020, 2006.
- [55] C. Su, C. Yang, Y. Chen, F. Wang, F. Wang, Y. Wu, and X. Zhang. Natural multimodal interaction in immersive flow visualization. *Visual Informatics*, 5(4):56–66, 2021.
- [56] B. H. Thomas, G. F. Welch, P. Dragicevic, N. Elmqvist, P. Irani, Y. Jansen, D. Schmalstieg, A. Tabard, N. A. M. ElSayed, R. T. Smith, and W. Willett. *Situated Analytics*, pp. 185–220. Springer International Publishing, Cham, 2018.
- [57] L. Van der Maaten and G. Hinton. Visualizing Data using t-SNE. *Journal of machine learning research*, 9(11), 2008.
- [58] L. Wang, J. Pan, and Q. Yao. Virtual Reassembly of Fractured Bones for Orthopedic Surgery. In *2018 International Conference on Virtual Reality and Visualization (ICVRV)*, pp. 20–27, 2018.
- [59] X. Wang, S. K. Ong, and A. Y. Nee. A comprehensive survey of augmented reality assembly research. *Advances in Manufacturing*, 4(1):1–22, 2016.
- [60] M. Weiß, K. Angerbauer, A. Voit, M. Schwarzl, M. Sedlmair, and S. Mayer. Revisited: Comparison of Empirical Methods to Evaluate Visualizations Supporting Crafting and Assembly Purposes. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):1204–1213, 2021.
- [61] J. Woodward and J. Ruiz. Analytic Review of Using Augmented Reality for Situational Awareness. *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–1, 2022.
- [62] C. Xu, G. Sun, and R. Liang. A survey of volume visualization techniques for feature enhancement. *Visual Informatics*, 5(3):70–81, 2021.
- [63] J. Xu, M. Zhou, Z. Wu, W. Shui, and S. Ali. Robust surface segmentation and edge feature lines extraction from fractured fragments of relics. *Journal of Computational Design and Engineering*, 2(2):79–87, 01 2015.
- [64] K. Yang, X. Cao, G. Geng, K. Li, and M. Zhou. Classification of 3D terracotta warriors fragments based on geospatial and texture information. *Journal of Visualization*, 24(2):251–259, 2021.
- [65] W. Yang, Z. Mingquan, Z. Pengfei, and G. Guohua. Matching Method of Cultural Relic Fragments Constrained by Thickness and Contour Feature. *IEEE Access*, 8:25892–25904, 2020.
- [66] S. Ye, C. Zhu-Tian, X. Chu, Y. Wang, S. Fu, L. Shen, K. Zhou, and Y. Wu. ShuttleSpace: Exploring and Analyzing Movement Trajectory in Immersive Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):860–869, 2021.
- [67] L. Yu, J. Ouwering, P. Svetachov, P. van Ooijen, J. Kosinka, et al. VeLight: A 3D virtual reality tool for CT-based anatomy teaching and training. *Journal of Visualization*, 25(2):293–306, 2022.
- [68] K. Zhang, W. Yu, M. Manhein, W. Waggenspack, and X. Li. 3D Fragment Reassembly Using Integrated Template Guidance and Fracture-Region Matching. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 2138–2146, 2015.
- [69] Y. Zhao, J. Jiang, Y. Chen, R. Liu, Y. Yang, X. Xue, and S. Chen. Meta-verse: Perspectives from graphics, interactions and visualization. *Visual Informatics*, 2022.
- [70] C. Zhu-Tian, Y. Su, Y. Wang, Q. Wang, H. Qu, and Y. Wu. MARVisT: Authoring Glyph-Based Visualization in Mobile Augmented Reality. *IEEE Transactions on Visualization and Computer Graphics*, 26(8):2645–2658, 2020.
- [71] C. Zhu-Tian, W. Tong, Q. Wang, B. Bach, and H. Qu. Augmenting Static Visualizations with PapARVis Designer. In *CHI '20: CHI Conference on Human Factors in Computing Systems, Honolulu, HI, USA, April 25-30, 2020*, pp. 1–12. ACM, 2020.
- [72] C. Zhu-Tian, Y. Wang, T. Sun, X. Gao, W. Chen, Z. Pan, H. Qu, and Y. Wu. Exploring the design space of immersive urban analytics. *Visual Informatics*, 1(2):132–142, 2017.
- [73] C. Zhu-Tian, W. Zeng, Z. Yang, L. Yu, C. Fu, and H. Qu. LassoNet: Deep Lasso-Selection of 3D Point Clouds. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):195–204, 2020.