



This is a repository copy of *Reassembly of fractured object using fragment topology*.

White Rose Research Online URL for this paper:
<https://eprints.whiterose.ac.uk/181957/>

Version: Accepted Version

Proceedings Paper:

Alzaid, A. and Dogramadzi, S. orcid.org/0000-0002-0009-7522 (2019) Reassembly of fractured object using fragment topology. In: 10th International Conference on Pattern Recognition Systems. 10th International Conference on Pattern Recognition Systems, 08-10 Jul 2019, Tours, France. IET Conference Proceedings (CP761). IET Digital Library , pp. 98-105. ISBN 9781839531088

<https://doi.org/10.1049/cp.2019.0256>

This paper is a postprint of a paper submitted to and accepted for publication in 10th International Conference on Pattern Recognition Systems and is subject to Institution of Engineering and Technology Copyright. The copy of record is available at the IET Digital Library : <https://doi.org/10.1049/cp.2019.0256>

Reuse

Items deposited in White Rose Research Online are protected by copyright, with all rights reserved unless indicated otherwise. They may be downloaded and/or printed for private study, or other acts as permitted by national copyright laws. The publisher or other rights holders may allow further reproduction and re-use of the full text version. This is indicated by the licence information on the White Rose Research Online record for the item.

Takedown

If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.



eprints@whiterose.ac.uk
<https://eprints.whiterose.ac.uk/>

Reassembly of Fractured Object Using Fragment Topology

Asma Alzaid* and Sanja Dogramadzi**

* Faculty of Engineering, University of Bristol, UK, {aa14624@bristol.ac.uk.}

** Faculty of Engineering, University of the West of England {sanja.dogramadzi@bri.ac.uk}

Keywords: 3D Reassembly, Broken object, Surface matching, Feature extraction .

Abstract

This work presents our results on reassembly of broken objects using a newly developed fragment topology and feature extraction methodology. The reassembly of broken objects is a common problem in different domains including computer-aided bone fracture reduction [1] and reassembly of broken artefacts [2].

The new fragment topology combines information from intact and fractured region boundaries to reduce possible correspondences between the fragments and optimise our iterative matching process. Experiments performed on different multi-fragment objects show that the proposed topology can be effectively applied, completing the process in a small number of iterations and with average alignment error 0.12mm.

1 Introduction

The evolution of 3D object modelling and processing methods has improved the process of matching 3D fragments and reassembling broken objects. However, reassembling fractured objects can not be compared to the general 3D shape matching problem. The main reason is that the correlated fragments share matchable properties surrounding the fractured area only in contrast to the full or partial matching problem where the overlap patterns are usually more notable. Thus the correlations between fractured pieces are weak and hard to define. Also, erosion and missing parts of fragments can complicate the problem significantly.

Two approaches have been introduced in the literature to solve the reassembling problem [2]. The first approach attempts to match the fragments to a template model, while the second one depends only on the fragments geometries to define relations between fragments and return the original object. These approaches focus on different regions of the fragment model i.e intact regions, the outer surfaces of the fragment, and fractured regions, the surfaces that are generated when the object is broken. Figure 1 shows the intact and fractured regions. The template approach uses the intact regions of the model to locate the fragments in a template model. However, the template model of the broken object is not always available. Methods on the second approach usually focus on the fractured sides of the fragment. Indeed, using the fractured surface to assem-

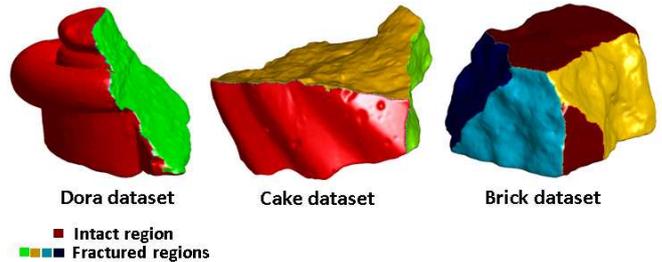


Figure 1. Examples of intact and fractured facets for different fragments.

ble the fractured object depends on the existence of salient features on the fractured side. These salient properties are difficult to define and might not be generated when the object is broken which leads to a large number of possible matching fragments and increases the complexity of finding the correct correspondence pairs.

We aim to tackle the above problems by introducing a new representation of fragments. This representation is inspired by the manual assembly of broken objects. When an expert attempts to reconstruct the original broken object, his main focus is on the fractured boundary of each fragmented piece. In addition, the surrounding intact area of fractured boundary might provide further clues about the corresponding fragment. From these observations we define a fragment representation by decomposing the fragment's surface into intact and fractured facets. Each facet will be represented with a graph node and two nodes link with edge if they are spatially adjacent. We refer to this representation as a fragment topology (Figure 2). This topology simplifies searching for potential matching fragments using both fractured facets and its adjacent intact properties. In addition, the orientation of matching fragment is easy to identify using the adjacent intact facets information of node i.e. matching fractured boundary and intact boundary.

The first step of this method is segmenting each fragment into intact and fractured facets using a new surface feature that improve the region merging and classification process. Each fragment is represented by using the introduced fragment topology, which describes fragments in terms of the relationship between the fractured and intact surfaces. This is followed by an iterative matching and representation process to find optimal matching pairs and form the original object.

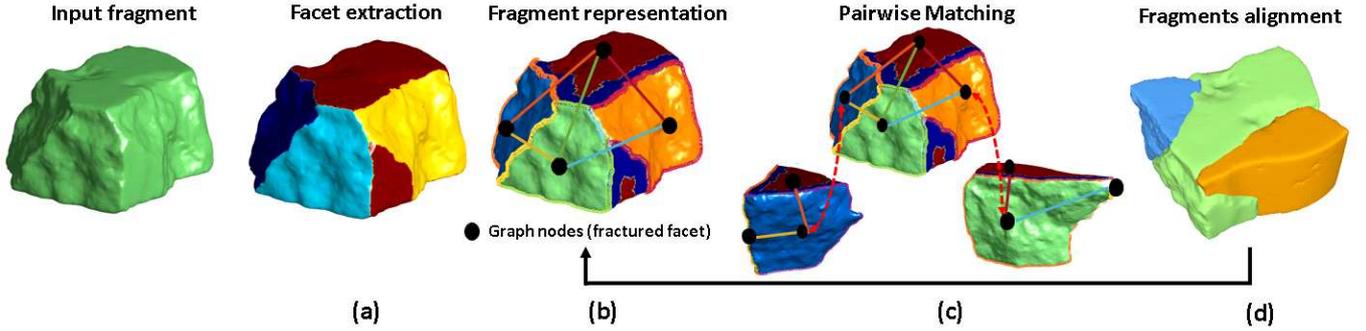


Figure 2. The general workflow of the proposed method. (a) segmentation of fragment surface into intact and fractured facets.(b) Creation of fragment topology and descriptors. (c) Matching between graphs node to find possible matches (d) Iterative Matching and representation of fragments.

2 Related Works

The existing method for reassembling fractured objects can be categorised into three main categories; reassembling based on fragment regions matching, reassembling using template as guidance to reconstruct the object and methods that depend on expert support.

Region-based matching methods - Those methods utilise the local surface properties to find similarities between adjacent fragments. Some approaches rely on the intact regions only to match fragments. For instance, Sagioglu et al. [3] introduce the idea of using the predicted region outside the border of fragments to find a correlation with the right neighbour. Other approaches consider the fracture regions to reconstruct the object. For thin shell objects such as pot or fresco fragments, the fracture region is commonly treated as a boundary curve of the fragment and the problem is reduced to 2D puzzle solving [4, 5, 6].

For general 3D fragments reassembly, Papaioannou et al. [7] extract the fracture regions by segmenting each fragment surface, then use the depth map for matching. Huang et al. [8] suggest a 4 steps framework to reassemble broken solid objects using integral invariants. However, matching small pieces might not produce an accurate result due to difficulties in segmenting small fragments into intact and fracture regions. To overcome this problem, Zhang et al. [9] combine template matching and fracture region matching. Yet, the template model is not available for all fractured objects. Rather than relying on feature computation, the alignment between two fragments is used such as in Mavridis et al. [10]. They define a matching score between two extracted fracture surfaces using a three-level coarse-to-fine search strategy that is based on the residual distance between fragments.

Template-based Models - In this category of methods, fragments are assembled by finding their best match to a template model using the intact surface information such as in skull reassembly [11, 12]. These methods mainly depend on the availability of the template model and can not construct a general model.

Interactive Expert-based Methods - User involvement for manual guidance is central to this category of reassembly

methods. Lee et al. [13] developed a semi-automatic tool for pelvic fracture reduction which integrates expert selection of matching points and the geometry driven for aligning the fragments. Another approach, by Mellado et al. [14] proposes a real-time interaction loop system that enables the user to approximate the initial position and orientations between two fragments and continuously correct and validate the pose. Many objects broke into a large number of pieces. Therefore, the user intervention might be time consuming and error prone.

When reassembling thick-shell fragments, the common methods use the properties of the fractured region to find the pairwise matching. This usually results in a large number of wrong pair matching due to weak discrimination properties in the fractured facets. However, to the best of our knowledge, no algorithm represents fragmented object by integrating the intact facet properties with the fractured boundary curves.

3 Proposed Assembly Approach

In this work, we introduce a novel, fully automatic method for reassembling fractured objects. We propose the fragment topology that represents the fragment’s surface based on its part arrangement. This topology is used to guide the search for possible matching fragments.

Given a set of fragments F_i , optimal matches are identified to reconstruct the original shape of the object. The matching has four main steps (Figure 2): 1- segmentation of fragment surfaces into intact and fracture facets (Section 3.1). 2- a graph representation of the fragments as facets and boundary edges is created (Section 3.2). 3- extraction of boundary curve features and other properties to measure the pairwise matching between fragments (Section 3.3). 4- Iterative assembly of the object by matching the selected corresponding pairs and updating the representation with the combined fragments (Section 3.4).

3.1 Fragment segmentation

The initial step in matching fragments is analysis of fragments’ surfaces and extraction of regions of interest. When the object breaks, new surfaces are generated that form a fractured region. Therefore a fragment’s surface can be categorised as: intact

and fractured. Each of these regions provides different characteristics that can support finding correct matching between the fragments. The intact regions identify continuity of patterns and geometries between fragment's surfaces, especially on the boundary areas close to fractured regions. On the other hand, the fractured regions define complementarity mating between pieces.

The segmentation process is performed to divide each fragment surface into distinct regions to avoid the wrong matches and reduce the computational effort. The primary goal in this step is extract segments to design a topology. Each segment should include either the intact or fractured surface but not both.

Our proposed workflow starts with an initial segmentation phase using Region Growing approach [15] with specific criteria. This step is followed by a merging phase that includes classification of regions into fractured and intact. The following sections explain the segmentation process.

3.1.1 Region Growing

Given a 3D mesh of a fragment, it starts by selecting a random face from the surface as a seed element and grows into a region by iteratively adding neighbours based on specific conditions. Let S be the seed element of the current segment and L be all unassigned neighbouring faces. The compatibility score D_{l_i} is calculated based on the angle between the average normal of the seed element and its 1-ring neighbouring face (n_s) and the normal vector of the examined face (n_{l_i}):

$$D_{l_i} = \cos^{-1}(n_s \cdot n_{l_i}) . \quad (1)$$

The local average normal n_s is:

$$n_s = \frac{\sum_{k \in S_r} n_k}{|S_r|} . \quad (2)$$

where n_k is the k^{th} face normal within 1-ring neighbouring of seed element S_r .

The above method works well on a planar surface but results in multiple segmentation on a fractured surface due to the presence of highly irregular surfaces (see Figure 3 (a)). Therefore, a post-processing step is required to improve the segmentation accuracy.

3.1.2 Region Merging

The region growing step provides an initial rough segmentation of a surface. In this phase, we categorise the segments based on two criteria: the surface area of the segment and the type of surface (fractured or intact). Rather than merging the segments and then classifying as in the existing approaches [8, 10, 16], we exploit the type of segment to perform the merging.

Based on the first merging criterion, we iteratively combine all small segments, with the segment area less than a threshold, to adjacent large segment based on a minimum compatibility score (Figure 3 (b)). The compatibility score is determined as follows:

$$S_{l_i} = \min(\cos^{-1}(n_{s_l} \cdot n_{s_s})) . \quad (3)$$

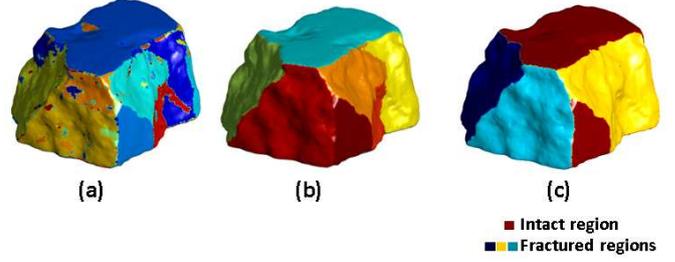


Figure 3. Segmentation step (a) Region growing (over segmentation problem) (b) Merging criterion (segment area) (d) Merging criterion (segment type + compatibility score)

where n_{s_l} is the average normal vector for the large adjacent region and n_{s_s} is the average normal vector for the small region.

The second merging criterion considers merging segments of the same type and within the compatibility score (Equation (3)). Broken objects vary in term of the intact surface. Fragments with planar surfaces are easy to classify, as opposed to, fragments with highly curved intact surfaces or detailed patterns. Therefore, we take the ratio of surface curvature for each segment and compare to the average curvature of the fragment surface to define the type of segment. For each fragment, we compute the average of maximum curvature (ϵ) as:

$$\epsilon = \frac{\sum_{k \in P} C_{max_k}}{|p|} . \quad (4)$$

Where C_{max_k} is the maximum curvature for point k in the fragment and $|p|$ is the number of points on the fragment. We use (ϵ) to define the ratio of maximum curvature in each segment as:

$$R_s = \frac{|C_{s_i} \geq \epsilon|}{|P_s|} . \quad (5)$$

where C_{s_i} is the maximum curvature of points P_s on segment s_i . Then, we cluster the resulted segments ratio (R_s) into two groups using k-mean algorithm. Clusters with a larger centroid value represent fractured segments.

3.2 Fragment Representation

In this work, we introduce a fragment topology to simplify fragment surface representation and guide the search for optimal pair matching. The matching fragment will have a similar topology in terms of intact and fractured facets. We seek to filter the wrong potential matches by defining simple properties along the fragment topology and find out how this can reduce the search for potentially matching fragments.

Given the segmented facets from the previous step, we define a graph $G = (V, E)$ where each node $n_i \in V$ is denoting a segmented facet f_i associated with a type of facet (intact or fractured) and the boundary curve of the fractured facet. The extraction and description of the boundary curve will be illustrated in Section 3.3. The edge $e_{i,j} \in E$ between nodes n_i and n_j denotes the boundary connecting two facets associated with

attributes such as the arc length of the boundary curve connecting two facets, and the start and end points of the boundary curve. Figure 2 shows an example of the fragment representation.

3.3 Feature Extraction

3.3.1 Boundary curve of fractured facet

As illustrated before, when trying to reassemble the broken object, our focus is on the boundary of the broken facet. Accordingly, we seek to define a measure of pairwise similarity based on these boundaries. We use multi-view of a fragment to extract 2D boundary curves. As the fractured facet is identified in the previous steps, we use the centroid point of a fracture facet as a viewpoint to project the facet boundary into the XY plane. The number of viewpoints is based on the number of fractured facets. Figure 4 shows an example of the extracted boundary curves.

3.3.2 Boundary curve descriptor

The extracted boundary curve is highly affected by noise and modelling variations. Also, the matching boundaries might be extracted in different orientations. To mitigate these problems, we describe boundary curves with a robust descriptor that is invariant to scale and rotation. Fourier descriptors show efficient descriptions for shapes that include scale, rotational and translational invariance [17]. The boundary points are first converted into complex numbers to extract the curve signature, as in the following:

$$S(k) = X(k) + jY(k) . \quad (6)$$

where $X(k)$ and $Y(k)$ represent x and y coordinates of the curve. Fast-Fourier transform of the boundary signature provides a Fourier descriptor and is defined as:

$$a(u) = \sum_{k=0}^{K-1} s(k)e^{-j2\pi uk/K} . \quad (7)$$

where $u = 0, 1, \dots, k - 1$.

In order to achieve translational invariance, the DC component of the Fourier descriptors is set to 0. For scale invariance, all the coefficients of the Fourier descriptor are normalised by the second coefficient:

$$a(0) = 0, \quad a(u) = \frac{a(u)}{a(1)} . \quad (8)$$

Both magnitude and phase values are considered however, phase values are affected by rotations and start points variations. For this reason, we use the topology representation of fractured facets to define rotations and starting points of the boundary curve.

3.4 Matching

The proposed method of reconstructing the final object is to define the whole to whole matching between the fragments,

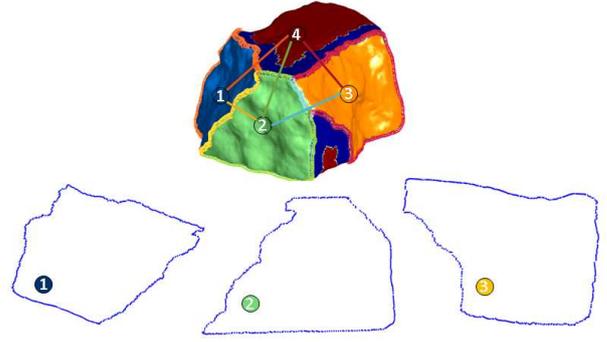


Figure 4. Examples of extracted 2D boundary curves.

combine the matching fragments group, recompute the new representation and iterate the matching until we find the final assembly of the fragments.

3.4.1 Pairwise Matching

We seek to simplify the process of finding potential match fragments list by integrating simple properties based on the defined representation. The typical way of searching for the potential matching fragment is based on finding similarities between extracted features points, which results in a large set of wrong potential matching points. This is typically followed with a refinement step such as using the RANSAC algorithm [18]. On the other hand, we propose to search for potential matching fragments using several factors: The area of the fractured facet. The topology of the fracture facet, for example, a number of intact and fractured neighbours and arc length of the boundary curve connecting intact and fractured facets. Boundary of fractured facets. The above geometrical properties are used to reduce the potential matching pairs. In order to identify exact matching pairs, we define a similarity score between possible pairs as the Euclidean distance of their Fourier descriptors of the boundary curve.

3.4.2 Multi-piece matching

All fragments are encoded with their extracted potential correspondences in a reassembly graph $G = (V, E)$. Each node $n_i \in V$, denoting a fragment F_i and each edge between the nodes n_i and n_j denotes the correspondences between the fragments. Each connected component in the graph will be considered as a whole fragment. So its new representation is recalculated and the reassembly graph is updated until all graph nodes are connected.

4 Experimental results

4.1 Facet extraction

We assess the proposed facet extraction method on different kind of fractured objects (see Table 1 for an overview). These objects are made of different materials and broken into various

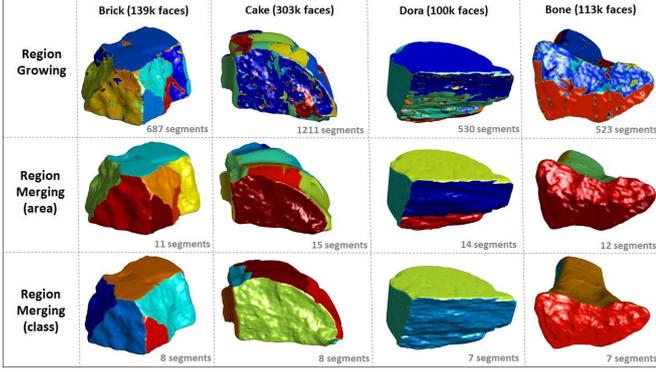


Figure 5. Result of facet extraction in each step for different datasets (see Table 1.)

sizes and shapes. Also, some fragments are exposed to weathering and erosion which change surface properties. The input to our method is a digital model of broken fragments. The number of vertices for each fragment is between 40k to 151k and the total number of tested fragments is 36. Figure 5 shows segmentation phases for each of 3 datasets we use.

Dataset	Model name	Type	# fragments	Vertex range
Vienna	Brick	stone	6	70k - 111k
	Cake	mortar	11	57k - 151k
	Sculpture	clay	7	95k - 198k
PRESIOUS	Nidaros Crypt Tombstone	stone	5	110-150
	Nidaros Cathedral Column Base	stone	5	40k - 70k
Bone Model	Femur	Foam cortical shell	2	57k - 64k

Table 1. Fractured objects used to evaluate the facet extraction methodology

As the aim of the facet extraction step is to identify the fractured facet, the evaluation of this step will examine the performance of segmentation and classification. For the segmentation, we measure the similarity between the manual segmentation of the fragment-surface and the resulted segmentation. The Dice Coefficient (DC) can be used to measure similarity by computing the spatial overlap between two sets of segmentation. The DC score ranges between 0 (not similar) and 1 (similar) and defines as follows:

$$DC(R_a^i, R_g^{i_t}) = \frac{2 \times ||R_a^i \cap R_g^{i_t}||}{||R_a^i|| + ||R_g^{i_t}||} . \quad (9)$$

Where R_a^i and $R_g^{i_t}$ are the automatic and manual segmentations, respectively, and i_t is the index of the closest segment from S_g to R_a^i which defined as:

$$i_t = \operatorname{argmax}_k (||R_a^i \cap R_g^k||) . \quad (10)$$

The DC between two segmentation S_a and S_g is computed as follow:

$$DC(S_a, S_g) = \frac{\sum_{i=1}^k DC(R_a^i, R_g^{i_t})}{k} . \quad (11)$$

To evaluate the efficiency of the fractured facet prediction model, we measure the overall accuracy (OA), the ratio between correctly predicted facet type and the total number of

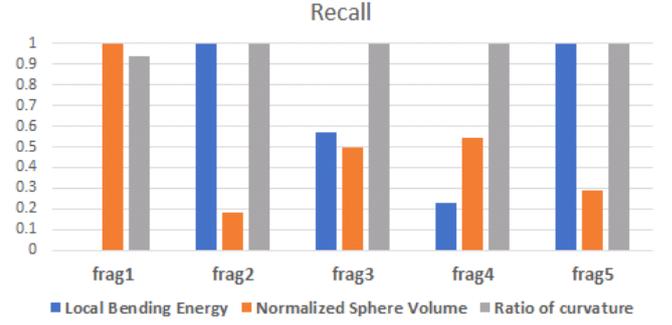


Figure 6. The Recall value for Huang et al. [8] (Local Bending Energy), PRESIOUS project method [10] (Normalized Sphere Volume) and our method (Ratio of curvature).

facets. The results are further assessed using Recall to measure the rate of fractured facets that classified correctly and is defined as:

$$Recall = \frac{||S_a^f \cap S_g^f||}{||S_g^f||} . \quad (12)$$

Where S_a^f represents the facets that classified as fractured and S_g^f is the ground truth of fractured facets.

Results - Figure 5 shows the results of the facet extraction phases for different types of fragments. The first row presents the Region Growing algorithm results with ($D_l = 5^\circ$). It works well on the flattened surface, whereas provides over-segmentation in the curved regions. Using the merging criteria;- the area and the class, the segmentation is improved and provides 0.87 DC score (for area threshold $< 2\%$ and $D_l = 5^\circ$). Choosing a good threshold is a difficult problem, therefore we attempt to choose it empirically.

In addition, the introduced feature for classifying fractured facet provides 82% overall accuracy and 97% of the fractured facets classified correctly. The comparison of our results to existing techniques [8, 10] using the same dataset is shown in Figure 6. Both methods require an expert intervention to adjust the parameters for segmenting and classifying the fractured facet. On the other hand, our method provides better results without user intervention. Figure 6 shows the Recall value for extracting fractured facet using Nidaros Cathedral dataset. Also, we examine our proposed method on several type of dataset, while Presious method [10] seems to works on type specific fractured objects.

4.2 Fragments assembly

Evaluating the reassembly of broken object method is restricted by the lack of the original model. In this work, we validate our approach using both simulated and real models. For the simulated models, we create different 3D models and shatter these models into a different configuration (See Figure 7 (a) as an example). We also evaluate our approach using real broken object provided in [8] (see Figure 7 (b)).

To compare the reassembly result to the original model, we should exclude the fractured regions from each fragment which

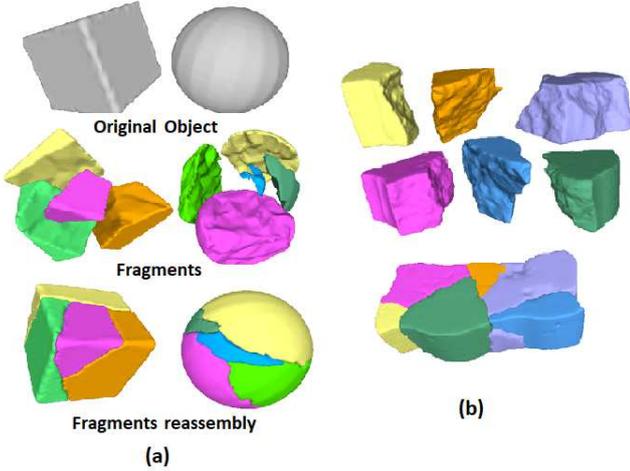


Figure 7. (a) Example of the simulated dataset. (b) Brick fractured model [8].

require a precise segmentation that is quite difficult to achieve and will add errors. Instead, we evaluate our method by comparing the original fragments assembly to the resulted assembly. For the simulated dataset, we first break the object and store the connected fragments as ground truth. Then, perform random transformations to simulate the shatter effects. Figure 7 shows the stages of generating the simulated dataset. For the real fractured object, we use the manually aligned fragments as ground truth.

To measure the accuracy of the reassembly we define the error measure (E_{ref}) as the root mean square error between the reference and reassembled pieces.

$$E_{ref} = \sqrt{\frac{\sum_{i=1}^N (F_{r_i} - F_{a_i})^2}{N}}. \quad (13)$$

Where F_{r_i} , F_{a_i} are the i^{th} point on the reference and aligned model respectively and N is the number of the points.

Results - Our method was ran on desktop with 3.60 GHz Core i7 CPU and 16 GB RAM. Table 2 shows the run-time of our method. The total computation requires 14 seconds for 6 fragmented objects and might increase depending on the number of fragments. In the reassembly process, the most time-consuming step is the search for potential matching fragments. The method of Huang et al. method [9] requires 2 seconds to set the potential matches for the brick fragments. Also, Son et al. [18] requires about 16 seconds to find potential matches. Our approach finds matching fragments in a brick model within 0.7 seconds. This is decreased by introducing the topology of the fragment, which restricts the number of possible matching combinations.

We first test our method on the simulated data. Figure 8 and 9 show the reconstruction of fragmented object 1 and object 2. These objects are shattered into a different configuration and composed of fragments that have a partial relationship to each other. The algorithm can identify the initial matching pieces correctly and reconstruct the final shape effectively. We mea-

Model	#V	#F	t_{rep} (s)	t_{pm} (s)	t_{mm} (s)
obj1	80k	3	1.34	0.1055	0.01
obj2	108k	4	2.3	0.32	0.07
brick	534k	6	13	0.7	0.13

Table 2. Performance of the method: (model name, number of vertices of all fragments, number of fragments and time in seconds for representation process of all fragments (t_{rep}), create potential matching (t_{pm}) and multi-piece matching(t_{mm}))

sure the pairwise alignment between the fragments based on the matched fractured boundary using the ICP method and provide efficient and accurate alignment with average error $E_{ref} = 0.047$ mm. Figure 8 and 9 show examples of the pairwise alignment between two fragments. We also examined our method on a real model (brick model) [8] fractured in six fragments. The broken brick model is affected by erosion and each fragment can be matched with one or more of the other fragments. Figure 10 (a) illustrates the multi-piece matching construction between the brick fragments. The introduced topology representation can achieve 0.19 mm matching accuracy after two iterations (Figure 10 (b)).

The previous reassembly methods, [8, 18], proposed a complex descriptor of fractured facets that requires a large number of discriminating points in order to accurately describe the facet. In both these methods, the potential pairwise matching process resulted in incorrect matches that required further refinement steps to reduce the possibility of wrong matches, leading to increased complexity of the algorithm and the matching time. To overcome this complexity, our work proposes a new representation that combines the fractured region boundary and its relation to adjacent intact facet to define matching fragments. Our initial results are obtained using a simple description of the intact facets.

5 Conclusion

We present a new method that combines intact and fractured regions properties of fragment for efficient reassembly of thick-shell pieces. The method represents each fragment by its fractured facets and its relation to the intact facets, then uses this information to define the matching fragments and reconstruct the final object by iterative merging of pairwise matching pieces.

Future work will include extending the fragment representation with more properties that can more accurately describe the intact surface. In addition, the method will be further evaluated on 3D model of fractured bones to test the method on more complex and noisy fragment shapes and define the limitations.

Acknowledgements

Asma is a PhD student who is sponsored by King Saud University, Saudi Arabia.

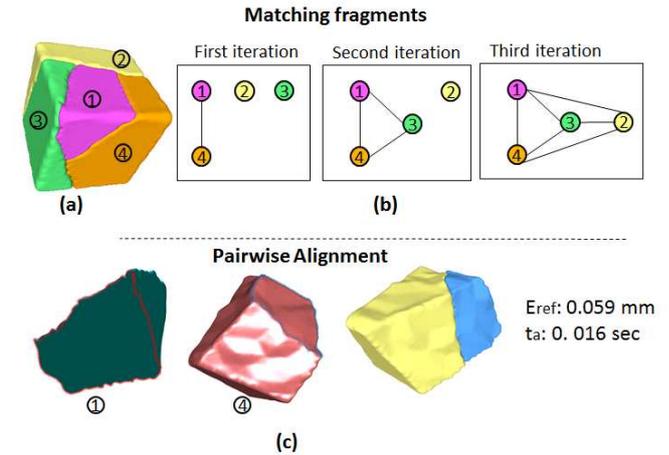
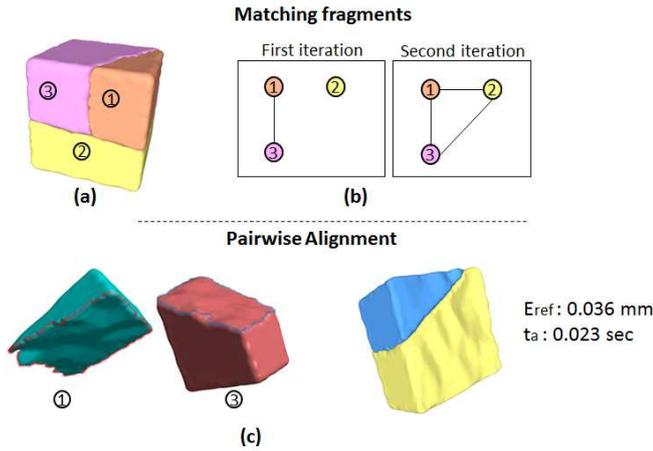


Figure 8. Object1 model: (a) Multi-piece reconstruction of the fragments. (b) iterations to reconstruct object. (c) Pairwise alignment, the resulted reference error (E_{ref}) and the alignment time (t_a).

Figure 9. Object2 model: (a) Multi-piece reconstruction of the fragments. (b) iterations to reconstruct object. (c) Pairwise alignment, the resulted reference error (E_{ref}) and the alignment time (t_a).

References

- [1] J. J. Jimenez-Delgado, F. Paulano-Godino, R. PulidoRam-Ramirez, and J. R. Jimenez-Perez, "Computer assisted preoperative planning of bone fracture reduction: Simulation techniques and new trends.," *Medical image analysis*, vol. 30, pp. 30–45, 2016.
- [2] G. Papaioannou, T. Schreck, A. Andreadis, P. Mavridis, R. Gregor, I. Sipiran, and K. Vardis, "From Reassembly to Object Completion," *Journal on Computing and Cultural Heritage*, vol. 10, pp. 1–22, mar 2017.
- [3] M. Sagioglu and A. Ercil, "A Texture Based Matching Approach for Automated Assembly of Puzzles," in *18th International Conference on Pattern Recognition (ICPR'06)*, pp. 1036–1041, IEEE, aug 2006.
- [4] W. Kong and B. B. Kimia, "On solving 2D and 3D puzzles using curve matching," *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 583–590, 2001.
- [5] C. Papaodysseus, T. Panagopoulos, M. Exarhos, C. Triantafillou, D. Fragoulis, and C. Doulmas, "Contour-shape based reconstruction of fragmented, 1600 BC wall paintings," *IEEE Transactions on Signal Processing*, vol. 50, pp. 1277–1288, jun 2002.
- [6] J. C. McBride and B. B. Kimia, "Archaeological Fragment Reconstruction Using Curve-Matching," in *2003 Conference on Computer Vision and Pattern Recognition Workshop*, pp. 3–3, IEEE, jun 2003.
- [7] G. Papaioannou, E.-A. Karabassi, and T. Theoharis, "Reconstruction of three-dimensional objects through matching of their parts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 1, pp. 114–124, 2002.

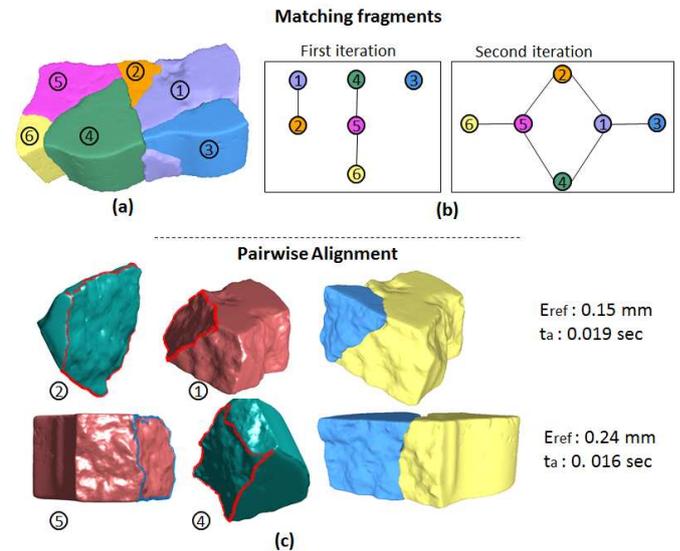


Figure 10. Brick model: (a) Multi-piece reconstruction of the fragments. (b) iterations to reconstruct object. (c) Pairwise alignment, the resulted reference error (E_{ref}) and the alignment time(t_a).

- [8] Q.-X. Huang, S. Flöry, N. Gelfand, M. Hofer, and H. Pottmann, "Reassembling fractured objects by geometric matching," in *ACM SIGGRAPH 2006 Papers on - SIGGRAPH '06*, vol. 25, p. 569, ACM, jul 2006.
- [9] K. Zhang, W. Yu, M. Manhein, W. Waggenspack, and X. Li, "3D Fragment Reassembly Using Integrated Template Guidance and Fracture-Region Matching," 2015.
- [10] P. Mavridis, A. Andreadis, and G. Papaioannou, "Fractured Object Reassembly via Robust Surface Registration," *Eurographics*, 2015.

- [11] Zhao Yin, L. Wei, X. Li, and M. Manhein, "An automatic assembly and completion framework for fragmented skulls," in *2011 International Conference on Computer Vision*, pp. 2532–2539, IEEE, nov 2011.
- [12] W. Yu, M. Li, and X. Li, "Fragmented skull modeling using heat kernels," *Graphical Models*, vol. 74, pp. 140–151, jul 2012.
- [13] P. Y. Lee, J. Y. Lai, S. A. Yu, C. Y. Huang, Y. S. Hu, and C. L. Feng, "Computer-assisted fracture reduction and fixation simulation for pelvic fractures," *Journal of Medical and Biological Engineering*, vol. 34, no. 4, pp. 368–376, 2014.
- [14] N. Mellado, P. Reuter, and C. Schlick, "Semi-automatic geometry-driven reassembly of fractured archeological objects," *VAST 2010: The 11th International Symposium on Virtual Reality, Archaeology and Cultural Heritage*, p. 00, sep 2010.
- [15] A. Mehnert and P. Jackway, "An improved seeded region growing algorithm," *Pattern Recognition Letters*, vol. 18, pp. 1065–1071, oct 1997.
- [16] T. P. Thomas, D. D. Anderson, A. R. Willis, P. Liu, M. C. Frank, J. L. Marsh, and T. D. Brown, "A computational/experimental platform for investigating three-dimensional puzzle solving of comminuted articular fractures," *Computer Methods in Biomechanics and Biomedical Engineering*, vol. 14, pp. 263–270, mar 2011.
- [17] C. Gonzalez Rafael, "Digital image processing, /rafael c. gonzalez, richard e. woods," 2008.
- [18] T.-g. Son, J. Lee, J. Lim, and K. Lee, "Reassembly of fractured objects using surface signature," *The Visual Computer*, vol. 34, pp. 1371–1381, oct 2018.