Facet Extraction and Classification for the Reassembly of Fractured 3D Objects

Anthousis Andreadis Pavlos Mavridis Georgios Papaioannou

Department of Informatics, Athens University of Economics & Business

Abstract

*The reassembly of fractured 3D objects is a critical problem in computational archaeology, and other application domains. An essential part of this problem is to distinguish the regions of the object that belong to the original surface from the fractured ones. A general strategy to solve this region classification problem is to first divide the surface of the object into distinct facets and then classify each one of them based on statistical properties. While many relevant algorithms have been previously proposed ([PKT01], [HFG*06], [WW08]), a comparative evaluation of some well-known segmentation strategies, when used in the context of such a problem, is absent from the bibliography. In this poster we present our ongoing work on the evaluation of the performance and quality of segmentation algorithms when operating on fractured objects. We also present a novel method for the classification of the segmented regions to intact and fractured, based on their statistical properties.*

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computer Graphics—Computational Geometry and Object Modeling.

1. Introduction

The automatic reassembly of fractured 3D objects can be greatly improved by a good partitioning of the fragments into fractured and intact facets. Not only it improves the efficiency of the process by drastically reducing the search space of potential matches, but it also avoids trivial and erroneous solutions. We solve this problem by first using a general segmentation algorithm to create an initial set of regions and then classifying each region as *intact* or *fragmented*. A post-processing step then improves the solution by merging suitable neighbouring regions.

The quality of a mesh segmentation is directly related to the requirements imposed by a particular application. In our case, for the reassembly of 3D objects, the desired segmentation should consist of coherent segments that can be strictly classified as either fractured or intact (no mixed regions). Furthermore, the number of the fractured regions should be as small as possible, since this directly affects the pair-wise combinations a matching algorithm performs.

Two aspects of a segmentation algorithm have a direct impact on the quality of the results: the ordering of operations and the distance metrics used in order to test the compatibility of two regions.

2. Ordering of Operations

Region Growing proceeds by developing a single region based on a seed element. The region grows by iteratively merging neighbouring elements that meet some user-defined criteria. When no other neighbours can be found, the process starts with a new seed and the algorithm terminates when all input elements belong to a region. We experimented with two variations of the algorithm, one that selects a random neighbour (RGN) and a second that selects the one with the closest metric distance (RGBF).

Hierarchical Agglomerative Clustering (HC) starts by considering each element as a region (cluster). Regions are then incrementally clustered, by merging the two regions with the closest distance, according to a user-specified metric. The merging process stops when no neighbor clusters meet the merging criteria. A custom 2-level caching scheme is used by our implementation in order to avoid redundant distance and sorting calculations.

Post-Processing. The greedy nature of the region growing algorithm can lead to severe over-segmentation. This is fixed by a custom post-processing algorithm that first decomposes small regions into single elements, which are subsequently merged to larger neighbouring segments. The same post-

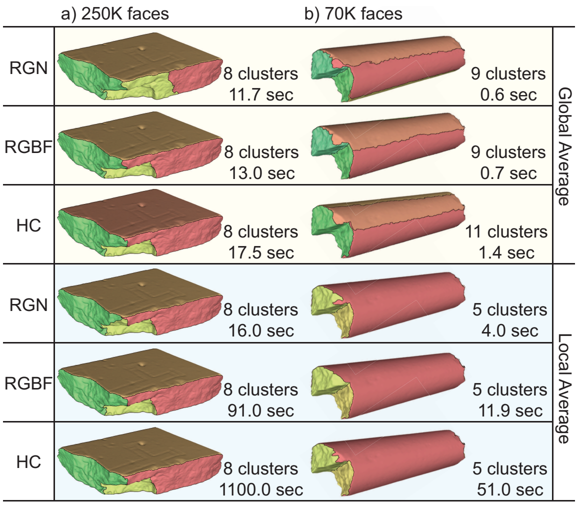

Figure 1: *Facet extraction using different merging order and criteria. Model a) has planar regions and both merging criteria result in similar segments. In contrast, model b) has a curved surface and there is a clear difference in the resulting segmentation.*

processing also improves the results of the hierarchical clustering method.

3. Distance Metrics

One of the simplest distance metrics for mesh segmentation is the angle between the average normal of two segments. While this metric performs well on planar surfaces, it results in over-segmentation on curved ones (see Figure 1). A simple way to properly handle both planar and curved surfaces is to use the average normal computed on the local neighbourhood on the boundary of the two segments. The size of this local neighbourhood is a parameter that determines the tendency of the algorithm to break curved regions to multiple segments. This local metric on the other hand significantly increases the computational complexity as each distance computation involves the determination of the local neighbourhood within a segment's borders, near the contact line with the adjacent segment.

4. Classification

Our classification method is based on the observation that fractured regions are typically more rough than the intact ones. To this end, we utilise the median value of the local variance of *sphere volume integral invariant*:

$$V_r(\mathbf{p}) = \frac{4\pi r^3}{3} - V_o(\mathbf{p}), \tag{1}$$

which represents the part of the sphere volume of radius r at point $\mathbf{p}$ "inside" the surface [PWHY09]. $V_o(\mathbf{p})$ is the open volume of the sphere and is computed using a fast approximation inspired by [MOBH11]. Assuming a smoothly varying tangential elevation around $\mathbf{p}$, vector $\mathbf{q}_i - \mathbf{p}$ from central point to any sample $\mathbf{q}_i$ within the Euclidean neighborhood $S(\mathbf{p}, r)$ approximates the horizon in this direction with respect to the normal vector $\mathbf{n}$ at $\mathbf{p}$ at a distance scale equal to $\|\mathbf{q}_i - \mathbf{p}\|$. Taking a uniform rotational and radial distribution of samples $\mathbf{q}_i$ in $S(\mathbf{p}, r)$, $V_o(p)$ is computed as:

$$V_o(\mathbf{p}) = \frac{4\pi r^3}{3N} \sum_{i=1}^{N} \frac{(\mathbf{q}_i - \mathbf{p})\mathbf{n}}{\|\mathbf{q}_i - \mathbf{p}\|}. \tag{2}$$

The classification algorithm should avoid both false negatives (fragmented regions classified as intact) and false positives (intact regions classified as fragmented). However, the first case will certainly prevent the matching algorithm to find a correct solution, while the second can potentially lead to an incorrect matching between two intact regions, however this is rarely the case in our experiments. Therefore, the threshold value used for classification is selected in order to favor false positives instead of false negatives.

5. Results and Conclusion

We tested the facet extraction and classification algorithms with 3D fragments from the archaeological site of the {removed for blind review purposes}. Using global average as the distance metric we can see that the naive region growing approach can give erroneous results, while objects with curved surfaces are oversegmented regardless of the ordering method. On the other hand, by using local average as the distance metric, the quality of segmentation for curved surfaces increases without affecting the planar ones, at the cost of polynomial growth of computations. The ordering of operations does not seem to affect the resulting segmentation in this case, favoring thus the naive approach due to the performance impact inflicted by a more sophisticated approach (see Figure 1). Finally we should note that a robust post-processing is essential for making region growing practical, since omitting this step leads to over-segmentation.

Acknowledgements. Datasets are from the PRESIOUS project data collection. This work was supported by EC FP7 STREP Project PRESIOUS, grant no. 600533.

References

[HFG*06] Huang Q.-X., Flöry S., Gelfand N., Hofer M., Pottmann H.: Reassembling fractured objects by geometric matching. In *ACM SIGGRAPH 2006 Papers* (2006), ACM, pp. 569–578. 1

[MOBH11] McGuire M., Osman B., Bukowski M., Hennessy P.: The alchemy screen-space ambient obscurance algorithm. In *High-Performance Graphics 2011* (August 2011). 2

[PKT01] Papaioannou G., Karabassi E.-A., Theoharis T.: Virtual archaeologist: Assembling the past. *Computer Graphics and Applications, IEEE 21*, 2 (2001), 53–59. 1

[PWHY09] Pottmann H., Wallner J., Huang Q.-X., Yang Y.-L.: Integral invariants for robust geometry processing. *Computer Aided Geometric Design 26* (2009), 37–60. 2

[WW08] Winkelbach S., Wahl F. M.: Pairwise matching of 3D fragments using cluster trees. *International Journal of Computer Vision 78* (2008), 1–13. 1