Region-based 3D artwork indexing and classification

M. Alcoverro, S. Philipp-Foliguet, M. Jordan, Laurent Najman, Jean Cousty

HAL Id: hal-00351635

https://hal.archives-ouvertes.fr/hal-00351635

Submitted on 9 Jan 2009

REGION-BASED 3D ARTWORK INDEXING AND CLASSIFICATION

M. Alcoverro, S. Philipp-Foliguet, M. Jordan

ETIS - UMR CNRS 8051
ENSEA / Université de Cergy-Pontoise
95014 Cergy-Pontoise Cedex – France

L. Najman, J. Cousty

Université Paris-Est
LABINFO-IGM – UMR CNRS 8049
77454 Marne-la-Vallée Cedex 2 – France

ABSTRACT

We present a method for 3D surface segmentation based on watershed cuts computed on local curvatures. The segmentation algorithm is applied to artwork database classification by mean of a search engine based on 3D region descriptor bags. The comparison with a search engine based on global descriptors clearly shows an improvement of performances.

Index Terms— 3D surface segmentation, watershed cut, 3D descriptors, region bags, artwork database, indexing.

1. INTRODUCTION

Partial shape matching and detection of self-similarities are some of the most active tasks in the context of 3D mesh indexing and retrieval. For instance, [1] present the "saliency" of a region computed from its relative size and some curvature measures, and use it for partial shape matching and shape alignment. Another recent work covers the identification of *most distinctive parts* of an object seen as a mesh database item: not only the local shape properties of a region are considered, but also how consistent they are with other instances of the same class and how different from objects of other classes, leading to a definition of regions *distinctive*, *not distinctive*, or *in the middle* for each mesh of the database [2, 3]. Some other approaches are based on the decomposition of 3D models and are well suited for CAD or simple shape objects, shape descriptors are then computed for each part of the model and compared in order to classify the database [4].

We present here a first application of the *watershed cuts*, a recently introduced framework [5, 6], to the segmentation of 3D surfaces and show how these segmentations combined with an appropriate use of 3D shape region descriptors and a search engine based on region descriptor bags improve indexing and retrieval task. This will be applied to a 3D model database dedicated to various artworks: greek vases, terracota gallo-roman figurines, moulds, fragments, *etc*.

2. 3D SURFACE SEGMENTATION

The 3D surface segmentation algorithm is based on the local curvature estimation, followed by a watershed process providing a partition of the 3D surface.

### 2.1. Local curvature computation

The *normal curvature* κ_n of a surface in some direction is the reciprocal of the radius of the circle that best approximates a normal slice of surface in that direction. The normal curvature can be expressed as $\kappa_n = \kappa_1 s^2 + \kappa_2 t^2$ where κ_1 and κ_2 are the *principal curvatures* and (s, t) are the *principal directions*, which are the directions where the normal curvature reaches its minimum and maximum. The principal curvatures are computed thanks to the Trimesh algorithm [7].

As the curvature map will be used to partition the surface by using the watershed operator, a single scalar function is desired. From (κ_1, κ_2) we compute the *mean curvature* $H = (\kappa_1 + \kappa_2)/2$ and we use $H_{inv} = (1/\pi)(\arctan(-H) + \pi/2)$ which behaves like the inverse of the mean curvature H and takes always positive values. It gives high values to concave zones and low values to convex zones. For manufactured objects, we consider the *max curvature* $\max(\kappa_1^2, \kappa_2^2)$, which gives high values on both convex and concave zones. The max curvature has also high values on zones that are flat in one direction, and convex or concave in the other. These zones are commonly the edges that divide planes of an object, as the division between the roof and the doors of a car. In fig. 1 are depicted the values of these scalar functions in gray scale for the sculpture left. Low values are black, while great values are white.

2.2. Watershed and filtering

2.2.1. Watershed cuts

In recent papers [5, 6], we investigate the watersheds in a framework different from the one of vertex-weighted graphs: we consider a graph whose edges are weighted by a cost function. A watershed of a topographic surface may be thought of as a separating line-set on which a drop of water can flow down toward several minima. Following this intuitive idea, we introduce in [5] the definition of watershed cuts in edge-weighted graphs. We establish the consistency (with respect to characterizations of the catchment basins and dividing lines)
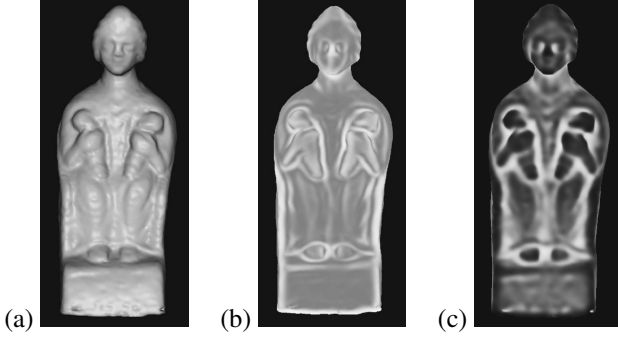
Fig. 1. Curvature scalar functions: (a) 3D object; (b) max curvature; (c) inverse curvature.

of watershed cuts, prove their optimality (in terms of minimum spanning forests) and propose efficient linear algorithms to compute them. As far as we know, the framework of edge-weighted graphs is the only generic discrete framerwork in which all these properties hold true. In particular, it is the first framework in which the drop of water principle is used as a definition for watershed and in which this principle leads to fast algorithms and an optimality theorem.

2.2.2. Watershed cuts on curvature meshes

Watershed cuts can be extended [8] to *simplicial complexes*, and especially manifold-like meshes. Consider a 3D surface mesh M (composed of *triangles*, *sides of triangles* and *points*) so that for any side e in M there is exactly one pair of triangles (g, h) such that $e \in g$ and $e \in h$. We build a graph with one vertex for each triangle of M and an edge connecting two vertices if the corresponding triangles share a side (fig. 2(b)).
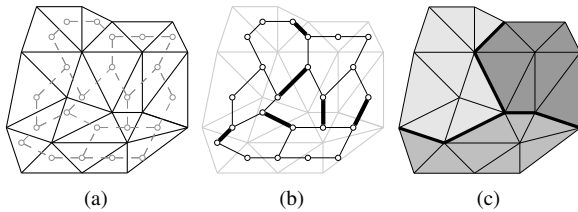
Fig. 2. (a) A triangle mesh. (b) Segmentation on edges of the graph (in bold). (c) Segmentation on the mesh.

To compute a watershed cut, we need a map on the edges. Let e be any side of a triangle in M and (x, y) the pair of points such that $e = \{x, y\}$. As described in section 2.1, we have computed the curvature values in each point of the mesh. Then we will compute for each e in M the mean of κ_1 and κ_2 at x and y. Considering then the scalar curvature functions explained above (section 2.1), we obtain then a map from the set of edges of the graph E into $\mathbb{R}$ that we denote by F, that will represent the curvature between each two adjacent triangles of the mesh. With such a map, we can compute a water-

shed cut that leads to a mesh segmentation. The cuts are thus performed on edges of the mesh, leading to a more accurate segmentation: the borders of the regions are constituted by sides of the meshes (in bold in fig. 2(c)).

2.2.3. Filtering

In order to classify artwork meshes, we are interested in partitionning a mesh into its most significant regions. Due to high number of minima the watershed cut produces a strong over segmentation. Hence, the need to filter the map F so that the watershed cut is guaranteed to produce exactly the needed regions. In mathematical morphology, a powerfull tool to solve this problem consists of using the component tree [9] to "remove" from F the "less significant" minima, according to some criterion such as the height, area and volume of the regions or a combination of them.

3. SEARCH AND RETRIEVAL

An object retrieval system must satisfy two main requirements: an effective data representation and an effective classification strategy. SVMs are state-of-the-art large margin classifiers which have demonstrated remarkable performances in object recognition. We use local features computed on regions for the object representation and we summarized them into bags on which we applied a SVM classification through specific kernels.

3.1. Local feature computation

In a previous work [10], we demonstrated the use of some different 3D descriptors computed on whole 3D models: Extended Gaussian Images (EGI) and Complex Extended Gaussian Images (CEGI), as well as 1D and 2D cord histograms. Thus we computed these features for each 3D surface region of the model instead of computing them on the global model. Finally, the set of descriptors available for each region is as follows: *(i)* EGI, *(ii)* CEGI, *(iii)* 1D and 2D cord histograms.

3.2. Surface region bags

After the region feature computation, each object i is represented by a bag $B_i = \{b_{ri}\}_r$ of region descriptors b_{ri}. b_{ri} represents one of the feature histograms. As the input space is constituted of bags (that is to say of sets of unordered vectors) of variable size, these bags must be mapped into a Hilbert space in order to use a linear classifier such as SVM. This can be achieved thanks to a kernel function. Several kernel functions have been recently proposed, for example in [11] and modified in [12] by:

$$K(B_i, B_j) = \left(\sum_{b_{ri} \in B_i} \sum_{b_{sj} \in B_j} \left(k\left(b_{ri}, b_{sj}\right) \right)^q \right)^{\frac{1}{q}}$$

where k is the minor kernel measuring the similarity between regions (we used a triangular minor kernel).

4. RESULTS

4.1. Segmentation results

We show on figure 3 some results of the 3D surface segmentation: (a) is computed on a CAD-design object and is the result of a watershed cut on a M curvature map; (b-d) are the results on three different figurines of the EROS-3D database, which mainly contains terracota figurines and vases, including fragments and moulds. The chess piece segmentation seems to be accurate, the frontiers clearly separate the different elementary shapes composing the model. The segmentation algorithm applied to real 3D models allows to identify some specific parts of the models (baby heads, legs, arms, coiffures) which are characteristic of the statues.


(a)   (b)   (c)   (d)
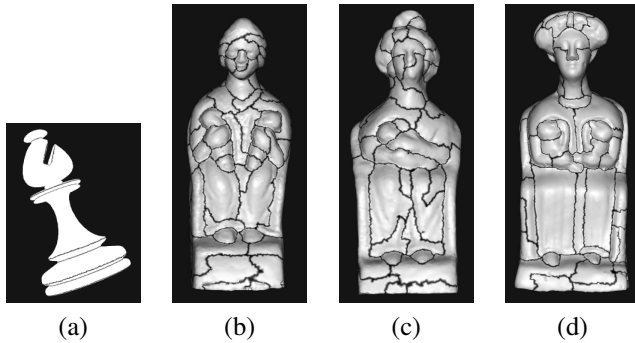
Fig. 3. Segmentation of (a) an artificial object (chess piece) and (b-d) several figurines of the EROS-3D database.

4.2. 3D model classification

We computed region features for the 700 models of the EROS-3D database. EGI where computed leading to 32 ("EGI:1") and 128 ("EGI:2") bin histograms, CEGI are 64 ("CEGI:1") and 256 ("CEGI:2") bin histograms; 1D cords are 384 bin histograms, and 2D cords 256 bin histograms. We performed the classification tests using the following protocol: for each model belonging to a category, we launched the search engine and computed the *Mean Average Precision* (MAP) value after 25 model annotations (the search engine shows the user some models to be annotated either as belonging to the same category than the query or not, and a new search is performed taking into account the user annotations); the MAP value for a given category is the mean value of the MAP of all queries belonging to this category. Figure 4 shows the MAP values for various feature vectors, either computed on the whole model (green bars) or on surface regions (yellow bars).
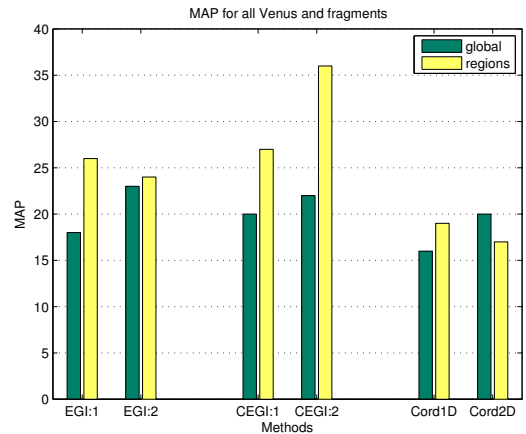
Figure 4 shows a real improvement of classification performances when using region attributes, especially EGI and
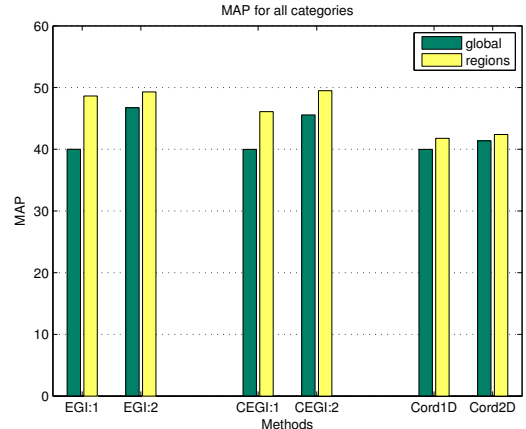

(a)
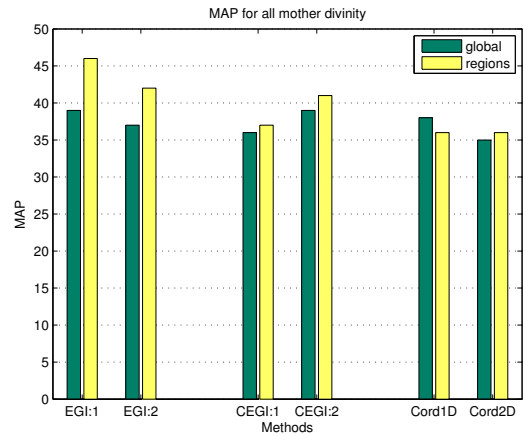
(b)

(c)

Fig. 4. Comparison of classification results for the EROS-3D database, using EGI, CEGI, or Cord feature vectors: (a) whole database; (b) mother divinity category; (c) Venus statue category, including fragments of statue and moulds.

CEGI. Moreover, EGI and CEGI features give better results than Cord features when computed on regions. The local computation of EGI and CEGI dramatically improves the recognition, especially for categories including full objects, fragments and moulds (*cf.* fig. 4(c) — "Venus statue" category contains statues, moulds and fragments such as heads, trunks, *etc.*) On the opposite, cord features, which are computed in reference to the center of the model (which differs when considering the whole statue or a fragment) do not lead to better performances. Figure 5 shows the results using only CEGI features for different specific categories: the performance gain clearly appears to be larger and significant when the searched category is specific (mother divinity with only one child) or when the category contains not only full models but also parts or fragments.
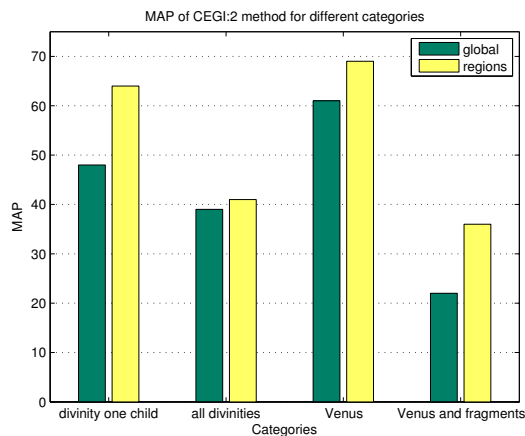
Fig. 5. Comparison of classification results on the EROS-3D database, using CEGI feature vectors for different categories.

5. CONCLUSION

In this paper, we have shown how the surface segmentation of 3D models and a region bag classification engine could help to perform a fine classification of 3D model databases, including real world complex objects as well as parts of them. We have developed a segmentation framework based on curvature computation and watershed cuts on curvature meshes, which give accurate results. The search engine, based on region bags and some well-known feature vectors, allow the classification of various categories in an artwork database: for instance, it let it possible to recognize "sub-categories" such as divinities with one child or two children, or divinities carrying their child on left or right arm.

6. REFERENCES

[1] R. Gal and D. Cohen-Or, "Salient geometric features for partial shape matching and similarity," *ACM Trans. on Graphics*, vol. 25, no. 1, pp. 130–150, 2006.

[2] Ph. Shilane and Th. Funkhouser, "Distinctive Regions of 3D Surfaces," *ACM Trans. on Graphics*, vol. 26, no. 2, 2007.

[3] Th. Funkhouser and Ph. Shilane, "Partial Matching of 3D Shapes with Priority-Driven Search," in *Symposium on Geometry Processing*, 2006.

[4] M.T. Suzuki, Y. Yaginuma, T. Yamada, and Y. Shimizu, "A 3D Model Retrieval Based on Combinations of Partial Shape Descriptors," in *IEEE North American Fuzzy Information Processing Society Annual Conference (NAFIPS 2006)*, 2006.

[5] J. Cousty, G. Bertrand, L. Najman, and M. Couprie, "Watershed cuts," in *8-th International Symposium on Mathematical Morphology*, 2007, pp. 301–312.

[6] J. Cousty, G. Bertrand, L. Najman, and M. Couprie, "Watersheds, minimum spanning forests, and the drop of water principle," 2007, Submitted, also in technical report IGM2007-01, http://igm.univ-mlv.fr/LabInfo/rapportsInternes/2007/01.pdf.

[7] S. Rusinkiewicz, "Estimating Curvatures and Their Derivatives on Triangle Meshes," in *Symp. on 3D Data Processing, Visualization, and Transmission*, 2004.

[8] J. Cousty, G. Bertrand, M. Couprie, and L. Najman, "Lignes de partage des eaux dans les espaces discrets," in *Morphologie Mathématique*, L. Najman and H. Talbot, Eds., Traité IC2. Hermès Science Publications, 2008, To appear.

[9] L. Najman and M. Couprie, "Building the component tree in quasi-linear time," *IEEE Trans. on Image Processing*, vol. 15, no. 11, pp. 3531–3539, nov. 2006.

[10] D. Gorisse, M. Cord, M. Jordan, S. Philipp-Foliguet, and F. Precioso, "3D Content-Based Retrieval in Artwork Databases," in *Proc. of the 3DTV-Conference*, 2007.

[11] S. Lyu, "Mercer Kernels for Object Recognition with Local Features," in *IEEE Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[12] P.-H. Gosselin, M. Cord, and S. Philipp-Foliguet, "Kernel on Bags of Fuzzy Regions for fast object retrieval," in *IEEE Int. Conf. on Image Processing (ICIP 07)*, 2007.