EUROGRAPHICS Workshop on Graphics and Cultural Heritage (2026), pp. 1–10
R. Pintus and C. Bosch (Editors)

Adaptive Surface Classification of Fractured Objects

I. Evangelou[1] and G. Papaioannou[1]

[1]Department of Informatics, Athens University of Economics and Business, Greece

Abstract

Fractured object reassembly and monument anastylosis are important archaeological processes that can benefit from supportive computational methods. In the past two decades, significant progress has been made in proposing working computational methods for the alignment of broken artefacts, including methods of specialized categories, such as potsherds and frescos. Unless strong priors are available about the overall reconstructed shape of the target object(s), a common task in all methodological approaches is the identification and isolation of fractured regions, a step that is necessary to prevent trivial matches and direct the alignment to meaningful solutions. In this paper we argue that the determination of the appropriate scale is important when attempting to detect features compatible with a fracture and propose an adaptive classification pipeline to address this. Finally, we contribute a curated and extensible dataset of annotated and pre-aligned fractured objects as a validation benchmark of both our method and future approaches.

CCS Concepts
• ***Computing methodologies*** → ***Shape analysis;*** *Mesh geometry models;* • ***Applied computing*** → ***Archaeology;***

1. Introduction

The restoration of fractured objects, using computational reassembly approaches, regardless of scale, involves the precise alignment of the pieces over their fracture surface or fracture line, in the case of fragments of negligible thickness. At an abstract level, the process resembles solving a 2D or 3D puzzle, where the number of target objects, the number of missing pieces, the shape and number of contact sides and sometimes even the final shape, are all unknown. Often, constraints such as planarity, texture continuity, self-similarity or distinctive geometric features provide additional priors that we can exploit. However, in the purely geometric sense, without focusing on areas of the geometry that we consider as a potential fracture, trivially compatible regions, such as flat or smoothly curving sides would be prioritized and trivially skew or totally skip any useful pairing, as demonstrated in Figure 1. This is particularly true for global registration methods used when no manual pre-alignment or other priors are assumed [PSA*17]; these regions simply provide highly-compatible and large support to be ignored.

One way to dispense with them is to seek salient features and their corresponding counterparts on other fragments, as in [HFG*06]. Another (orthogonal) option is to restrict the registration process to regions where curvature exhibits a high variance, indicating a chaotic relief, compatible to typical fractures. In both cases, the scale of surface features is critical in successfully detecting candidate variations, as typical detectors rely either in some mean-curvature-related metric (e.g. volume descriptors [PWHY09]) or dihedral angles, which are sensitive to surface sampling rate. In the literature, feature scale is one of the tunable parameters that varies among input specimens.

In this paper we propose a surface characterization approach that labels mesh vertices (or in fact, any arbitrary sample) in the continuous range between potentially intact and fractured, respecting spatial coherence but without explicitly requiring or enforcing a compact segmentation of the mesh. We propose an adaptive scheme to determine the appropriate scale of feature detectors and apply the approach to volume descriptors in order to characterize clusters of points as potentially fractured.

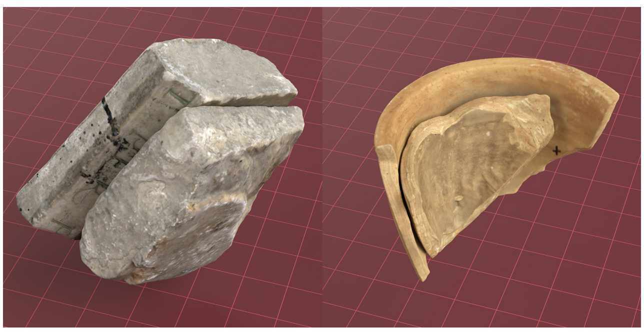

Figure 1: *Typically problematic trivial alignment of fragments, when registration is not limited to fractured surfaces.*

Second, we provide a Monte Carlo estimator for volumetric descriptors that operates in continuous space and eliminates problems caused by the discrete volume representation of previous integral descriptor implementations.

We also argue that segmentation of the surface into distinct potentially fractured and intact areas is indeed a relevant task but *not a necessary* stage by itself. Individually labeled points can drive any importance-driven sampling of the mesh surface to generate candidate correspondences between parts.

Finally we provide a curated benchmark dataset of real digitized fragments with pre-labeled surfaces in the range $[0,1]$, signifying the confidence of them being intact (0), or fractured (1), using a combination of automated processing and manual weight painting. Most available fragment datasets are either unlabeled (e.g. [TP13]), or too simplistic and unrealistic to be used in tuning labeling or segmentation methods for actual physical artefacts [SCW*22]. The benchmark dataset includes digitized and pre-assembled fractured objects of varying degree of completeness, symmetry and thickness. We test our method with the benchmark dataset and report detection success.

Please note that our method is purely procedural, not relying on learned features and neural models, as the latter would require a substantial amount of actual labeled fragments for training that simply do not exist. Additionally, many neural assembly approaches are trained on the complete set of parts, known apriori, which is rather unrealistic for actual cultural heritage artefacts. Conversely, we process each fragment individually and in isolation.

2. Related Work

Here we provide some background on tasks and quantities that we use in our method and a brief overview of the related work.

2.1. Volume Descriptor

The *integral volume descriptor* $V_r(\mathbf{x})$ was introduced in [GMGP05] and was further studied in [PWY*07, PWHY09]. Simply put, it measures the portion of a ball $B_r(\cdot)$ of radius r centered at a surface point $\mathbf{x}$ intersected by the interior volume "under" the surface. Points in the object's interior would satisfy an indicator function $Id(\mathbf{y})$, so that:

$$V_r(\mathbf{x}) = \int_{B_r(\mathbf{x})} Id(\mathbf{y}) d\mathbf{y}. \tag{1}$$

The volume descriptor for small values of r is related to the mean curvature H of the surface:

$$V_r(\mathbf{x}) = \frac{2\pi}{3} r^3 - \frac{\pi H}{4} r^4 + O(r^5), \tag{2}$$

making it a good candidate for examining the local, non-directional presence of relief at various scales of observation. Very distinctive features, such as ridges and deep recesses, would have similar V_r values at multiple scales, facilitating the detection of object edges.

In our work, we heavily rely on the estimation of the volume descriptor, too, but employ a normalized form:

$$\hat{V}_r(\mathbf{x}) = \frac{3V_r(\mathbf{x})}{4\pi r^3}. \tag{3}$$

Contrary to existing literature [PWHY09, APM16], we employ a Monte Carlo estimator in continuous space and avoid discretization problems that can be especially problematic at small volume descriptor scales r, both in terms of precision and efficiency (see Section 4.2).

2.2. The Reassembly Problem

In a computational reassembly or anastylosis problem, a system attempts to combine fragments and parts of one or more *intact* objects, under the assumption of a global minimum residual error and the avoidance of any inter-penetration. For the alignment assessment of pair-wise compatibility, one form or another of either feature or raw surface sample registration is used, minimizing the difference of examined pairs but not always in a least-squares sense [BTP13]. When *multi-part* reassembly is considered, as in [HFG*06] and [PSA*17], a combinatorial problem involving pair-wise solutions must be solved, also addressing the issue of indirect part penetration, i.e. the fact that even if the relative pose between two fragments is penetration-free, there is no guarantee that this is still the case when the cluster is expanded. Multi-part reassembly also typically involves a *bundle adjustment* stage, to refine contacts.

Contrary to the case of scan alignment, where overlap maximization and a low count of outliers makes the use of a local optimization approach ideal, in the case of fragment reassembly, too many local minima may exist ([PKT02]), necessitating the use of a combination of global search for the optimal alignment and the reduction of the search space via the detection of corresponding landmarks on feature-rich surfaces and the restriction of the search for matches on the potentially fractured faces of the parts.

For reader interested in the general problem of geometric reassembly, a comprehensive overview of both procedural and neural reassembly methods can be found in the recent survey by Lu et al. [LLH*25], covering application areas beyond the reconstruction of archaeological fragmentary material.

2.3. Segmentation

Segmentation and surface region or point sample labeling as potentially fractured is an important task in reassembly problems but also in more generic registration tasks, as it a) helps split the search over the general domain of all possible $SE(3)$ transformations per object into a more manageable set of targeted problems, enabling branch and bound approaches to efficiently attach the problem (e.g. [WW08]), and b) avoid undesired trivial configurations that are not meaningful for the context of the application. Explicit segment or cluster labeling is not always performed.

In the literature, the segmentation of the surface is performed first to dice the original geometry into congruent regions under some assumption of planarity or slow large-scale orientation change to accommodate curved surfaces. Dehedral angles between surface patches are a common criterion for merging evolving clusters. [PKT00] uses the dihedral angle between the evolving cluster and the candidate polygons in a region-growing approach, while [AMP14] relies on an agglomerative best-first clustering of evolving regions. While using the dehedral angle between the average normal directions of the compared clusters is sufficient for nearly planar surfaces, the criterion does not work well for highly curved surfaces as it tends to break up the surface into too many segments. Instead, a variation can be employed where only polygons near the boundary of the clusters are considered for the average normal direction computation and comparison.

The process typically results in non-contiguous regions and small unmatched patches left unclustered. These are assimilated by larger regions in a follow-up clean-up stage.

Xu et al. [XZW*15] performs segmentation after first smoothing the surface using a Laplace operator. Although this approach can eliminate small spurious relief changes and result in a cleaner landscape of segments, large morphological changes within broken and intact regions can still cause significant over-segmentation.

A different approach has been adopted by Winkelback et al [WW08], where a hierarchical clustering of the surface points is performed based on an initial decomposition using k-means clustering of the 6DOF field of combined positions and normals. The authors directly use the derived clusters for registration and the approach is tolerant to different per-object cluster generation.

Another way to address the surface segmentation problem is via the dual graph of the neighboring segments, which builds a connected graph of segment boundaries. The notion has been illustrated in [PKT00] and effectively employed as a segmentation strategy by Huang et al. [HFG*06]. In the latter, consistent values of the volume descriptor, corresponding to non-zero mean curvature, signify sharp overall surface orientation change (edges). The authors trace the largest connected paths of such sharp points to detect face boundaries, which are subsequently merged according to a roughness metric (see below) to form fewer and more meaningful clusters.

While in our method we do not expressly perform segmentation in this manner, we mark ridge or sharp ravine micro-clusters prior to segment development as a special "guard" segment in order to avoid crossing over sharp surface bends during cluster evolution. Sharp features are assimilated into regular segments at the end of the segmentation.

2.4. Labeling of Fractured Objects

Surface characterization as fractured or intact can be either explicitly performed via labeling that typically uses a surface variance metric (e.g. [PKT00, AMP14]) or omitted and the statistics of the cluster used during registration to filter candidate pairs (e.g. [WW08]).

Most methods in the literature (e.g. [HFG*06, PSA*17, XZW*15, AMP14]) rely for the segment labeling on some variant of the *local bending energy* [HFG*06] metric for a given point defined over samples in its neighborhood, averaged over all segment points:

$$e(\mathbf{x}) = \frac{1}{N_{neigh}} \sum_{k=1}^{N_{neigh}} \frac{||\mathbf{n}-\mathbf{n}_k||^2}{||\mathbf{x}-\mathbf{x}_k||^2}. \tag{4}$$

This approach attempts to address the fact that surfaces are generally unevenly sampled during digitization and at small scales of observation, where fracture detection attempts to observe variance in the relief of the surface, volumetric descriptors computed for larger feature detection (e.g. ridges, salient compound structures) tend to be unreliable [HFG*06]. In this work we attempt to lift this restriction by leveraging an adaptive strategy in the computation of multi-scale volumetric descriptors, which does not rely on a global and uniformly defined detection scale (radius range). The reason is that the volume integral descriptor can be indeed more informative as a detector of local surface morphology, compared with the local bending energy, whose statistics are indirectly related to crack relief changes. Furthermore, local bending energy is very dependent on the topology of the surface and sensitive to non-uniform sampling or mesh simplification.

Finally, when a regular (re-) sampling of the surfaces is available, as in [PKT00], the differential operation of *bumpiness*, which is many ways similar to the local bending energy can be computed instead. This approach aligns and projects a segment to a projection plane and performs regular depth sampling and grid-based distance differences to estimate the variance of the segment's relief. This metric is also sensitive to observation scale, although it can be efficiently computed in multiple scales exploiting image-space mip-mapping.

We expressly not include in the related work learning-based methods as the proposed literature requires extensive datasets for training that are provided in the form of artificial (simulated) fractured objects of very low fidelity and unreliable and simplified mechanics, whose results significantly deviate from real-world fractured objects. However, striving to enrich the availability of reliable data for training, in this work we provide a pre-labeled set of fragmented objects ready for use in segmentation and alignment experiments. The interest reader may refer to the comprehensive reassembly survey by Lu et al. [LLH*25] for more information on neural approaches.

It is evident from the formulation of the commonly used descriptors in the segmentation and labeling tasks for fragmentary material that the choice of scale (ranges) is critical to successfully capture the micro-scale of a fractured surface. Differences in material (see Figure 2), post-fracture exposure to weathering and even digitization itself (sampling, errors, units) require the adjustment of parameters, hindering fully automated pipelines. To this end, in this paper, we attempt to remove detector scale from the parameterization of the segmenter/classifier.

3. Method Overview

The proposed geometric processing pipeline provides a unified framework for multi-scale, per-vertex surface descriptor computation and subsequent macro-scale structural mesh segmentation. Designed to identify complex geometric features, such as locally rough regions and fractures, the pipeline operates in the following major steps:

- **Geometry preparation**. The input geometry is assumed to be represented as a triangular mesh. If necessary, simplify the mesh, build triangle connectivity and check manifold correctness.

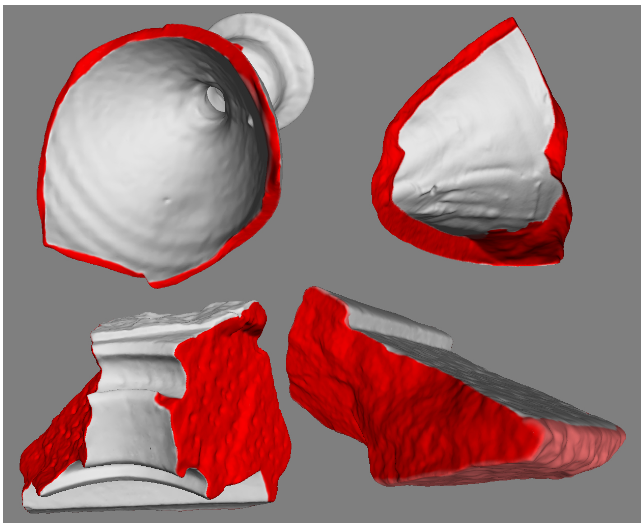

Figure 2: *Fracture relief for objects of different type of material. Top: Two variations of clay. Bottom left: soapstone. Bottom right: white marble.*

- **Micro-cluster extraction**. For each mesh vertex $\mathbf{v}_i$ build a cluster $\mathcal{C}_i$ containing the k geometrically closest vertices to $\mathbf{v}_i$. $\mathcal{C}_i$ clusters are by design overlapping and are used for the estimation of statistical quantities necessary for the detector radius estimation.
- **Fracture detector scale (radius) estimation**. The pipeline estimates $\hat{V}_r(\mathbf{x})$ (see Section 2.1) at multiple scales, using a continuous domain stochastic integrator for each radius r. We sweep a wide range of candidate radii and determine the volume descriptor radius for which variance between statistics of different clusters is maximized, indicating a peak in its discriminative power.
- **Fracture confidence estimation**. Each vertex is then attributed a fracture confidence score based on the micro-cluster statistics for $\mathcal{C}_i$ using the globally adjusted radius r^* (see Section 4.6).
- **Segmentation and classification**. An optional segmentation stage partitions the mesh into topologically continuous structural components using an orientation-driven region-growing approach based on normal similarity (see Section 4.7), similar to other methods in the literature (see related work section). A binary label is appointed to each segment signifying it being fractured or not.

An overview of the pipeline is presented in Figure 3.

In this work we rely on the volume integral descriptor, but the idea of adaptively determining the proper descriptor scale for fracture determination can be extended to other local descriptors, such as local bending energy.

4. Method Details

In this section we provide a detailed description of the individual stages involved in the vertex labeling and surface segmentation.

4.1. Input and Geometry preparation

To maintain computationally feasible execution times, we ensure that the total triangle count of the input mesh is reasonably bounded, targeting approximately 200–500 K elements. The (optional) decimation pass is efficiently executed using the public *Meshoptimizer* open-source library [Kap26]. The target number of polygons has an indirect implication on the parameterization of the method and in particular the maximum allowed number of micro-cluster polygons. A higher mesh resolution implies a denser sampling and therefore the need to define larger clusters given a fixed expected cluster radius. To avoid introducing a hyperparameter for this step, we constrain the mesh resolution, so that the micro-cluster size can be fixed.

We also perform a topological verification step to identify and resolve disconnected components resulting from duplicate or unmerged vertices. Although not critical, ensuring topological consistency is beneficial to a high-quality labeling of the surfaces, especially if segmentation is involved; the downstream micro-cluster generation phase relies heavily on structural neighborhood adjacency across valid manifold paths. A badly formed mesh will skew the micro-cluster extents, introducing some small bias to the statistics computation in the neighborhood of each vertex.

4.2. Volume Descriptor Estimator

Our approach requires the repetitive computation of a (normalized) volume descriptor $\hat{V}_r(\mathbf{x})$ for the vertices $\mathbf{x}$ at various radii r. In contrast to conventional techniques that deploy complex, custom acceleration data structures (ADS) to evaluate signed distance fields, occupancy grids or nearest-neighbor queries, our pipeline directly leverages native GPU hardware acceleration to solve visibility queries via ray tracing, eliminating the discretization error or computational overhead when utilizing custom spatial partitioning structures and intersection queries [PWHY09].

We efficiently estimate the localized volume descriptor $V_r(\mathbf{x})$ of a designated radius r at a surface point $\mathbf{x}$ by employing stochastic Monte Carlo integration over the ball of radius r. Let N_s denote the total number of samples distributed within the spherical interior via stratified volume sampling [HN04, SJP06]. From each randomly generated point in the unit ball $\mathbf{o}_j$, a ray is cast along an arbitrary direction $\mathbf{d}_j$ for $j = 1, \ldots, N_s$. If a ray $\{\mathbf{x} + r\mathbf{o}_j, \mathbf{d}_j\}$ misses the geometry entirely or intersects the surface from outside, as indicated by a negative dot product with the hit-point normal $\mathbf{n}_j$ (i.e., $\mathbf{d}_j \cdot \mathbf{n}_j \leq 0$), the corresponding sample is classified as external, otherwise $\mathbf{o}_j$ is an interior sample. The final normalized volume integral is evaluated as:

$$\hat{V}_r(\mathbf{x}) = 1 - \frac{1}{N_s} \sum_{j=1}^{N_s} \chi(\mathbf{x} + r\mathbf{o}_j, \mathbf{d}_j), \tag{5}$$

where the indicator function $\chi(\cdot)$ equals 1 when the traced ray hits a surface from the inside:

$$\chi(\mathbf{x} + r\mathbf{o}_j, \mathbf{d}_j) = \begin{cases} 1, & \mathbf{d}_j \cdot \mathbf{n}_j > 0, \\ 0, & \text{otherwise}. \end{cases} \tag{6}$$

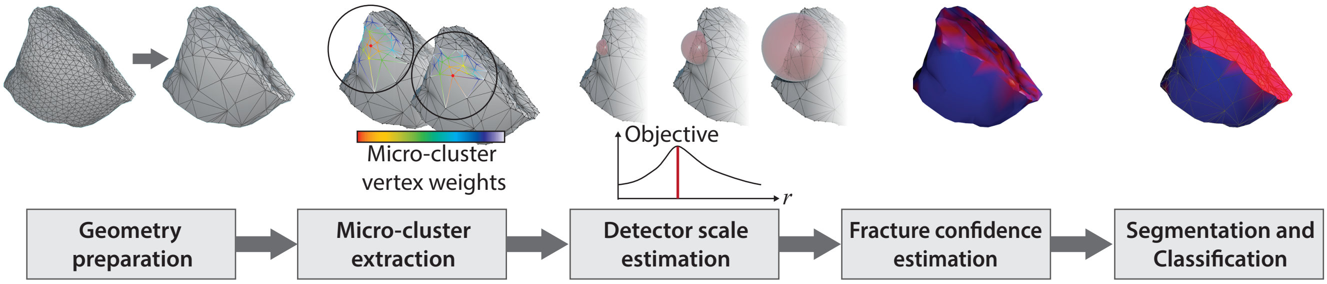

Figure 3: *The adaptive surface classification pipeline. The provided example is over-simplified for clarity.*

$\hat{V}_r(\mathbf{x})$ outputs a bounded scalar value in the range $(0,1)$. Values close to 0 signify a highly convex local surface neighborhood, whereas values approaching 1 reveal a highly concave neighborhood profile. 0.5 represents a flat local manifold. Boundary values of 0 and 1 signify a folded, self-overlapping geometry.

In this work (and in surface roughness detection, in general), we are interested in accentuating sharp features. However the volume descriptor does not provide sufficient contrast, in the same way the mean curvature is grading slowly with respect to local surface slope. To enhance its response to prominent features, we instead use a descriptor that filters $\hat{V}_r$ through a non-linear *smoothstep* function to enforce a steeper transition across geometric profiles:

$$D_r(\mathbf{x}) = \text{smoothstep}\left(0, 1, \hat{V}_r(\mathbf{x})\right). \tag{7}$$

Our implementation executes these parallelized visibility queries by leveraging the NVIDIA OptiX [PBD*10] GPU ray tracing architecture on a per-vertex basis. The adaptive mechanism for adjusting the bounding ball radius r_i to accommodate individual mesh variations is detailed in Section 4.6. To minimize variance, we allow the correlation of the sampled rays by reusing the same ball samples and ray directions $\{\mathbf{o}_j, \mathbf{d}_j\}$.

In our experiments, we use 1024 rays per volume descriptor computation, a number which is more than sufficient in terms of accuracy. Alternatively, an iterative progressive adaptive sampling, batched across all computed descriptors, can be employed.

4.3. Vertex Micro-Clusters

To evaluate localized statistics for each fragment vertex $\mathbf{x}_i$, we construct neighborhood configurations centered at $\mathbf{x}_i$, which we call *micro-clusters* $\mathcal{C}$ for the rest of the paper. We develop the micro-cluster $\mathcal{C}_i$ and topologically searching for candidate vertices $\mathbf{x}_j$ in the neighborhood of $\mathbf{x}_i$ via the triangle connectivity. A candidate neighbor vertex $\mathbf{x}_j$ is integrated into $\mathcal{C}_i$ until a maximum percentage of the model's surface area is reached. This is set to 1% of the model's surface area. This propagation terminates once the cluster size reaches a population limit $M_{\max} = \max\{0.005\% \cdot M_{tr}, 16\}$, where M_{tr} is the number of polygons in the model. This limit prevents fragments of different size to form micro-clusters of incompatible radius. The computed extents of the micro-cluster are expressed via its updated radius:

$$\tilde{\rho}_i = \max_{\mathbf{x} \in \mathcal{C}_i} ||\mathbf{x} - \mathbf{x}_i||_2. \tag{8}$$

Also note that $\mathcal{C}_i$ overlap and there are no disjoint sets, since we construct these sets for each vertex of the input mesh separately. Finally, to create roughly disk-shaped clusters, we push vertices into the cluster group using a priority queue based on distance from the seed $\mathbf{x}_i$.

4.4. Micro-cluster Statistics

Given the micro-cluster $\mathcal{C}_i$, we estimate local statistical properties to measure deviation patterns against the macroscopic topology. Each vertex $\mathbf{x}_j$ is accounted for in the statistical measure using a weight representing an exponential fall-off with respect to the sample's distance from the cluster seed $\mathbf{x}_i$:

$$w_i(\mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\,(\tilde{\rho}/3)^2}\right). \tag{9}$$

Applying a weighted contribution of micro-cluster members to the computed quantity makes the choice of a (relatively large) radius $\tilde{\rho}$ less critical.

For a given volume descriptor radius r_k, we compute the localized weighted mean μ_i and weighted local variance σ_i^2 across the cluster:

$$\mu(\mathcal{C}_i; r_k) = \frac{\sum_{j \in \mathcal{C}_i} D_{r_k}(\mathbf{x}_j) \cdot w_i(\mathbf{x}_j)}{\sum_{j \in \mathcal{C}_i} w_i(\mathbf{x}_j)}, \tag{10}$$

$$\sigma^2(\mathcal{C}_i; r_k) = \frac{\sum_{j \in \mathcal{C}_i} (D_{r_k}(\mathbf{x}_j) - \mu_i)^2 \cdot w_i(\mathbf{x}_j)}{\sum_{j \in \mathcal{C}_i} w(\mathbf{x}_j)}. \tag{11}$$

4.5. Deriving the Global Radius

The discriminative power of a score function $s(\cdot)$ that attempt to characterize the vertices according to fracture confidence is directly dependent on the selection of the detector scale, which is here the radius of the volume descriptor. For instance, selecting an excessively large radius will dilute high-frequency surface features, incorrectly implying that most geometric regions are microscopically flat (i.e., $D_r \approx 0.5$). Conversely, choosing an overly narrow kernel makes the descriptor highly susceptible to discretization errors and point-sampling noise, which similarly yields a misleading signature. To resolve this scale-dependency, we compute statistical variations across multiple discrete bands and identify the highly dis-

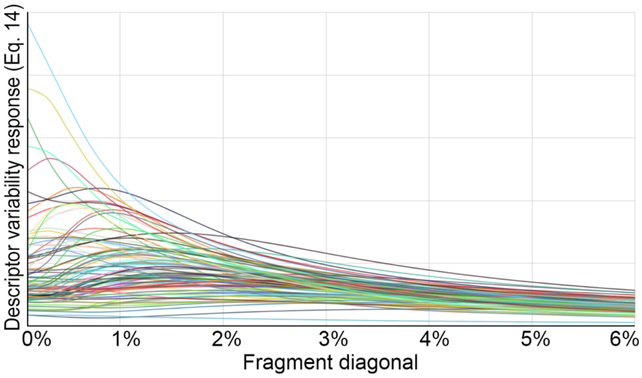

Figure 4: *The response curves of all the fragments in the benchmark dataset according to Equation 13.*

criminative model-specific radius configuration before executing the downstream vertex classification phase.

Our radius selection heuristic is based on a key morphological observation: surface fracture domains are inherently characterized by high-variance micro-surface variations. This implies that the volume descriptor $D_r(\mathbf{x}_i)$ will diverge significantly from the descriptors of its adjacent manifold neighbors within its localized micro-cluster. Leveraging this insight, we seek an optimal radius r^* that satisfies the following objective for the entire set of surface vertices $\mathbf{V}$:

$$r^* = \underset{r}{\operatorname{argmax}} \sum_{\mathbf{x} \in \mathbf{V}} h(r, \mathbf{x}), \tag{12}$$

where function $h(\cdot)$, captures the average local deviation multiplied by the standard deviation (Equation 11) at micro-cluster level defined as follows:

$$h : \{r \in \mathbb{R}, \mathbf{x} \in \mathbf{V}\} \mapsto \sigma(\mathcal{C}_i; r) \sum_{\mathbf{x}_j \in C_i} \frac{w_{x_i}(\mathbf{x}_j) \left| D_r(\mathbf{x}_i) - D_r(\mathbf{x}_j) \right|}{\sum_{\mathbf{x}_j \in \mathcal{C}_i} w_{x_i}(\mathbf{x}_j)}. \tag{13}$$

Intuitively, this optimization formulation aims to identify the specific radius parameter that maximizes the deviation between the descriptor at the central point of a cluster and the rest of the samples in the neighborhood, over all micro-clusters.

The selection of the optimal radius r^* can be efficiently implemented as an iterative process. In each step, an increasing radius r is used for the per-vertex estimation of the descriptor $D_r(\mathbf{x})$ and the evaluation of the error function of Equation 13. The search starts from a radius equal to 1% of the input model's AABB diagonal and advances with a step of 0.2% of the AABB diagonal until a local maximum response is detected. We use this radius as r^* and capture the corresponding localized surface statistics (Equations 7, 10, and 11). 1% is a sufficiently small fraction of the model's diagonal to account for the variation in size of the input fragments.

This discrete sampling strategy works reasonably well in practice because the response curve demonstrates a stable, roughly concave behavior around its global maximum. The measured mean deviation steadily increases as the volumetric domain expands from an overly compressed scale into a meaningful, feature-revealing profile. Once the scale exceeds this discriminative threshold, the average deviation begins to decrease. At larger radii the ball becomes large enough to encapsulate non-local, macroscopic features. We also demonstrate this behavior for each model in the dataset in Figure 4.

4.6. Surface Characterization

After fixing the observation scale r^*, the pipeline proceeds to characterize each discrete surface vertex based on its likelihood of belonging to a physical fracture zone. Intuitively, we define a continuous per-vertex scoring function $s : \mathbb{R}^3 \to [0,1]$ that represents our classification confidence (or structural importance) regarding whether a queried point resides within a fracture boundary. This continuous score is subsequently (and optionally) mapped into a discrete binary classification.

To compute the final scoring function $s(\cdot)$ we first compute Equation 13 for all the vertices of the input model using the derived radii r^*. Then, we compute the minimum and maximum values $\{h_{\min}, h_{\max}\}$ over all scores and finally, normalize all values to unit interval, namely:

$$s(\mathbf{x}; r^*) = \frac{h(r^*, \mathbf{x}) - h_{\min}}{h_{\max} - h_{\min}}. \tag{14}$$

This normalization is done so that $s(\cdot)$ can be directly used for sampling points based on their fracture score.

4.7. Surface Segmentation

Once the per-vertex confidence score is computed, the final phase of our pipeline partitions the global mesh topology into distinct, contiguous structural segments and classifies them at a macroscopic level. This process acts as a global regularizer, suppressing isolated false-positive vertex classifications by assessing the structural consensus of connected surface regions. Note that this step is optional, since $s(\cdot)$ can be directly used as per-vertex importance to drive an importance sampling of the fragment's surface during part alignment, completely dispensing with segmentation and classification. However, this step can be beneficial if we require compact class annotation over the model's surface. For this particular case directly using Equation 14 will produce noise results.

Our segmentation process attempts to evolve triangle clusters by tracking triangle-to-triangle adjacency to assemble continuous surface regions. It operates in three steps: boundary isolation, island extraction and island assimilation.

Boundary Isolation. We first isolate a temporary and not necessarily spatially connected boundary set of triangles $\mathcal{I}_0$, containing triangles at sharp macroscopic surface features. By first constructing such a set, we implicitly prevent subsequent evolving triangle islands from crossing natural object faces, leading to more meaningful segments. However, in contrast to [HFG*06] and [XZW*15], we do not strictly require the formation of such boundaries at locally maximal yet globally potentially insignificant curvature points, as the elements of $\mathcal{I}_0$ do not necessarily form connected paths.

A face f_i defined by vertices $\{v_{i0}, v_{i1}, v_{i2}\}$ is categorized into this boundary set if the mean volume descriptor score for the radius r^* among its vertices exceeds a threshold $\tau_{\text{high}} = 0.25$, although the exact value is not critical, as it only marginally affects the shape of the initially formed islands:

$$\max\left\{\left|\frac{1}{2}-\mu(\mathcal{C}_{i0})\right|,\left|\frac{1}{2}-\mu(\mathcal{C}_{i1})\right|,\left|\frac{1}{2}-\mu(\mathcal{C}_{i2})\right|\right\} \geq \tau_{\text{high}}. \quad (15)$$

Island formation. Our region-growing scheme builds upon the topological segmentation principles outlined in [AMP14]. The process initializes by evaluating triangle-to-triangle adjacency structures to aggregate continuous surface patches. An unassigned face minimizing the local patch variance $\sigma^2(\mathcal{C}_s; r^*)$ is selected as a cluster seed s. Neighboring unassigned triangles are iteratively appended to the expanding segment $\mathcal{I}_i$ provided that the candidate face normal $\mathbf{n}_j$ satisfies a local coplanarity constraint relative to the running segment average normal $\tilde{\mathbf{n}}_i$, governed by the threshold $\theta_{\text{running}} = 35^\circ$. To prevent the expansion front from bleeding across sharp features due to accumulator lag in the running average, we introduce a global anchor constraint against the seed normal $\mathbf{n}_s$ with a looser threshold $\theta_{\text{seed}} = 45^\circ$, yielding the joint criteria:

$$(\mathbf{n}_j \cdot \tilde{\mathbf{n}}_i \geq \cos\theta_{\text{running}}) \wedge (\mathbf{n}_j \cdot \mathbf{n}_s \geq \cos\theta_{\text{seed}}) \quad (16)$$

This dual-threshold formulation allows the cluster to adapt to high-frequency local surface wrinkles while establishing a strict directional boundary cone. Candidate faces exhibiting severe structural deflections from the seed orientation are immediately rejected, halting expansion at geometric boundaries. The loop terminates when all non-boundary faces are partitioned into distinct, contiguous islands $\mathcal{I}_k$ $(k \geq 1)$.

Hierarchical Cluster Assimilation. As explained in [PKT00], fine-grained region growing can over-segment complex surfaces and this is why we also perform a dual-pass hierarchical consolidation. First, any active structural segment $\mathcal{I}_k$ whose cumulative surface area accounts for less than 3% of the total mesh surface area is flagged for elimination. A small segment is dissolved and assimilated into the adjacent segment with the smallest normal dihedral angle.

Next, a boundary breakup loop dissolves the initial boundary set $\mathcal{I}_0$. Because $\mathcal{I}_0$ can include detailed fracture boundaries that we need to track as closely as possible, we iteratively inspect every face within $\mathcal{I}_0$. If a boundary face shares an edge with an already established non-boundary island $\mathcal{I}_k$ $(k > 0)$, it is detached from $\mathcal{I}_0$ and absorbed into that neighboring island. This peeling process repeats until $\mathcal{I}_0$ is empty.

Segment Classification. In the final step, we evaluate the classification for each unified island $\mathcal{I}_k$ evaluating the average score using Equation 14:

$$S_{\text{avg}}(\mathcal{I}_k) = \frac{1}{|\mathcal{I}_k|}\sum_{\mathbf{x}\in\mathcal{I}_k} s(\mathbf{x}), \quad (17)$$

normalizing based on the minimum and maximum scores:

$$S_{\min} = \min_{\mathcal{I}_k} S_{\text{avg}}(\mathcal{I}_k),\ S_{\max} = \max_{\mathcal{I}_k} S_{\text{avg}}(\mathcal{I}_k), \quad (18)$$

and assigning a binary class, namely:

$$c(\mathcal{I}_k) = \begin{cases} 1, & \frac{S_{\text{avg}}(\mathcal{I}_k)-S_{\min}}{S_{\max}-S_{\min}} \geq 0.5, \\ 0, & \text{otherwise.} \end{cases} \quad (19)$$

A class value of 1 represents a member of the fracture class, while 0 implies an intact region.

5. Evaluation

We provide here an evaluation of our method against an open reference dataset of pre-labeled fractured parts, which we also contribute for the evaluation of future methods.

5.1. The Dataset

In order to be able to evaluate our surface classification method, but also to provide a reference for future algorithms, we have created an extensible repository of real fractured and annotated objects, initially comprising 101 pieces, all of which are summarized in Figure 5. Most of the objects have been captured using an Artec 3D Eva scanner, with the exception of the small vessel, which has been digitized using photogrammetry. The fractured objects have been processed to reduce their polygon count so that no superfluous samples are present, which contribute no information about the structure of the piece. All pieces are stored in binary glTF format and bear a base color texture, embedded in the file. Although color is not used by our purely geometric method, future approaches may benefit from it.

Fracture confidence is baked on a separate texture to decouple mesh resolution from label data and allow researchers to paint their own confidence weights. The initial label data in the repository were produced with a custom tool that assigned weights according to the proximity of a vertex on a fragment to another fragment surface in the assembled object (see Figure 6), followed by manual editing in Adobe Substance Painter [Ado26]. The tool source code can be downloaded from GitHub at: `https://github.com/cgaueb/baker`.

By default, the confidence weights are grayscale values stored as RGB data in the provided PNG files. However, we assume that the information is only carried in the red channel, freeing G and B for additional label information in future versions. The labeled fractured objects repository is available at: `https://cgaueb.github.io/fracturedobjects/`. Researchers are welcome to contribute to expand the variety and number of cases.

5.2. Experiments

In this section, we present a thorough quantitative and qualitative evaluation of our proposed procedural segmentation pipeline. We evaluated the performance of the adaptive multi-scale volumetric descriptor framework against the structurally diverse set of real-world physical objects presented in the previous Section. All our experiments were conducted on a workstation equipped with an Intel i9-14900K CPU and an NVIDIA RTX 5090 GPU. The source

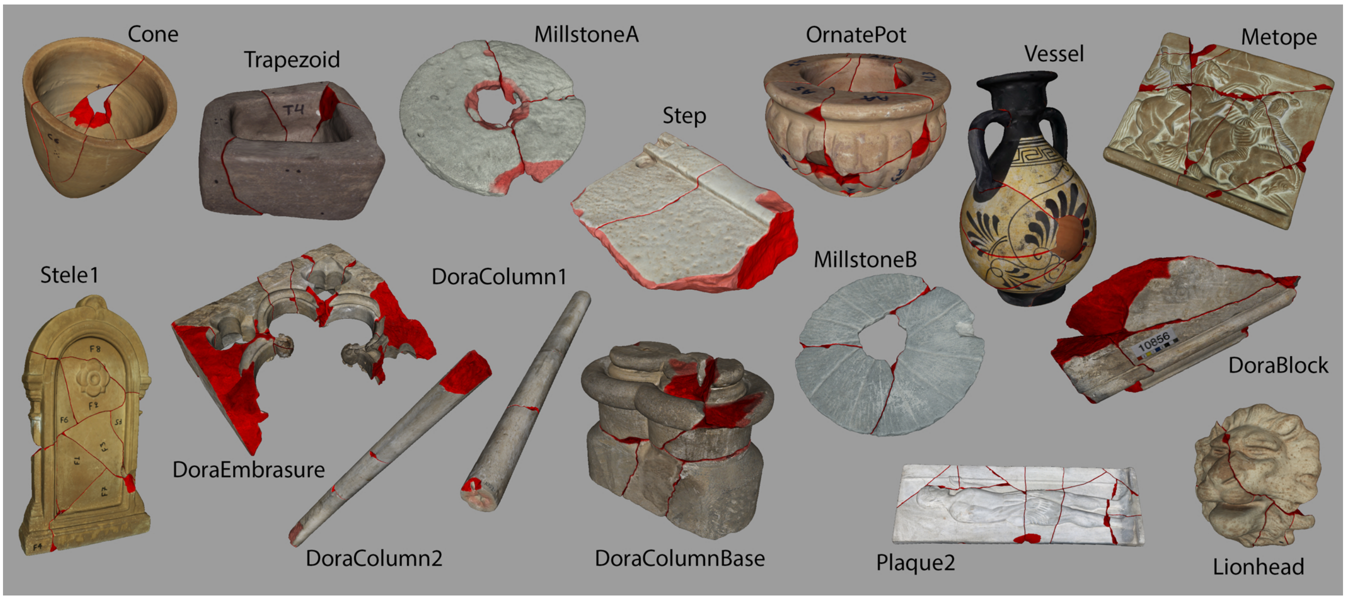

Figure 5: *The 101 fragments in the original dataset repository grouped by object and assembled. Fracture confidence mask is overlaid in red.*

Figure 6: *Screenshot of our fracture mask baking tool.*

code is available at https://github.com/cgaueb/GCH_segmenter.git.

We compare the performance of our framework against optimized implementations of the region-growing (RG) and hierarchical agglomerative (HA) segmentation methods demonstrated by [AMP14]. Because both baseline approaches rely heavily on a hand-tuned parameter to classify correctly the fractured parts, we established a unified evaluation methodology to ensure a fair comparison. Specifically, we first computed the per-vertex local bending energy formulated in Equation 4. For consistency, this bending energy was evaluated at the micro-cluster level using the exact structural configurations and hyperparameters defined in our approach (see Section 4.3). The resulting energy scores were subsequently normalized following Equation 14, and the final binary fracture annotations were generated using the classification pipeline detailed in Section 4.7 and Equation 19.

5.2.1. Qualitative Evaluation

For the qualitative evaluation, we converted the provided ground-truth masks into binary classification scores. We then evaluated the output against both the baseline segmenters and our method using standard confusion matrix metrics. Indicative results from our adaptive-scale segmentation method are presented in Figure 7.

It is important to note here that the provided classification masks are non-binary, as there are ambiguous zones on the fragments with respect to their class. To remove uncertainty from the classification correctness, in our qualitative measurements, we completely exclude points where the confidence score is not exactly 0 or 1.

Figure 8 illustrates the quality scores for all methods, averaged per object. As supported by the statistics, our method consistently improves the quality of the fractured segments across all test cases.

In Figure 9, we present a more detailed analysis of the recall and intersection-over-union scores of the fractured areas, since these metrics are more related to the task of fragment reassembly. As can be deduced from some low-scoring objects, there are indeed certain parts for which segments were misclassified, due to non-fractured regions having more craggy geometric detail than the fractured ones. Such indicative cases are presented in Figure 10.

5.2.2. Performance Evaluation

We have measured the execution time of our segmentation method against the region growing (RG) and hierarchical agglomerative clustering (HAC) reference methods. Our method is consistently an order of magnitude faster than the region growing segmenter and two orders of magnitude faster than the HAC variant, with end-to-end pipeline time below or up to ~ 1 sec and fragments of up to 445K triangles.

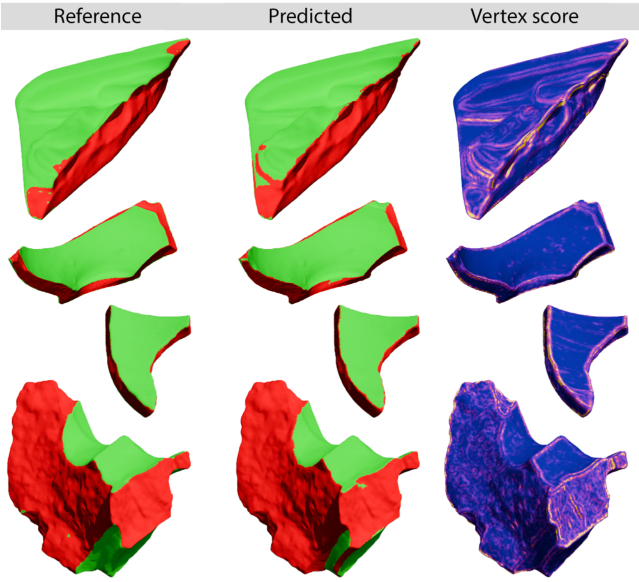

Figure 7: *Indicative examples of segmentation results from different types of fragments. First and second column illustrate the reference and predicted fragments with red color indicating the fractured surface and green the intact. Third column is the associate heatmap generated from the per-vertex scores (see Equation 14).*

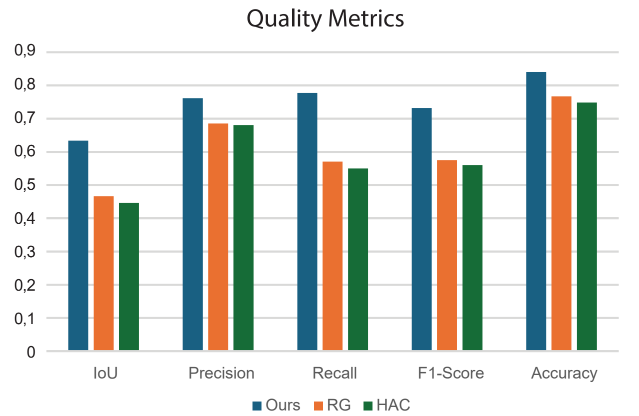

Figure 8: *Quality metrics averaged over all individual artifacts comparing our proposed method, the classification based on the region growing algorithm (RG) and the hierarchical agglomerative approach (HA) pipelines from [AMP14].*

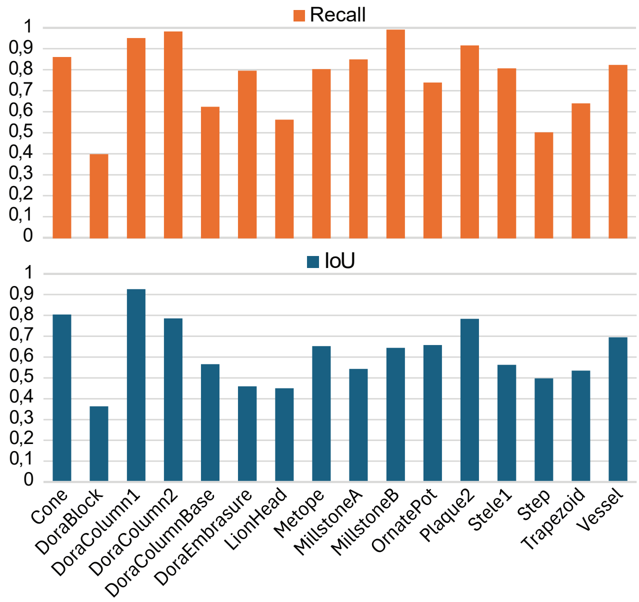

Figure 9: *Average intersection-over-union and recall statistics for the individual objects in the benchmark dataset.*

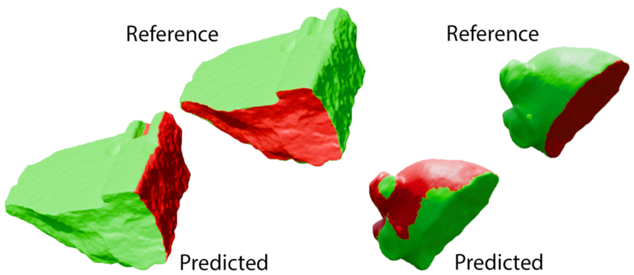

Figure 10: *Indicative failure cases. Left: Fractured side has less pronounced micro-structure than other regions, at the same scale. Right: Small-scale decorative features prevail over a clean cut. Red color indicates fractured surface area.*

5.2.3. Ablation study

To confirm the design choices of our adaptive surface classification schema we incrementally adjust the algorithm and report the same quality metrics as in Section 5.2.1. Specifically we start from the baseline surface classification algorithm which applies a fixed radius to evaluate the volume descriptors instead of the methodology derived in Equation 12 (AR), and does not use the weighting schema of Equation 9 (W). We abbreviate this method as **ours (-AR -W)**. For the fixed radius we use the average optimal radii over the whole dataset which is $r = 1.5\%$ of the diagonal of the AABB (see also Figure 4 for reference). We then test against the same approach but we include the adaptive radius **ours (-W)**. In Table 1 we summarize our results and present the improvements in quality.

In the next experiment, we vary the micro-cluster size discussed in Section 4.3. The maximum surface area ratio per micro-cluster remains a reasonable constant (1%). Setting the micro-cluster size in the range $\{2^i\}_{i=3}^{7}$ we have not recorded a significant performance variation, with the maximum drop in the IoU score being 0.60% relative to our fixed neighborhood of 2^4 vertices.

6. Conclusion and Future Work

In this paper, we presented a novel, procedural geometric processing pipeline for the automated surface classification and segmenta-

Method	IoU	Precision	Recall	F1-Score	Acc.
-AR-W	0.58	0.72	0.74	0.68	0.81
-W	0.59	0.73	0.74	0.69	0.82
Final	0.63	0.76	0.77	0.73	0.84

Table 1: *Ablation Study and Model Performance Metrics.*

tion of fractured cultural heritage artifacts. By designing an empirical multi-scale optimization scheme, our framework eliminates the traditional requirement for manual, per-object detector scale configuration, establishing a more robust and adaptive feature detection domain across artifacts of diverse material, density and morphology. Central to our method is a continuous-space volumetric descriptor estimator launched via hardware accelerated ray-tracing, which actively circumvents the accuracy and precision bottlenecks imposed by conventional discrete voxel or distance-field representations and provides the necessary performance to re-evaluate the descriptor fast during scale optimization.

Several avenues remain open for future development. First, we intend to explore the integration of our per-vertex confidence scoring function into global registration and sparse branch-and-bound optimization architecture to assess its impact on part reassembly times and pair-wise alignment precision. Second, we anticipate that our curated, high-fidelity real-world dataset will serve as a foundational benchmark for training un-biased neural assembly networks, ultimately bridging the current fidelity gap between simplified synthetic simulations and the chaotic physical realities of archaeological preservation.

Acknowledgments

The present research has received funding from the European Union in the framework of the ARTEMIS project, funded by the European Union under Grant Agreement No. 101188009.

References

[Ado26] ADOBE: Adobe Substance 3D Painter, 2026. version 12.0.3. URL: https://www.adobe.com/products/substance3d/apps/painter.html.

[AMP14] ANDREADIS A., MAVRIDIS P., PAPAIOANNOU G.: Facet extraction and classification for the reassembly of fractured 3D objects. In *Eurographics 2014 - Posters* (2014), The Eurographics Association. doi:10.2312/egp.20141060.

[APM16] ANDREADIS A., PAPAIOANNOU G., MAVRIDIS P.: Gpu accelerated computation of geometric descriptors in parametric space. In *Computer Vision, Imaging and Computer Graphics Theory and Applications* (Cham, 2016), Braz J., Pettré J., Richard P., Kerren A., Linsen L., Battiato S., Imai F., (Eds.), Springer International Publishing, pp. 41–61.

[BTP13] BOUAZIZ S., TAGLIASACCHI A., PAULY M.: Sparse iterative closest point. *Computer Graphics Forum 32*, 5 (2013), 113–123. doi:https://doi.org/10.1111/cgf.12178.

[GMGP05] GELFAND N., MITRA N. J., GUIBAS L. J., POTTMANN H.: Robust global registration. In *Proceedings of the Third Eurographics Symposium on Geometry Processing* (Goslar, DEU, 2005), SGP '05, Eurographics Association, p. 197–es.

[HFG*06] HUANG Q.-X., FLÖRY S., GELFAND N., HOFER M., POTTMANN H.: Reassembling fractured objects by geometric matching. *ACM Trans. Graph. 25*, 3 (July 2006), 569–578. doi:10.1145/1141911.1141925.

[HN04] HANNAY J. H., NYE J. F.: Fibonacci numerical integration on a sphere. *Journal of Physics A: Mathematical and General 37*, 48 (nov 2004), 11591. URL: https://doi.org/10.1088/0305-4470/37/48/005, doi:10.1088/0305-4470/37/48/005.

[Kap26] KAPOULKINE A.: meshoptimizer: Mesh optimization library that makes meshes fewer bytes and faster to render. https://github.com/zeux/meshoptimizer, 2026.

[LLH*25] LU J., LIANG Y., HAN H., HUA J., JIANG J., LI X., HUANG Q.: A survey on computational solutions for reconstructing complete objects by reassembling their fractured parts. *Computer Graphics Forum 44*, 2 (2025), e70081. doi:https://doi.org/10.1111/cgf.70081.

[PBD*10] PARKER S. G., BIGLER J., DIETRICH A., FRIEDRICH H., HOBEROCK J., LUEBKE D., MCALLISTER D., MCGUIRE M., MORLEY K., ROBISON A., STICH M.: Optix: A general purpose ray tracing engine. *ACM Trans. Graph. 29*, 4 (July 2010). URL: https://doi.org/10.1145/1778765.1778803.

[PKT00] PAPAIOANNOU G., KARABASSI E.-A., THEOHARIS T.: Segmentation and surface characterization of arbitrary 3d meshes for object reconstruction and recognition. In *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000* (2000), vol. 1, pp. 734–737 vol.1. doi:10.1109/ICPR.2000.905491.

[PKT02] PAPAIOANNOU G., KARABASSI E.-A., THEOHARIS T.: Reconstruction of three-dimensional objects through matching of their parts. *IEEE Transactions on Pattern Analysis and Machine Intelligence 24*, 1 (2002), 114–124. doi:10.1109/34.982888.

[PSA*17] PAPAIOANNOU G., SCHRECK T., ANDREADIS A., MAVRIDIS P., GREGOR R., SIPIRAN I., VARDIS K.: From reassembly to object completion: A complete systems pipeline. *J. Comput. Cult. Herit. 10*, 2 (Mar. 2017). doi:10.1145/3009905.

[PWHY09] POTTMANN H., WALLNER J., HUANG Q.-X., YANG Y.-L.: Integral invariants for robust geometry processing. *Computer Aided Geometric Design 26*, 1 (2009), 37–60. doi:https://doi.org/10.1016/j.cagd.2008.01.002.

[PWY*07] POTTMANN H., WALLNER J., YANG Y.-L., LAI Y.-K., HU S.-M.: Principal curvatures from the integral invariant viewpoint. *Computer Aided Geometric Design 24*, 8 (2007), 428–442. Discrete Differential Geometry. doi:https://doi.org/10.1016/j.cagd.2007.07.004.

[SCW*22] SELLÁN S., CHEN Y.-C., WU Z., GARG A., JACOBSON A.: Breaking bad: a dataset for geometric fracture and reassembly. In *Proceedings of the 36th International Conference on Neural Information Processing Systems* (Red Hook, NY, USA, 2022), NIPS '22, Curran Associates Inc.

[SJP06] SWINBANK R., JAMES PURSER R.: Fibonacci grids: A novel approach to global modelling. *Quarterly Journal of the Royal Meteorological Society 132*, 619 (2006), 1769–1793. doi:https://doi.org/10.1256/qj.05.227.

[TP13] THEOHARIS T., PAPAIOANNOU G.: PRESIOUS 3d cultural heritage fragments, 2013. URL: https://www.ntnu.edu/presious/3d-data-sets.

[WW08] WINKELBACH S., WAHL F. M.: Pairwise matching of 3d fragments using cluster trees. *Int. J. Comput. Vision 78*, 1 (June 2008), 1–13. URL: https://doi.org/10.1007/s11263-007-0121-5, doi:10.1007/s11263-007-0121-5.

[XZW*15] XU J., ZHOU M., WU Z., SHUI W., ALI S.: Robust surface segmentation and edge feature lines extraction from fractured fragments of relics. *Journal of Computational Design and Engineering 2*, 2 (2015), 79–87. doi:https://doi.org/10.1016/j.jcde.2014.12.002.