EUROGRAPHICS Workshop on Graphics and Cultural Heritage (2024)
M. Corsini, D. Ferdani, A. Kuijper, and H. Kutlu (Editors)

Computer-Assisted Collaborative Fragment Matching of Incomplete Stone Artifacts

P. Houska[1], A. Masur[1], S. Kloiber[1], S. Lengauer[1], S. Karl[2], R. Preiner[1]

[1]Graz University of Technology, Institute of Computer Graphics and Knowledge Visualisation, Austria
[2]University of Graz, Institute of Classics, Austria

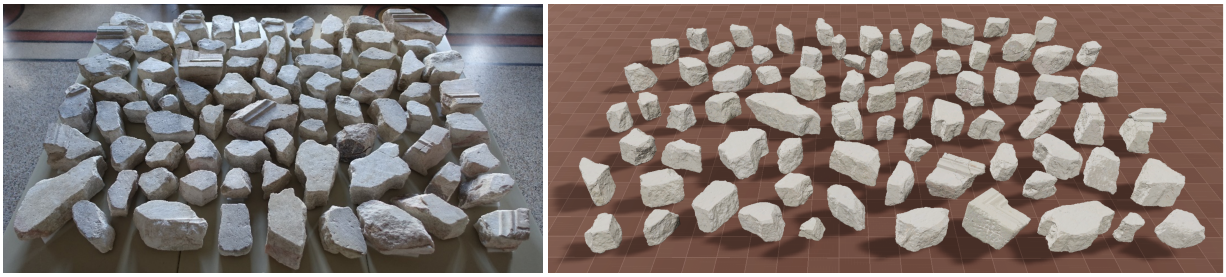


Figure 1: *Stone fragments of an early Christian marble slab from the 5th century CE (left). Digital twins rendered in an interactive web application for collaborative assembly by citizen scientists (right).*

Abstract

Archaeological artifacts are often only preserved in fragments. Their reassembly is thus a common task for conservators and archaeologists. Unfortunately, this reassembly process is anything but trivial. Major complicating factors are given if fragments are (1) eroded and weathered, (2) incomplete and missing, and (3) expose little to no geometric surface features or surface-color variations. Artifacts made of white marble, which was generally used in antiquity, pose an additional challenge for reassembly due to (4) their weight and (5) the fragility of the crystalline material. This holds for both freestanding marble sculptures and relief stones or otherwise decorated slabs. In this work, we focus on the reassembly problem of marble slabs with a flat surface, for which fragments are pre-oriented such that their top faces point upward. While computer-based reassembly algorithms exist, they are typically tuned for specific types of artifacts and rely on joining fragments based on a combination of a geometric and surface-color correspondence. Hence, methods that reassemble broken and deteriorated white marble slabs typically perform worse than the respective domain experts. Due to the foreseeable shortcomings of reassembly algorithms when applied to eroded, incomplete, and widely feature-less white marble fragments, we refrain from designing an algorithm for the specific problem at hand. Instead, we incorporate a large number of users from the broad public in the reassembly process. To that end, we provide an intuitive interactive web platform that allows users access to 3D digital twins of the Christian marble-slab fragments. Transferring the fragments to the virtual domain also enables the introduction of algorithmic automatisms to assist the users in the process. In this paper, we present the system design and the provided basic geometric automatisms, and report on their efficiency and utilization in an ongoing large-scale citizen-science experiment that involves several thousand users.

CCS Concepts

• ***Human-centered computing*** → ***Collaborative and social computing systems and tools;*** • ***Applied computing*** → ***Archaeology;*** • ***Computing methodologies*** → *Point-based models;*

1. Introduction

Archaeological artifacts are typically excavated in a fragmented state. Hence, an important part of finds processing is to put the fragments together to their original holistic form, thereby enabling a better assessment, evaluation, and documentation of the artifact in question. The difficulty of reassembling archaeological finds depends on several factors, such as the number of fragments and whether fragments of an artifact are missing, the degree of fragment erosion and weathering, and whether surface colors and geometric

© 2024 The Authors.
Proceedings published by Eurographics - The European Association for Computer Graphics.
This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

features are present, as these can give distinctive clues for fragment matching.

The otherwise laborious and isolated work of a typically small group of archaeologists and conservators, that have access to the physical fragments of an archaeological find, can be facilitated by digitizing the fragments and then reassembling the model virtually. Digitization not only opens up the dataset to a broader audience than just the domain experts, but also prevents further damage that otherwise happens in the process by merely touching and placing match candidates close to each other many times. Furthermore, working in the virtual domain enables a variety of possibilities for interactive inspection and analysis of the fragments to assist users in the reassembly process. Nevertheless, due to the combinatorial nature of the problem, assembling a large number of potentially heavily eroded and incompletely preserved fragments manually on the computer can still be an infeasibly laborious task. Therefore, it is common to apply computer-aided processing techniques, which can greatly enhance the efficiency of fragment matching in the virtual domain. These techniques enable a reduction of the solution space by, e.g., using an automatic feature-based pre-clustering of fragments or pre-selection of match candidates, but can also allow for a more accurate, optimized alignment of match candidates, which minimizes the accumulation of small alignment errors across successive adjacent fragments. Moreover, an automated quantitative assessment of matches, e.g., based on the geometry or texture of fragments along their common fracture surface, allows for establishing an objective validation process. We should note, however, that these automated methods are only reliable when applied to fragments that exhibit a sufficient amount of distinctive geometric or texture features, which additionally have to be widely preserved to provide a minimum of ambiguity and degeneration due to weathering and erosion. Unfortunately, this is rarely the case for real-world artifacts.

In our work, we consider white marble slabs such as the one shown in Figure 1. These slabs exhibit a mostly flat surface with little salient surface texture, such as color or engravings, which would otherwise convey additional valuable cues for automated fragment matching. Due to the absence of distinctive texture features, the geometric degeneration of fracture surfaces due to erosion, and the general incompleteness of such fragment collections, this type of artifacts is among the hardest to reassemble. Because of the inherent ambiguities, entirely automatic approaches are hardly feasible, as the combinatorial and geometric nature of the problem leaves almost no room for error: just one incorrectly placed fragment, or drift resulting from accumulated errors of several slightly inaccurate alignments of logically correctly connected fragments, can prevent reassembly algorithms from ever finding the correct solution. Therefore, most computer-aided reassembly approaches employ automatic methods only to support users, e.g., by optimizing pre-defined matches, or by merely presenting match suggestions that users still need to verify. Nevertheless, ambiguities are also challenging for experts. In the presence of heavily eroded fracture surfaces, different experts might disagree upon the correctness of a match. Hence, a key insight is that manual and automatic approaches should be combined, as they can complement each other.

We present a system that involves a large number of citizen scientists from the broad public to collaborate on the reassembly of digitized fragments in a web-based interactive 3D environment (Figure 1, right), which was conceptualized in a previous publication [HKM*23]. Our system offers various levels of computer-assisted support functionalities to the users. In general, such functionalities can work on different levels of user involvement: while simple fine-registrations of matches that were already found and placed by the user are a mostly manual and labor-intensive search process, computer-generated match suggestions still require diligent inspection by the user in order not to hastily accept potentially incorrect matches proposed by the system.

At the time of writing we are conducting a large-scale experiment that to date records several thousand involved citizen scientists. We analyze the utilization and measurable quality of our provided automatisms during the users' attempts to solve the archaeological puzzle. We thereby address the fundamental question which system parameters (one key parameter being the provided level of computer-assisted support for a specific group of users) are best suited to aid the ongoing reassembly progress, and the user motivation (derived for example from the recorded number of repeated logins of each user, and the overall number of their proposed joins) in the given virtual setup. For this experiment, we specifically chose a broken stone slab with a profiled frame made of white marble having almost no geometric and textural surface features, which additionally exhibits the aforementioned challenges of erosion, abrasion, and chipping. The approximately 100 fragments of this ancient artifact were found during the 1952 and 1953 excavation campaigns in the western part of the early Christian bishop's church at Kirchbichl in Lavant [Mil54], where it originally served as an altar slab (the so-called mensa). The first attempts to partially assemble these pieces using adhesive were made shortly after they were found, but were abandoned due to the predominantly inadequate and inaccurate, in some cases even faulty, connections of fracture surfaces, and the time required for the reassembly. Using our web-based collaborative reassembly system and the computer-assisted geometric support functionalities presented and evaluated in this paper, we aim to provide citizen scientists with means to efficiently and effectively participate in the archaeological process, and thereby contribute to science in a playful manner.

2. Related Work

The task of assembling virtual fragments is typically formulated as either a registration problem [APM15], a pattern-matching problem [PPY*16] or a generic geometric estimation problem [WC08]. The existing reassembly methods mostly revolve around the search for corresponding fragments based on characteristic features such as their fracture boundary curves and surfaces, or their texture.

2D Reassembly along Fracture Curves. The methods in this category reduce the combinatorial search space by focusing on the breaking curves of relatively thin fragments. Published techniques match and align contour curves of flat fragments [KK01; PPE*02], or of thin-walled pottery fragments [ZHLW14]. These geometry-steered approaches can be augmented by color information [DTS21]. This way, flat painted fragments can be correctly reassembled even when boundaries are abraded and colors faded.

© 2024 The Authors.
Proceedings published by Eurographics - The European Association for Computer Graphics.

Reassembly of Planar 3D Objects. For flat objects with a relevant thickness, it is advisable to restrict the degrees of freedom of the match-search space, which decreases the number of transform permutations to process and thus the complexity. Brown et al. [BTN*08] resample the fragment edges uniformly into a "ribbon" (extracted profiles), and use a 3D contour matching approach for reconstructing wall paintings. Their work was later extended by a machine learning approach [FST*11]. In the Stanford's Digital Forma Urbis Romae project, since the approach of converting the fragment edges to 2D contours does not result in new matches of the heavily eroded fragments [Kol08], the reconstruction is carried out by matching the incisions on the top surface of this ancient town map. Papaodysseus et al. [PAE*12] introduce matching criteria of restrictions for the reassembly of broken fragments having one almost planar surface and focus on the contact surfaces between them. All these mentioned techniques assume that all fragments in consideration belong to one object.

User-Assisted Fragment Matching. Several papers propose a semi-automatic approach in which the user has control over the reassembly process, by being able to accept or refuse matches. Mellado et al. [MRS10] present a semi-automatic reassembly approach where the user can suggest a match in an interaction loop and subsequently refine it. Another method introduces a framework where the user can flexibly set constraints and the system searches for a suitable solution [PPCS13]. Thuswaldner et al. [TFK*09] pursue a similar strategy by incorporating the know-how of architects and archaeologists for the identification of matching blocks of an ancient marble monument. Ye et al. [YZC*23] present an interactive system allowing expert users to revise an erroneously reassembled artifact. De Lima-Hernandez and Vergauwen [dLV20] showcase a hybrid reassembly approach with a 3D puzzling engine for ancient decorated wall fragments. For the latter, fragment matches and rough alignments are determined automatically based on the geometry before an expert user can fine-tune the alignment using a set of registration tools. In 2010, the web-based framework HINDSITE [KA10] was proposed for the interactive assembly of 2D fragments. The authors emphasize that the interactive approach is essential to counteract the combinatorial explosion of possible matches. In contrast to our fully collaborative system for the broad public, HINDSITE is exclusively aimed at individual users, who are furthermore assumed to have expert knowledge.

Crowd Sourcing and Gamification. Besides the reassembly of excavated fragments, puzzling also serves the purpose of engaging the general public with cultural heritage. A prominent example is the 3D artifact puzzle by Rodriguez-Echevarria et al. [RSLW21]. In this interactive web-based puzzler, citizens have the opportunity to reassemble the pottery sherds of an Iron Age funeral urn. To this end, a real-world artifact was synthetically fragmented and participants snap the pieces to the inner volume of the vase. This simulation constitutes a virtual extension of a previous puzzle game consisting of 3D printed pieces of the very same puzzle [RSW20]. Similarly, Capece et al. [CEGA20] present a so-called *serious game* in which single users can reassemble artifacts in a virtual environment, however, without providing any automatic supporting functionalities such as snapping or match suggestions.

Even though gamification has been used for engaging citizens with archaeological puzzles before, there is a lack of research on intuitive interfaces that convey the game mechanics in an easily understandable manner. Also, existing puzzle games are mostly single-player experiences. There is, to our knowledge, no approach yet which engages citizens in a collaborative multiplayer puzzle game aimed at solving an archaeological real-world problem.

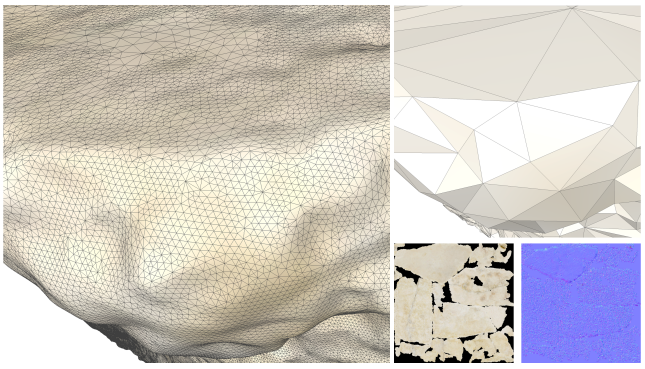
3. Digitalization

The dataset that is currently used in our system consists of digitized fragments of a broken mensa slab made of marble. The mensa slab was recently brought into focus as part of a larger restoration project of the early Christian bishop's church of Lavant [ABRP23]. As mentioned in Section 1, the mensa slab was in a bad state of convervation, which necessitated the careful disassembly of all bonded fragment groups and the dissolution of adhesive residues. After this restoration, there were 139 individual fragments, each weighing between 16 g to 9.5 kg with a total weight of 305 kg.

Once the stones were properly cleaned by the restorer, we digitized each fragment. Our primary capture device was a structured-light scanner with a field of view of 450 mm (x/y resolution of 109 µm). While the mensa slab is not painted, the subtle natural colorization of the marble stone (Figures 3 and 7) is indicative of coherent regions. In order to more faithfully capture the surface colors, we additionally digitized all fragments using Structure from Motion [YJHB14]. In another pre-processing step, several small, broken-off fragments were merged virtually with their vertically neighboring fragments so that all virtual fragments contain a part of the bottom and the top of the mensa slab. Fragments are pre-aligned vertically so that, after correct reassembly, their top surfaces form the mensa-slab's top plane. In the end, this resulted in 98 virtual puzzle pieces with a total file size of 14 GB. The fragments have an average diameter of about 150 mm, and a mean height of 94 mm. Figure 1 illustrates several real-world stone fragments alongside their virtual counterparts.

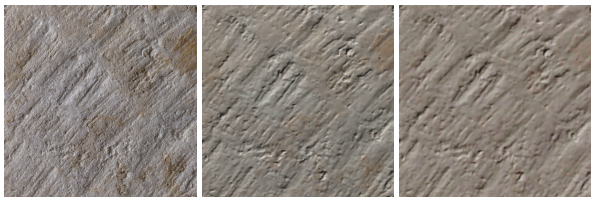
Each high-detail stone mesh resulting from the 3D scans contains an average of 3.5M vertices with vertex colors and 7M triangles, resulting in an approximate average file size of 140 MB. As, at the time of writing, it would be too taxing for consumer-grade computers to display close to 100 such meshes at interactive frame rates, the high-detail meshes were simplified in a pre-processing step. The simplified meshes that are rendered at runtime consist of 18K vertices (just positions, no vertex colors), and 33K triangles on average. Additionally, we create two $1K \times 1K$ textures, to store the albedo color and normal map for each stone. Figure 2 shows the difference in geometric detail as well as the generated albedo- and normal textures. The normal map, combined with a suitable shading model, manages to capture most of the high-frequency detail that would otherwise be lost after the geometric simplification. The data for such a simplified stone mesh and its textures amounts to 4.8 MB on average, which corresponds to roughly 3.4% the size of the high-detail meshes. Figure 3 shows that the simplified meshes accomplish to retain subtle toolmark indentations on the fragment surface, which play a pivotal role in match checking. Our user study (Section 6) shows that this visual representation of the stones does not hinder correct reassembly of the entire artifact, despite the significant geometric simplification.

© 2024 The Authors.
Proceedings published by Eurographics - The European Association for Computer Graphics.



(a) (b)

Figure 2: *Geometry preprocessing of high-resolution scanned stone (a) to low-resolution textured mesh with albedo- and normal map tailored for interactive rendering in the web browser (b).*



(a) *Real stone* **(b)** *High-detail mesh* **(c)** *Simplified mesh*

Figure 3: *Toolmarks captured in our digitalization and simplification asset pipeline. Photo of real stone (a), high-resolution mesh from scan (b), simplified mesh for rendering in the web browser (c). Color differences are mainly due to varying lighting conditions.*

Besides the simplified meshes used for rendering, the scanned high-resolution meshes need to be turned into point clouds for the algorithmic automatisms which assist users in the reassembly process. The high-resolution meshes are converted to point clouds by uniformly sampling their triangle surface. Since the puzzle is planar in nature, top- and bottom surfaces (identified by the magnitude of their normal vector's "up" component) are never needed for any of the computer-assisted alignment calculations, and are thus excluded from the point-cloud conversion. Point clouds are only used internally by our system and never transferred to the users.

Table 1 summarizes the average number of vertices, triangles, and the storage requirements of the stone representations. For the low-resolution meshes, the file size includes the associated textures. The high-resolution meshes are not used at runtime.

4. Collaborative Reassembly System

In this section we describe our system, starting with an overview of the overall architecture, and then detailing each building block, namely the game server, web client, compute server, as well as the logging-, and user databases.

As our application is open to the broad public, we dispatch users to one of several concurrently running, separate virtual instances of the puzzle to prevent too many users from interfering

Table 1: *Statistics of the three representations for a typical stone fragment. Rows from top to bottom denote high-resolution meshes (Scan), low-resolution meshes and their textures (Web), and point clouds for algorithmic fragment matching (PC). Columns from left to right show file size in MB, vertex count, triangle count, and texture count at respective texture resolutions.*

	Size [MB]	#Vertices	#Triangles	Textures
Scan	140.0	3.5M	7M	/
Web	4.8	18.0K	33K	2 (1K×1K)
PC	1.4	60.0K	/	/

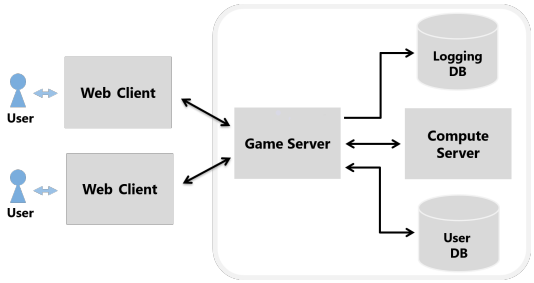


Figure 4: *System architecture overview of one virtual room.*

with each other. We call these separate instances of the same puzzle *rooms*. Currently 11 rooms are available to the public. Each room is run and managed by its own game server, which also stores the stone positions and connections already made, to persist the solution progress of the reassembly. We refer to single stone fragments as well as groups of stones that users have already connected, or joined, as *subassemblies*. Figure 4 illustrates the key components involved in running a single virtual room. While all virtual rooms contain the same set of fragments, we not only vary the maximum allowed number of concurrently active users, but also the number of computer-assisted tools (the so-called modes *A*, *B* and *C*, Section 5) available to the users. The calculations responsible for providing the geometric automatisms are carried out on a compute server (Section 4.3).

Our system manages a single global user database which associates each user to a server and tracks the number of logins per user. Additionally, each game server writes all user interactions to an associated logging database, which provides insights into how the application is utilized and about the solution progress. This not only includes details about the reassembly process, i.e., subassembly movement and rotation, but also how often users flipped stones, adjusted the light settings, performed up- and downvotes, and changed the color mode to name just a few. Detailed explanations of these interactions are given in Section 4.1.

4.1. Web Client

The web client provides many interaction features and access to all tools required for intuitive user collaboration. Due to the planar nature of our study object, puzzle pieces are pre-aligned to a certain height above the virtual table, so that users do not need to adjust this height offset. Translation of fragments is thus constrained to a

© 2024 The Authors.
Proceedings published by Eurographics - The European Association for Computer Graphics.

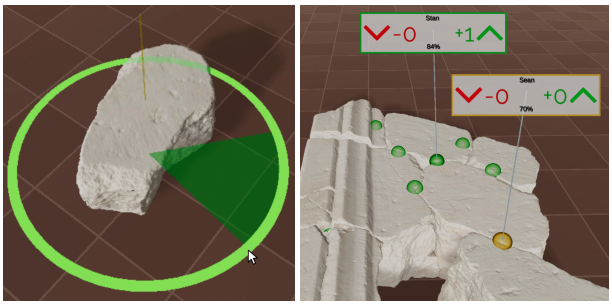


Figure 5: *Translation-, and rotation widgets (left), voting widgets (right). Selected subassemblies are surrounded by a circle that enables rotation by drag and drop (the green circle sector shows the rotation angle). Individual selected stones of the subassembly are marked with a vertical line. Connected stone boundaries are marked with a sphere that reveals a vote panel when clicked.*



Figure 6: *A stone subassembly seen from the top (left) and from the bottom after flipping (right), without having to move the camera.*

2D plane slightly above the virtual table, while rotation can be performed about the world-up axis (the normal of the table surface) only. Fragments are moved across the table by performing a drag and drop gesture. Similarly, rotation is performed via drag and drop on the circle displayed around the selection (Figure 5, left). The circle also serves as a progress indicator when geometric automatisms take longer to calculate to provide visual feedback to users.

Furthermore, while stones are pre-oriented so that their top faces point upward initially, they can be flipped upside down. Flipping is a global switch which always affects all stones. It is an important feature, as the top side of the mensa slab was intended to be mostly flat, which sometimes makes it nearly impossible to identify matching features between stones. Without the flipping functionality, the bottom sides of the stones could not be viewed by the users, since the camera may never penetrate the virtual table in order to avoid confusion. The bottom side of the stones, however, was only roughly carved, as this part of the stone was not meant to ever be visible, and thus can reveal important geometric hints for the correct alignment of neighboring stones. Figure 6 shows the top and bottom surface of a stone subassembly.

While the fragments are restricted to 2D transformations only, users can still view the scene, its fragments, and their fracture surface with a freely moving 3D camera. This setup enables fast inspection of the entire scene and intuitive alignment. Furthermore,



Figure 7: *Importance of lighting. Disadvantageous light parameters inadvertently hide surface detail (left), improved light settings reveal toolmarks (right).*

light directions and intensities can be adjusted in order to emphasize the subtle toolmarks, just as an archaeologist would vary the direction of incident light in order to reveal these otherwise invisible geometric features. These features constitute important reassembly hints as illustrated in Figure 7, which demonstrates that the enhancement of these toolmarks is a decisive factor for a successful reassembly.

Once users have connected single stones and subassemblies to form ever-growing subassemblies, these proposed joins can be inspected by all users in the same room. Each border of two connected stones is marked by a small colored sphere. Upon clicking this small sphere, a vote panel opens above the stone. The panel displays the name of the user who proposed the join, the calculated quality and also shows an up- and downvote arrow. Users can share their opinions about connections by clicking these vote buttons. The votes are then attributed to the user who created the join, boosting their ranking within the room. Furthermore, the joins are inserted into a list in the user interface, which is sorted by match quality. Each list entry contains a button that focuses the camera on the associated connected stones in the 3D view for direct inspection. Upon voting, the corresponding sphere for the join reflects the combined score of the calculated match quality as well as the user votes, where red indicates an overall bad match, yellow an acceptable match, and green an excellent match. The widgets for translation, rotation, and voting are illustrated in Figure 5.

Besides realistic stone colors, users can select two additional color modes, namely random colors per stone (Figure 8, left), and random colors per subassembly (Figure 8, right). The first random color mode helps to quickly identify individual stones in already joined subassemblies to better assess how well the adjacent fracture surfaces match. The second random color mode immediately reveals all subassemblies.

When joining stones, merely looking at the touching borders from the top or bottom can be deceiving. Since our virtual fragments are not solid bodies like in the real world, they can overlap in the simulated environment. By disabling backface culling, we can not only see the mesh surfaces from the outside, but also from the inside of the fragments. This way, the connection surface of match candidates can be inspected for mutual interpenetration, or even z fighting in case of a near-perfect match. Figure 9 shows two joins

© 2024 The Authors.
Proceedings published by Eurographics - The European Association for Computer Graphics.

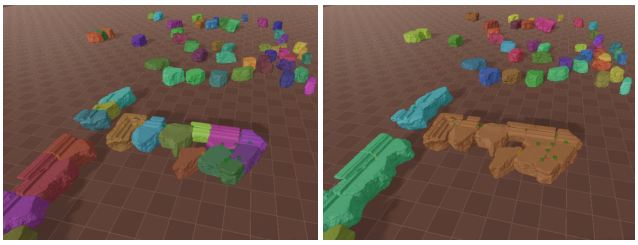


Figure 8: *Additional Color Modes. Random color per stone (left), random color per subassembly (right).*

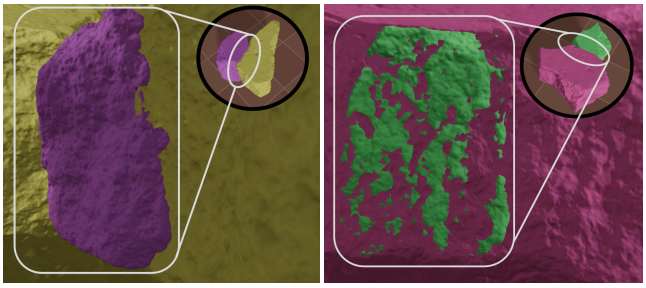


Figure 9: *Stone connection viewed from inside a stone. Bad match (left), good match (right). The inlay shows the stones from above.*

that both look like a good fit from above (see the respective inlays for the top views). However, on closer inspection from inside one the two fragments, Figure 9 (left) reveals extensive interpenetration of the connecting border surfaces, and thus an overall bad match, while in Figure 9 (right) we see a pattern of mutual small interpenetrations, hinting at a better match overall. Note that our system does not explicitly check for small-scale collisions, and thus allows for a certain degree of stone overlap. This prevents over-constraining manual placement by users as well as placement determined by the fitting algorithm, which, in turn, can also be calculated faster.

4.2. Game Server

Each game server is responsible for handling one room, and thus acts as the junction point for all system components. For one, it maintains a synchronized view of the scene for all users. Besides the networking orchestration, the game server persists created joins and all stone positions and rotations in the so-called *table state*. For each join, the table state stores the originating user – for which the game server queries the user database – along with the creation timestamp and the associated match quality. Subassemblies, i.e., one stone or a group of joined stones, are stored in a flat scene graph, where each stone contained in the subassembly is a direct child of its subassembly root node. The game server ensures that alignment and join-creation operations always happen between exactly two subassemblies.

Any join, split, or snap request initiated by the user is sent from the web client to the game server first. The game server then takes care of the asynchronous communication with the compute server, which processes the actual task. Upon receiving the result from the compute server, the game server ensures that the new table state is consistently replicated to all clients. Further, whenever users initiate an action, such as changing the position of a subassembly or splitting previously established joins, a new entry for the action is created in the logging database for later user-study evaluation.

4.3. Compute Server

The compute server is responsible for handling alignment-, as well as split- and join requests triggered by users from any game server. Since in our presented application the puzzle solving happens in 2D, the geometric automatisms calculated by the compute server enforce the respective movement and rotation constraints.

The compute server is stateless, i.e., it does not keep track of the table state of any of the rooms that it receives requests from. This lifts the burden of having to keep stone states in sync between the game server and the compute server. As a downside, this means that for every request, the compute server has to first reconstruct the current state of the two involved subassemblies. The compute server thus relies on the game server to transfer the state of the two involved subassemblies for each geometric operation. The transferred state is lightweight, though, and does not put any notable stress on the network connection between the two servers. It only includes the stone identifiers, the global poses of all involved stone fragments, and the connectivity graph which contains a node for each stone and an edge for each stone connection, along with the identifier of the user that proposed the join, as well as the join's associated calculated quality.

When reconstructing the state of the two subassemblies, the compute server replaces the referenced stone-fragment meshes with their associated pre-computed point clouds (Section 3). After placing all stone-fragment point clouds within a subassembly according to the supplied state, the points that form connection surfaces within larger subassemblies are filtered out, as an already connected part of a stone boundary cannot be connected a second time to another stone without overlapping (which cannot happen in the real world). This removal of "inner points" within a submassembly point cloud is done with a point-neighbor search among each contained adjacent stone point cloud pair. Any found point neighbors are excluded from the subassembly point cloud. After these composition and filtering steps, each of the two subassemblies is represented as a point cloud that only has "outer" boundary points that are eligible for a match. At this point, any requested geometric operation can be carried out. These calculations realize the computer-aided support functionalities, which will be discussed in the following in more detail.

5. Computer-Aided Support Functionalities

The two main computer-aided support operations provided by our system are snapping (or matching) and joining/splitting.

Snapping. Automatic snapping is a desirable feature, not least because the calculated alignment transformations may be hard to perform manually by users, and is provided in three flavors, namely *Mode A*, *Mode B*, and *Mode C*. *Mode A* performs fine registration of two carefully pre-aligned stone subassemblies, snapping them together at their matching surface. *Mode B* lifts the requirement

© 2024 The Authors.
Proceedings published by Eurographics - The European Association for Computer Graphics.

Table 2: *Search radii, sample count in % of the number of boundary vertices, and iteration count for each of our ICP stages.*

Stage	Radius [mm]	Sample Count [%]	ICP Iterations
1	30	0.25	10
2	10	1.00	10
3	4	10.00	10

of *Mode A* that subassemblies need to be pre-aligned and placed close to each other. Any two subassemblies can be selected and snapped by *Mode B*. *Mode C* goes one step further and lets the user select a single subassembly only, then finds a matching subassembly and snaps the two subassemblies together. Our virtual rooms are instances of three different types that differ by the snapping modes that are available to the users. We call these room types RT_A (*Mode A* matching only), RT_{AB} (*Mode A* and *B* matching), and RT_{ABC} (*Mode A*, *B*, and *C* matching).

Joining & Splitting. In contrast to the snap operation, the join- and split operations do not change any transformations. Joining two subassemblies merely updates the connectivity graph to reflect the new grouping of stones to their merged parent subassembly. Besides this structural update of the connectivity graph, the compute server also calculates a quality measure to communicate to the users how well the proposed join matches. The error e is calculated by summing the squared distances of all matching neighbor point pairs divided by the squared maximum neighbor-search distance and the number of found pairs. The calculated error lies in the range $[0;1]$. The quality q is then defined as $q = 1 - e$. Detaching stones from a subassembly is achieved by the splitting operation. The removed stones form new smaller subassemblies as a result.

5.1. Snapping Two Selected Pre-aligned Fragments (*Mode A*)

The most basic geometric support provided by our system is the automatic optimization of the alignment of two stone fragments based on their initial placement by a user, also widely known as *snapping*. Formally, the snap operation is an instance of the registration problem, i.e., the alignment of two or more separate surfaces that expose some degree of overlap to form a coherently merged shape. The task of aligning two stone fragments is a classic example of the subcategory of *rigid* registration problems and is stated as finding the parameters of a rigid transformation matrix, i.e., translation and rotation, that aligns one fragment with the other along their common fracture surface.

In practice, this type of operation is commonly realized by employing a variant of the Iterative Closest Point (ICP) algorithm [RL01]. The choice of the particular variant typically depends on the specific use case and requirements of the applications. As our system employs the ICP algorithm in an interactive, dynamic process of fragment alignment in a puzzle game, we aim for fast response times. At the same time, we need to ensure a high level of accuracy in the registration result to counter drift, i.e., successive accumulation of registration errors over multiple aligned stones, which could lead to deviations hindering the insertion of further pieces. Since all stones are pre-oriented towards their top side and constrained to the common planar surface level of the slab, we can also exploit the fact that the parameter space is restricted to three unknowns: a 2D translation and a single rotation angle around the world-up axis (Section 4.1).

ICP alignment is set up such that two subassemblies to be snapped need to be pre-aligned in close proximity, which enforces a conscious definition of the intended alignment by users. Further, this enables the reduction of the search radius for potentially corresponding surface points, which not only reduces the computation time for the involved neighbor queries, but also improves the robustness of the registration result.

ICP performs two steps iteratively: a matching step, which finds pairs of corresponding closest points across the two surfaces, and an alignment step, which updates the transformation parameters of one fragment to minimize its alignment error towards the other based on a certain error function. Variants of the ICP algorithm can be differentiated, among others, by their selection strategy for point samples (sampling), their point correspondence approach (matching), and their weighting of pairs [RL01]. In our implementation, we consider only two-way point correspondences, i.e., pairs of points from opposing fracture surfaces that are mutual nearest neighbors. The contribution of each pair to the alignment error is weighted by the degree of alignment of the opposing normals.

In order to maintain both fast processing as well as high accuracy, we employ a three-stage incremental fitting approach that first starts with a larger neighborhood for finding point correspondences on a smaller sample count for quick initial alignment, and then refines the alignment by increasing the sample count while at the same time decreasing the neighborhood search radius. Point candidates are sampled once per stage and only from the outer band of the fracture surfaces of a subassembly, disregarding the top and bottom parts of its surface (Section 3). If no matching point pairs are found at the first stage, the subassemblies are not translated or rotated at all. For an average-sized stone fragment of about 15 cm in diameter, the number of boundary vertices amounts to about 60K points (Table 1).

Table 2 lists the ICP parameters used in all three stages of our implementation. Note that we omitted a final fourth ICP stage that works on *all* points, as in our experiments on fitting two single average-sized fragments of about 15 cm diameter and about 60K boundary points each, it increased the ICP runtime by a factor of over 30 (from 0.06 to 2.3 seconds), while reducing the average converged distance between final corresponding pairs only by 0.1 mm (from 0.57 mm to 0.47 mm). Also note that with about half a millimeter, the final average error is about 0.3% the size of a stone fragment, and below the average vertex distance in the full model (0.667 mm). While on the compute server, the finally used three-stage ICP calculation takes less than 0.1 seconds for two such fragments, the total observed delay including networking overhead between triggering the snapping and observing the responding snapped position in the web client takes up to 0.8 seconds.

5.2. Automatic Matching of Any Two Fragments (*Mode B*)

Mode B supports snapping of two subassemblies that are located arbitrarily far apart from each other. It does not require any alignment hints based on the current global poses of the two subassemblies,

© 2024 The Authors.
Proceedings published by Eurographics - The European Association for Computer Graphics.

Algorithm 1 SnapToSelected (*Mode B*)

Require: $movingPC, fixedPC$ ▷ Point clouds to be snapped
Ensure: $(bestQuality, bestTrafo)$ ▷ Best alignment found

function SNAPTOSELECTED($movingPC, fixedPC$)
  $(bestQuality, bestTrafo) \leftarrow$ ICP2D($fixedPC, movingPC, \mathbf{I}, \mathbf{I}$)
  for $i \leftarrow 1$ to $maxIter$ **do** ▷ User-defined iteration count
    $ptFixed \leftarrow$ RANDPTFROMPC($fixedPC$)
    $ptMoving \leftarrow$ RANDPTFROMPC($movingPC$)
    $tMtx \leftarrow ptFixed - ptMoving$
    $rMtx \leftarrow$ ROTYOPPOSITEDIR($ptFixed_N, ptMoving_N$)
    $(q, mtx) \leftarrow$ ICP2D($fixedPC, movingPC, tMtx, rMtx$)
    if $currQuality > bestQuality$ **then**
      $(bestQuality, bestTrafo) \leftarrow (q, mtx)$
    end if
  end for
  return $(bestQuality, bestTrafo)$
end function

Algorithm 2 SnapToAny (*Mode C*)

Require: $movingPC$ ▷ Point cloud to be snapped
Ensure: $(bestQuality, bestTrafo, bestPC)$

function SNAPTOANY($movingPC$)
  $bestQuality \leftarrow -1$ ▷ Best quality found so far
  $bestTrafo \leftarrow \mathbf{I}$ ▷ Best trafo found so far
  $bestPC \leftarrow NULL$ ▷ Best-matching point cloud so far
  for $i \leftarrow 1$ to $maxIter$ **do** ▷ User-defined iteration count
    $fixedPC \leftarrow$ RANDOMOTHERSUBASSEMBLY()
    $(q, mtx) \leftarrow$ SNAPTOSELECTED($movingPC, fixedPC$)
    if $currQuality > bestQuality$ **then**
      $(bestQuality, bestTrafo) \leftarrow (q, mtx)$
      $bestPC \leftarrow fixedPC$
    end if
  end for
  return $(bestQuality, bestTrafo, bestPC)$
end function

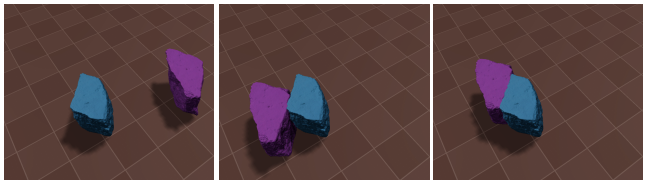
so that – contrary to *Mode A* – users are not required to perform any manual translations or rotations for Mode B to work. In order to still provide fast response times, *Mode B* does not calculate a perfect fit, but rather tests several random positions and returns the highest-quality match encountered.

Algorithm 1 outlines the operation of *Mode B*. It expects two point clouds, *movingPC* and *fixedPC*, as input, which correspond to the two subassemblies to be aligned and returns the required alignment transformation matrix *bestTrafo* with the respective quality measure *bestQuality* as a result. These variables are initialized by calling *Mode A* with the identity transformation **I**. If *Mode A* fails to find a match (i.e., the two subassemblies are located far apart), it returns a quality of -1 and an alignment matrix equal to the identity matrix **I**. Each loop iteration starts with picking random points *ptfixed* and *ptMoving* from each point cloud. The translation matrix *tMtx* aligns *ptMoving* and *ptFixed* on the xz-plane, ignoring their relative offset along the world-up y-axis. The rotation matrix *rMtx* around the y-axis ensures that the normal $ptMoving_N$ at position *ptMoving* points in the opposite direction of the normal $ptFixed_N$ at position *ptFixed*. Note that normals are assumed to point outside. Finally, the 2D-constrained ICP algorithm from *Mode A* is called and returns the quality q and the alignment matrix *mtx*. When the loop finishes, the best quality *bestQuality* calculated so far, along with the corresponding alignment matrix *bestTrafo* is returned to the caller. Thus, when called repeatedly, Algorithm 1 returns alignment transformations of non-decreasing matching quality.

Figure 10 illustrates an exemplary repeated application of *Mode B*, which finds a good alignment of two selected fragments, even when they are far apart and initially even located on opposing sides compared to the correct match. We can observe how the solution gradually improves until the correct match is found after just five applications of *Mode B*.

5.3. Match Suggestions for Single Fragments (*Mode C*)

Mode C is the snapping algorithm that requires the least user involvement. The user merely selects one subassembly, for which the system proposes a potential matching subassembly from all available other subassemblies. Just as *Mode B*, in favor of short execution times, *Mode C* does not necessarily find the perfect match.



(a) *Initial Position* **(b)** *Quality = 71.1%* **(c)** *Quality = 85.2%*

Figure 10: *Mode B matching enables users to automatically align two fragments by merely selecting both. Initial position (a), after one application of the auto alignment (b), gradual improvement after four more applications of the auto alignment (c).*

Algorithm 2 describes how alignments are found in *Mode C*. The algorithm expects the input point cloud *movingPC* and returns the best matching other subassembly point cloud *bestPC* along with the alignment transformation *bestTrafo* and the corresponding match quality *bestQuality*. In each loop iteration, a random other subassembly point cloud *fixedPC* is chosen. Then Algorithm 1 (*Mode B*) is invoked for the two point clouds, and returns the best alignment along with the quality of that alignment. The resulting quality *currQuality* of this invocation guides the selection of the best-matching fit across all randomly tested subassemblies. The subassembly point cloud *bestPC* for which the best quality was reached is returned as a result along with the alignment transformation *bestTrafo* and the corresponding quality *bestQuality*.

6. Results

In this section, we report results based on logging data we recorded after our system had been online and available to the public for about one month. In this timeframe, we were able to record 11 718 registered users and 12 244 logins across 11 virtual rooms (four RT_A, three RT_{AB}, and four RT_{ABC} room-type instances, respectively) of which 4K were one-time-only logins. Amongst the registered users, 1 518 users were actively involved in either joining, splitting,

© 2024 The Authors.
Proceedings published by Eurographics - The European Association for Computer Graphics.

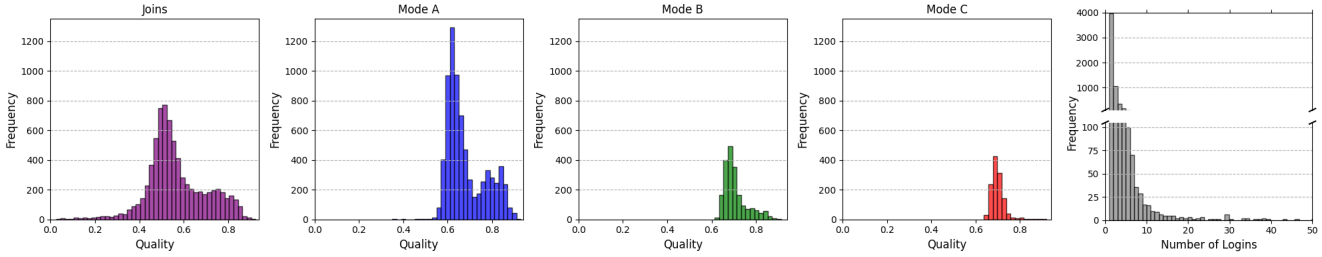


Figure 11: *From left to right: Quality distributions of join and automatic modes A, B, and C, respectively. The y-axis shows how many times the specific quality was reached across all rooms. The rightmost chart reports the repeated login frequency of all users.*

Table 3: *Raw statistics of user numbers and feature utilization over all virtual rooms as recorded by our logging database.*

	#Invocations	By Unique Users
Logins	12 244	12 244
Users	11 718	11 718
Mode A Matches	7 616	355
Mode B Matches	2 018	270
Mode C Matches	1 335	518
Votes	6 960	753
Joins	7 891	591
Splits	6 850	551
Color Mode Toggles	14 160	/
Flip Stones	354 746	/
Light Rotations	278 876	/

voting, or performing matches through *Mode A*, *B*, or *C*. The maximum number of concurrently active users recorded in any room was 32. We managed to attract several recurring users (Figure 11, right). On average, each user logged in two times, the maximum login count for a user being 190. Table 3 shows an excerpt of our logging database.

When comparing the number of joins to the number of splits in Table 3, we can deduce high user activity. Many proposed joins were revised and split again, however, not all joins were split, which hints at an overall solution progress. The number of votes recorded is another marker for high user activity. As the intended usage of the voting feature is the possibility for users to share their opinions about joins, and since the logged number correlates to the number of joins being kept, as well as splits being motivated, this hints at user adoption of this feature in the desired way.

Table 3 also reveals that *Mode A* snapping was used the most, which can be explained by the fact that it is available in all 11 rooms, followed by *Mode B* (available in seven rooms – three of type RT_{AB} and four of type RT_{ABC}), and *Mode C* (available in four virtual rooms of type RT_{ABC} only). Setting the number of application of computer-aided modes *A*, *B*, and *C* in relation to the number of joins indicates that the automatic modes played an important role in the reassembly process, and reflects a high adoption rate of the automatisms by the users of our system. This shows that users feel confident about the automatically calculated alignments and take advantage of the fast automatic fine-adjustments before joining stones.

Figure 11 shows how many times joins and snaps in modes *A*, *B*, and *C* reached a specific quality. The machine-assisted alignments (Section 5) reach match qualities between about 60% to 70%, while the quality for joins drops to 50% on average. Note that joins can be proposed by purely manual alignment without any computer-assistance. This shows that human choices for positioning might not always be better than automatic ones on average.

Besides login counts and adoption rates of reassembly-related system features, Table 3 reveals that participants also made heavy use of stone flipping, light rotation, and random color modes. This shows that these features, specifically implemented to assist users, were actually utilized by the participants. The effectiveness of our system for solving non-trivial reassembly tasks is documented by the fact that in three of the 11 public rooms, users were able to collaboratively reassemble large parts of the artifact within only two to four weeks time. The proposed solutions in these three rooms were correct up to at most five wrongly connected, or not yet joined, fragments. In one of these rooms, only one stone was left unconnected, while all other stones were joined correctly.

7. Conclusion & Future Work

We presented a collaborative framework that enables the broad public to piece together digitized real-world archaeological fragments in the browser. We demonstrated that with our tool citizen scientists can efficiently solve reassembly problems even if they present non-trivial challenges such as eroded or missing pieces, and lack of surface color or distinctive geometric features. Communication among users is based on a voting system that enables participants to evaluate proposed joins and share their opinions by up- or down-voting. Besides purely manual placement, users can take advantage of automatic snapping for fine registration, and even let the system find possible matching candidates for fragments to be joined. The 3D frontend offers users intuitive controls and features (e.g., adjustable light directions and intensities, a switch for flipping stones upside down, and viewing stone surfaces from the inside) for in-depth inspection of stones and their mutual alignments before fully committing to a new join or splitting and voting exisiting joins. Furthermore, while currently fragment interaction is restricted to 2D, this is not an inherent system limitation and an extension to 3D is well possible.

Future work will focus on the evaluation of the final collected user-study data in depth, with one of the main goals being the deter-

© 2024 The Authors.
Proceedings published by Eurographics - The European Association for Computer Graphics.

mination of system parameters that maximize the reassembly efficiency of participants. Examples for such system parameters are the maximum allowed number of collaborative users per virtual room, as well as the level of provided computer-aided methods.

Acknowledgement

This project is funded by the province of Styria (ABT12-603763/2022). We thank the museum Schloss Bruck (Lienz) for providing our study object, the mensa slab, and Paul Bayer for his support in the digitization of the fragments.

References

[ABRP23] AUER, MARTIN, BRUNNER, FRANZ, REKELHOFF, TIM, and PÖLL, JOHANNES. *Die frühchristliche „Bischofskirche" von Lavant. Ein archäologisches Juwel in seiner Kulturlandschaft*. wiederhergestellt 41. Vienna: Bundesdenkmalamt, 2023 3.

[APM15] ANDREADIS, ANTHOUSIS, PAPAIOANNOU, GEORGIOS, and MAVRIDIS, PAVLOS. "Generalized digital reassembly using geometric registration". *Proceedings of the 2015 Digital Heritage International Congress*. Vol. 2. 2015, 549–556. DOI: 10.1109/DigitalHeritage.2015.7419572 2.

[BTN*08] BROWN, BENEDICT J., TOLER-FRANKLIN, COREY, NEHAB, DIEGO, et al. "A system for high-volume acquisition and matching of fresco fragments: reassembling Theran wall paintings". *ACM Transactions on Graphics* 27 (2008), 1–9. ISSN: 0730-0301. DOI: 10.1145/1360612.1360683 3.

[CEGA20] CAPECE, NICOLA, ERRA, UGO, GRUOSSO, MONICA, and ANASTASIO, MARCO. "Archaeo Puzzle: An Educational Game Using Natural User Interface for Historical Artifacts". *Eurographics Workshop on Graphics and Cultural Heritage*. The Eurographics Association, 2020. ISBN: 978-3-03868-110-6. DOI: 10.2312/gch.20201301 3.

[dLV20] De LIMA-HERNANDEZ, ROBERTO and VERGAUWEN, MAARTEN. "A Hybrid Approach to Reassemble Ancient Decorated Block Fragments through a 3D Puzzling Engine". *Remote Sensing* 12 (2020), 2526. ISSN: 2072-4292. DOI: 10.3390/rs12162526 3.

[DTS21] DERECH, NIV, TAL, AYELLET, and SHIMSHONI, ILAN. "Solving archaeological puzzles". *Pattern Recognition* 119 (2021), 108065. ISSN: 0031-3203. DOI: 10.1016/j.patcog.2021.108065 2.

[FST*11] FUNKHOUSER, THOMAS, SHIN, HIJUNG, TOLER-FRANKLIN, COREY, et al. "Learning how to match fresco fragments". *Journal on Computing and Cultural Heritage* 4 (2011), 7. ISSN: 1556-4673. DOI: 10.1145/2037820.2037824 3.

[HKM*23] HOUSKA, PETER, KLOIBER, SIMON, MASUR, ALESSANDRA, et al. "Towards Crowd-Sourced Collaborative Fragment Matching". *Eurographics Workshop on Graphics and Cultural Heritage*. The Eurographics Association, 2023. ISBN: 978-3-03868-217-2. DOI: 10.2312/gch.20231180 2.

[KA10] KIMIA, BENJAMIN and ARAS, H. CAN. "HINDSITE: A user-interactive framework for fragment assembly". *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops*. IEEE Computer Society Conference, 2010, 62–69. DOI: 10.1109/CVPRW.2010.5543532 3.

[KK01] KONG, WEIXIN and KIMIA, B.B. "On solving 2D and 3D puzzles using curve matching". *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*. Vol. 2. IEEE Computer Society Conference, 2001, 583–590. DOI: 10.1109/CVPR.2001.991015 2.

[Kol08] KOLLER, DAVID R. "Virtual archaeology and computer-aided reconstruction of the severan marple plan". *Beyond Illustration: 2D and 3D Digital Technologies as Tools for Discovery in Archaeology*. British Archaeological Reports International Series 1805. Oxford: Archaeopress, 2008, 125–134. ISBN: 978-1-4073-0292-8 3.

[Mil54] MILTNER, FRANZ. "Die Grabungen auf dem Kirchbichl von Lavant/Osttirol. Dritter vorläufiger Bericht". *Jahreshefte des Österreichischen Archäologischen Institutes in Wien* 41 (1954), 43–84. ISSN: 2309-1207 2.

[MRS10] MELLADO, NICOLAS, REUTER, PATRICK, and SCHLICK, CHRISTOPHE. "Semi-Automatic Geometry-Driven Reassembly of Fractured Archeological Objects". *VAST: International Symposium on Virtual Reality, Archaeology and Intelligent Cultural Heritage*. The Eurographics Association, 2010, 33–38. ISBN: 978-3-905674-29-3. DOI: 10.2312/VAST/VAST10/033-038 3.

[PAE*12] PAPAODYSSEUS, CONSTANTIN, ARABADJIS, DIMITRIS, EXARHOS, MICHALIS, et al. "Efficient solution to the 3D problem of automatic wall paintings reassembly". *Computers & Mathematics with Applications* 64 (2012), 2712–2734. ISSN: 0898-1221. DOI: 10.1016/j.camwa.2012.08.003 3.

[PPCS13] PALMAS, GREGORIO, PIETRONI, NICO, CIGNONI, PAOLO, and SCOPIGNO, ROBERTO. "A computer-assisted constraint-based system for assembling fragmented objects". *2013 Digital Heritage International Congress (DigitalHeritage)*. Vol. 1. 2013, 529–536. DOI: 10.1109/DigitalHeritage.2013.6743793 3.

[PPE*02] PAPAODYSSEUS, C., PANAGOPOULOS, T., EXARHOS, M., et al. "Contour-shape based reconstruction of fragmented, 1600 BC wall paintings". *IEEE Transactions on Signal Processing* 50 (2002), 1277–1288. DOI: 10.1109/TSP.2002.1003053 2.

[PPY*16] PINTUS, RUGGERO, PAL, KAZIM, YANG, YING, et al. "A Survey of Geometric Analysis in Cultural Heritage". *Computer Graphics Forum* 35 (2016), 4–31. DOI: 10.1111/cgf.12668 2.

[RL01] RUSINKIEWICZ, SZYMON and LEVOY, MARC. "Efficient Variants of the ICP Algorithm". *Proceedings Third International Conference on 3-D Digital Imaging and Modeling*. 2001, 145–152. ISBN: 0-7695-0984-3. DOI: 10.1109/IM.2001.924423 7.

[RSLW21] RODRIGUEZ-ECHAVARRIA, KARINA, SAMAROUDI, MYRSINI, LLOYD, JACK, and WEYRICH, TIM. "Interactive 3D Artefact Puzzles to Support Engagement Beyond the Museum Environment". *Eurographics Workshop on Graphics and Cultural Heritage*. The Eurographics Association, 2021. ISBN: 978-3-03868-141-0. DOI: 10.2312/gch.20211403 3.

[RSW20] RODRIGUEZ-ECHAVARRIA, KARINA, SAMAROUDI, MYRSINI, and WEYRICH, TIM. "Fracturing Artefacts into 3D Printable Puzzles to Enhance Audience Engagement with Heritage Collections". *Journal of Cultural Heritage* 13 (2020), 6. ISSN: 1556-4673. DOI: 10.1145/3351343 3.

[TFK*09] THUSWALDNER, BARBARA, FLÖRY, SIMON, KALASEK, ROBERT, et al. "Digital anastylosis of the Octagon in Ephesos". *Journal of Cultural Heritage* 2 (2009), 1. ISSN: 1556-4673. DOI: 10.1145/1551676.1551677 3.

[WC08] WILLIS, ANDREW R. and COOPER, DAVID B. "Computational reconstruction of ancient artifacts". *IEEE Signal Processing Magazine* 25 (2008), 65–83. DOI: 10.1109/MSP.2008.923101 2.

[YJHB14] YI, GAO, JIANXIN, LUO, HANGPING, QIU, and BO, WU. "Survey of structure from motion". *Proceedings of 2014 International Conference on Cloud Computing and Internet of Things*. IEEE Computer Society Conference, 2014, 72–76. DOI: 10.1109/CCIOT.2014.7062508 3.

[YZC*23] YE, SHUAINAN, ZHU-TIAN, CHEN, CHU, XIANGTONG, et al. "PuzzleFixer: A Visual Reassembly System for Immersive Fragments Restoration". *IEEE Transactions on Visualization and Computer Graphics* 29 (2023), 429–439. DOI: 10.1109/TVCG.2022.3209388 3.

[ZHLW14] ZHENG, S., HUANG, R., LI, J., and WANG, Z. "Reassembling 3D Thin Fragments of Unknown Geometry in Cultural Heritage". *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences* II-5 (2014), 393–399. DOI: 10.5194/isprsannals-II-5-393-2014 2.

© 2024 The Authors.
Proceedings published by Eurographics - The European Association for Computer Graphics.