Artwork 3D model database indexing and classification

Sylvie Philipp-Foliguet, Michel M. Jordan, Laurent Najman, Jean Cousty

Artwork 3D Model Database Indexing and Classification

Sylvie Philipp-Foliguet[a,*], Michel Jordan[a], Laurent Najman[b], Jean Cousty[b]

[a] *ETIS, UMR 8051, CNRS, ENSEA, Université Cergy-Pontoise*
F-95000 CERGY-PONTOISE – France
[b] *Université Paris-Est, LABINFO–IGM — ESIEE*
F-93162 NOISY-LE-GRAND – France

Abstract

This paper presents a framework for the indexing and retrieval of artwork 3D models, allowing global and partial model classification and retrieval. The first part of the paper deals with database classification based on global shape descriptors. A search engine "RETIN-3D", using a SVM classifier coupled with an active learning strategy allows to retrieve categories of similar objects. In a second part, the classification is improved thanks to a local description of the models. A new framework for 3D surface segmentation is proposed. Shape descriptors are adapted to surface regions and kernels on descriptor bags are used to perform the database classification. Our system is designed for classifying and retrieving in ancient artwork 3D databases, and results from this application domain are presented and commented along the paper.

Keywords: 3D database classification, 3D shape descriptors, 3D surface segmentation, graph cuts, watersheds, artwork database, RETIN

1. Introduction

3D shape modeling and digitizing have received more and more attention for a decade, leading to an increasing amount of 3D model warehouses, either in domain-specific or wide-usage contexts. These 3D model databases require

*Corresponding author
Email addresses: sylvie.philipp-foliguet@ensea.fr (Sylvie Philipp-Foliguet), michel.jordan@u-cergy.fr (Michel Jordan), l.najman@esiee.fr (Laurent Najman), j.cousty@esiee.fr (Jean Cousty)

new tools for indexing, classifying, and retrieving the objects, in order to provide the final user an easy access to the models.

Content-based document retrieval (CBDR) has been a very active research field for a few years, and concerns textual documents, images, videos, and more recently 3D models. Usually CBDR is divided into two different steps: *(i)* an off-line step performs the document indexing by computing descriptors and features that are easily and fast compared, and thus builds an efficient summary of each document, called a signature; *(ii)* an on-line step, in which the user performs a search in the database thanks to a search engine. By means of signature comparison, the system ranks the database models according to their similarity to a query given as input. A feedback loop based on user interaction refines the results.

In this paper, we focus on 3D model indexing and retrieval and present our search engine "RETIN-3D". The first interactive 3D model search engines appear on the web around 2001–2002. The *Princeton 3D Model Search Engine*, associated to the widely used *Princeton Shape Benchmark (PSB)*, (http://shape.cs.princeton.edu/benchmark/) allows the user to perform text queries, 2D sketch queries, and to compare 3D models through some 3D shape descriptors [1]. The *3D Search Tool* from the University of Thessaloniki (http://3d-search.iti.gr/3DSearch) is based on the 3D generalized Radon transform and make comparisons within a 2.000 model database [2]; the results are only based on geometric comparisons, without learning, leading to some mis-classifications of the database. The European Network of Excellence *Aim@Shape* (http://www.aimatshape.net.) presents a *geometric search engine* which provides content-based retrieval with different matching methods (global or local, *etc.*). The SHREC 3D Shape Retrieval Contests allowed the comparison of 3D shape descriptors and 3D retrieval methods thanks to databases associated with ground-truthes [3]. Ohbuchi *et al.* [4] proposed a retrieval system based on multiresolution global features, which retrieves object categories from a single example.

We present here a 3D indexing and retrieval search engine dedicated to 3D artwork model databases. Our aim is to provide user friendly tools for classification, for content-based indexing, for retrieval, and for vizualization. These tools are firstly dedicated to historians and archeologists, who will be able to find, display and compare artworks in a few clicks. One can also imagine that museum visitors, provided with their PDA, could have the opportunity to interrogate a database in front of a statue and thus obtain a lot of additional information.

In section 2, we address the database classification by mean of global shape indexing. Unlike CAD models or artificial models that are often used in 3D model warehouses, artwork models are digitized in a high resolution (between 30,000 and 300,000 vertices) and do not exhibit regular surfaces. We compare several global shape descriptors for a classification task. We introduce the RETIN-3D search engine, which uses these shape descriptors (§ 2.1) to retrieve similar objects thanks to an active learning strategy (§ 2.2). Unlike search engines which are asked by textual requests or 2D/3D sketches, our query consists in a 3D model, and the search engine extracts from the database a category of models similar in a certain way to the query. The user leads the search toward the category, by annotating some objects as relevant or irrelevant for his search (§ 2.3).

Not surprisingly, global shape descriptors are not sufficient to discriminate objects differing by some specific details. In section 3, we propose to use local shape descriptors computed on regions of the surface. We introduce a new scheme for 3D surface segmentation, based on local curvature computation and watershed cuts (§ 3.1). Shape descriptors are computed for each region of the surface partition (§ 3.2), and the search engine is adapted to 3D region descriptor bags (§ 3.3). Partial matching results are shown and explained in § 3.4.

Major contributions of this paper are the 3D surface segmentation (§ 3.1) and the active learning for 3D model classification (§§ 2.2, 3.3), applied to a database of archeological objects. The database consists of high resolution models and includes broken or damaged items.

2. Global shape matching

At first, we compared some of the most used global shape features. We adapted our search engine RETIN [5], which is already used for image and video retrieval to this new modality.

2.1. Global shape features

The literature provides a lot of various 3D shape descriptors, describing geometric as well as topological properties of 3D shapes: global shape descriptors [6, 7, 8, 9, 10, 11, 12, 13, 14]; local descriptors [15]; graph based methods [16, 17, 18]; geometric methods based on 2D views of 3D models [19, 1, 20]. Interested readers can refer to some recent review papers [21, 22, 23, 24] for more details.

Before computing shape descriptors for each model of the database, we perform a spatial alignment preprocessing which aims to put the 3D models in a canonical coordinate system. The origin of the coordinate system is set at the center of gravity of the model, and the spatial alignment is achieved thanks to a principal component analysis transform.

Our 3D models are VRML meshes, featuring 3D vertices V (the model geometry) and 3D facets (triangles). We implemented most of the global shape descriptors [5] described above. We briefly describe below the ones which gave exploitable results for our high resolution models.

Cord histograms: [25] a cord is defined as the vector from a vertex to the center of the model, and is characterized by three features: *(i)* the length of the cord; *(ii)* its angle with the first principal axis; *(iii)* its angle with the second principal axis. We built a descriptor named "Cord2D" based upon two normalized histograms of the two first features (*cf.* fig. 1(a)).

Extended Gaussian Images (EGI): the 3D model is projected on a Gaussian sphere, and each point of the sphere is attributed with the total area of the faces having the same orientation [6]; for each facet of the Gaussian sphere, of orientation n_k:

$$P_{n_k} = \sum_{l=1}^{N_k} A_{l,n_k} \tag{1}$$

where N_k is the number of faces of the model in direction n_k and A_{l,n_k} is the area of the l-th face of orientation n_k (*cf.* fig. 1(b)). The main drawback of this index is that it is unable to differentiate convex and concave parts of the objects.

Complex Extended Gaussian Images (CEGI): in order to deal with this drawback, the CEGI feature [7] describes an object in the complex space through two attributes: the face orientation, and the distance between the center of the face and the center of the object. With the same notation as eq.(1):

$$P_{n_k} = \sum_{l=1}^{N_k} A_{l,n_k} e^{id_{l,k}} \tag{2}$$

where $d_{l,k}$ is the distance between the center of the model and the center of the face l (this distance is negatively signed if the face is directed towards the model center, and positively else). The CEGI descriptor is composed of the module and the phase of P_{n_k}.

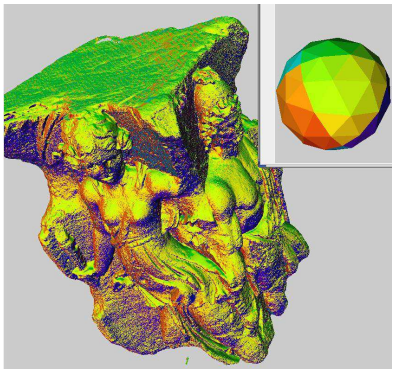
(a)  (b) 

Figure 1: Color representation of shape descriptors on a Venus figurine: (a) Cord2D descriptor; each color represents one cord length (b) EGI descriptor with the corresponding face orientations.

3D Hough Transform: the 3D Hough feature [8] is an extension of the Hough transform consisting in accumulating the spherical coordinates (s, θ, ϕ) which uniquely define the plane containing a model face. A 3D histogram is then computed, where each face contributes proportionally to its area.
Spherical harmonic representation: spherical harmonic transforms have been used to compute shape descriptors based on a voxel representation of 3D models [1]. More recently, spherical harmonic transform has been directly computed on 3D star-shape triangulated models [26, 27]. We use a descriptor which is based on the spherical harmonic transform computation using 32 concentric spheres centered in the center of gravity of each object.

2.2. Global shape indexing and retrieval

We present in this paper an extension of our search engine RETIN, which was originally built for image retrieval. We consider browsing a database or retrieving a category as a supervised classification problem. The user gives one or several examples of the objects he is looking for and the system returns the objects the most similar to these examples. In classification tasks, there are many ways to form the classes, depending on the user expectations. For example is he looking for greek vases, for vases with one handle, for vases with a particular shape or painting? Another problem is to initialize the search: if the database is large, how extract enough examples?

We have opted for an interactive mining of the database, using the following protocol, widely used in image retrieval: the search starts with a single example of the category, this example belongs or not to the database. The

system returns the objects the most similar to this example, according to a similarity measure. Then the system enters into a feedback loop where it displays some models to the user, the user annotates these models as relevant or irrelevant for his search and the system progressively builds two classes: the searched category, which contains all objects annotated as relevant and another class with all irrelevant objects. Of course in this scheme, only a small part of the database is annotated, the classification is thus "semi-supervised". To achieve such a classification task into two classes with very few examples, we used Support Vector Machines (SVM) [28], since this is a robust classification tool, well adapted to the context of noisy, complex and massive data. The SVM classifier is a linear two-class classifier, whose aim is to maximize the margin between both classes. It is generally used with kernel functions in order to deal with complex classes. We used Gaussian kernels in our experiments.

As the learning is performed with very few examples, RETIN works with an active learning scheme: the system displays the images (or the objects) the most likely to move the frontier between the two classes [28]. The extension of this system to 3D objects — RETIN-3D — includes a user-friendly interface (fig. 2). The top left part displays the models in 2D, ranked according to their membership grade to the searched category. The top right part of the screen displays an object in 3D, with the possibility to turn it and to zoom it. The "active learning panel" on the bottom of the screen displays the models that are the closest to the border between both classes: annotating these models helps to fast build the border, leading to better classification performances. The classification is performed online and according to the user expectations, thus giving a great flexibility to the system.

2.3. Global shape classification results

In order to test our system, we used a database of 3D objects – EROS-3D – provided by the French Museum Center for Research and Restoration (C2RMF, Le Louvre, Paris). This database now contains about 750 models of greek vases, terracota gallo-roman figurines, moulds, fragments, *etc.* (see fig. 3) and the database is still growing.

We focused on a particular application for historians concerning two model categories, terracota figurines representing either Mother-Divinity or Venus (see fig. 4(c) and 3(b)), with three different objectives:

1. extract the two categories from the database;

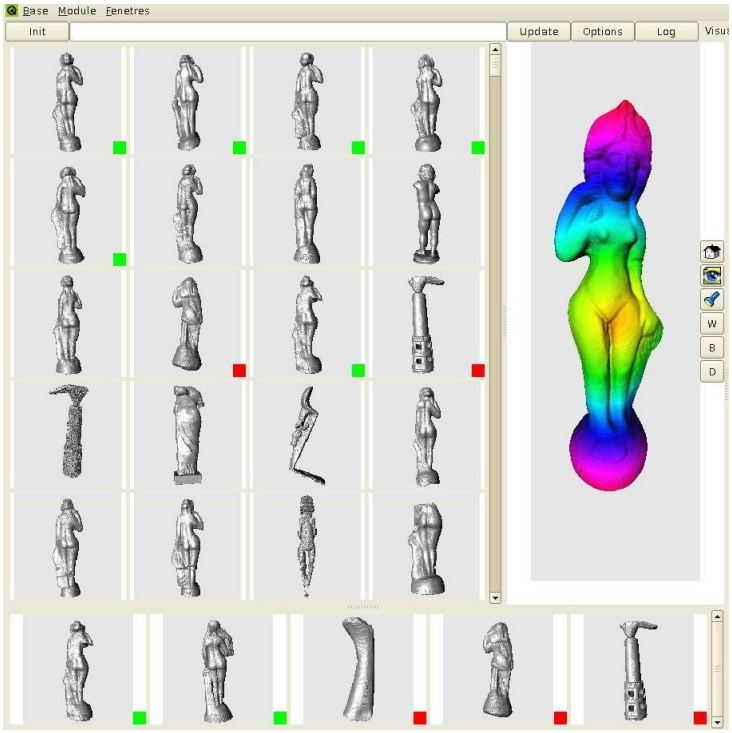


Figure 2: RETIN-3D user interface: left, the 3D models are ranked by their classification rate, top left is the model query; relevant (resp. irrelevant) objects are annotated with a green (resp. red) mark; at the bottom, the active learning panel. The zoom selected model (right panel) is colored according to the cord length values at each 3D surface point: yellow the low values, blue the high. The RETIN-3D interface allow to zoom and rotate the selected model.

Figure 3: Some objects of the EROS-3D database: figurines, moulds, vases.

2. separate Venus from Mother Divinity categories;
3. separate each category of Mother Divinities into three sub-categories: those carrying two children (fig. 4), those carrying one child in her right arm and those having their child in the left arm (fig. 9(f)).

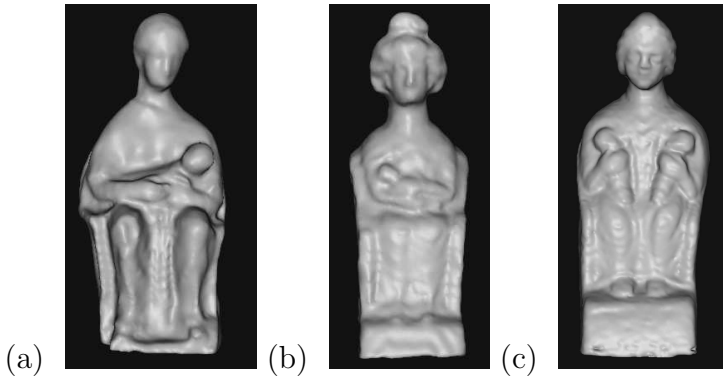
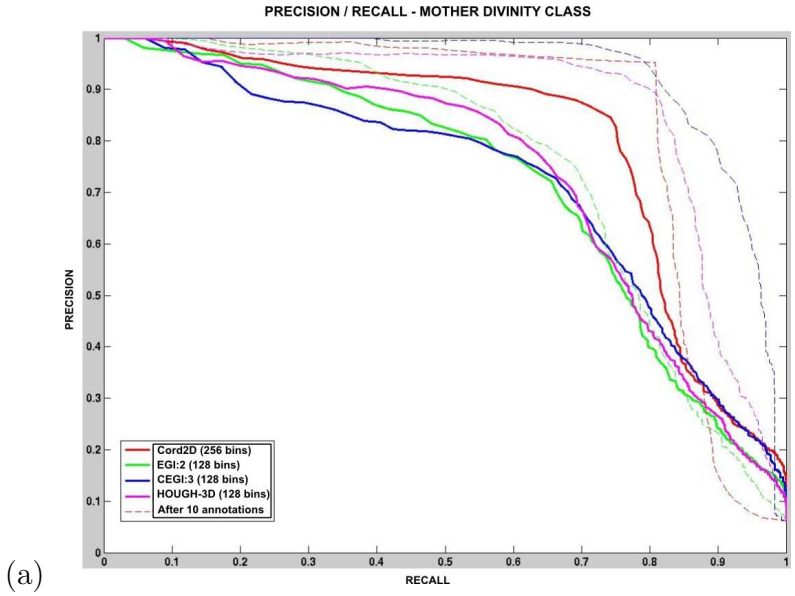


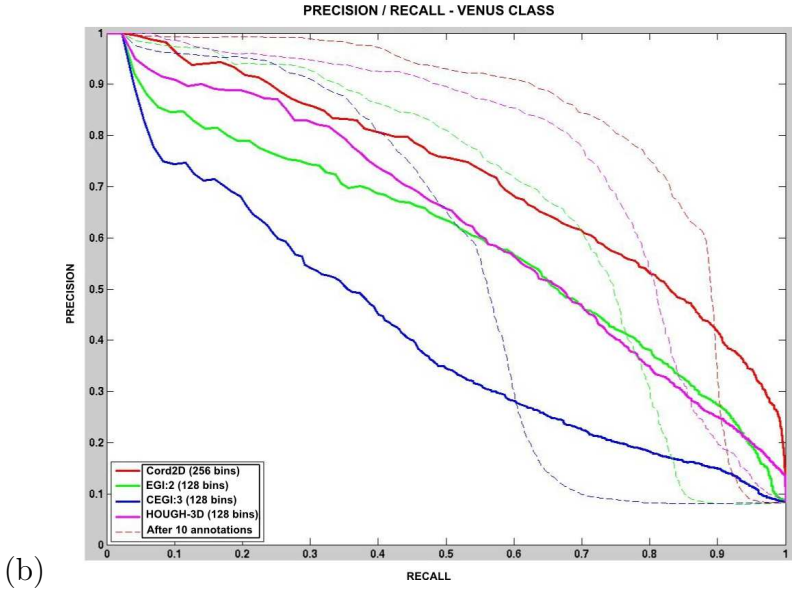
Figure 4: Mother-divinity figurines: (a) with a child on left arm; (b) with a child on right arm; (c) with two children.

In order to evaluate the performances of our algorithm, we built a ground-truth by manually extracting some categories of the EROS-3D database. We kept in the Venus and Mother-Divinity categories not only the entire models, but also damaged statues. For each feature, and for F returned models, precision P and recall R are computed as follows: $R = N_{OK}/T$ and $P = N_{OK}/F$ where N_{OK} is the number of correct models found among the F first returned models, and T is the size of the category. For each query, when F increases, one P/R curve is obtained. The process is repeated using each model of the category as query, and curves are averaged leading to one curve for each category and feature. We display P/R curves for Mother-Divinity (fig. 5(a)) and Venus (fig. 5(b)) categories. The solid curves present the results of the initial classification results with only one example of the searched category and the dotted curves with active learning, after ten annotations (either relevant or irrelevant).

The methods needs no parameter apart the number of bins (or classes) of the histograms. We give here the best results which were obtained for 256 bins for Cord2D (16 cord lengths times 16 angles), 128 orientations for EGI and for CEGI and 128 spherical coordinates for 3D Hough. Results with other parameters are displayed in fig. 10. For both categories, the best



(a)



(b)

Figure 5: Precision/recall diagrams for various shape descriptors: (a) mother-divinity category; (b) Venus category. Solid curves for initial classification (without annotations), and dotted curves after ten annotations.

performances were obtained with the Cord2D features, which seem to better represent the global shape of the model and particular details. The active learning process leads to a great improvement of performances: after 10 user annotations, the system returns about 80% of a category, with less than 10% of errors.

In order to have a single value, easier to achieve the comparisons, we computed the *Mean Average Precision* (MAP) value after 25 model annotations; the MAP value for a given category is the mean value of the integral of the P/R curve for all queries belonging to this category. Results are displayed in Table 1.

	Venus and fragments	**Mother divinities**	**All categories**
EGI:1	18	**39**	40
EGI:2	**23**	37	**47**
CEGI:1	20	36	40
CEGI:2	22	**39**	46
Cord1D	16	38	40
Cord2D	20	35	42

Table 1: Mean Average Precision (in %) for several categories and six descriptors.

The global MAP deeply depends on the categories. The retrieval with global descriptors is better for the Mother-Divinity category than for the Venus one. This can be explained by the fact that, in this database, the Venus category is sparser, including a significant number of damaged models. Moreover, if global descriptors are able to roughly classify the database, they are not sufficient for accurate purposes required by art historians. For example, they are not able to discriminate various categories of Mother-Divinities, carrying one or two children, with the child in right arm or in left arm (*cf.* fig. 4). For such an accurate discrimination, local features are necessary. This is the aim of the next section.

3. Partial shape matching

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, [29] presents the "saliency" of a region computed from its relative

size and some curvature measures (curvature value, its variance and changes), and uses 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 [30, 31]. Some other approaches are based on the decomposition of 3D models and well suits CAD or simple shape objects (see for instance [32]). For the purpose of 3D shape alignment, Gelfand *et al.* present the Integral Volume Descriptor, an invariant local shape descriptor robust to noise and based on the computation of the intersecting volume of a 3D model with a 3D ball centered on each vertex of the model [33]; these descriptors are then used to bring two different 3D shapes to coarse alignment.

We opted for a segmentation of the surface of the objects into convex (or concave) parts. Each surface region can then be described by the same features as the global shape, but restricted to the region. To compute the similarity between two models, we used a kernel between "bags" of regions.

3.1. 3D surface segmentation

Since the early work of Mangan and Whitaker [42], semantic-oriented approaches to mesh segmentation have gained a great interest in the research community (see, for instance, a comparative study in [43]). The 3D surface segmentation method adopted here falls into this category. It relies on the computation of local curvature combined with watershed cuts. The watershed cuts, introduced in [44] and [45], constitute a novel framework to solve the watershed problem. They verify strong mathematical properties which, as far as we know, are not satisfied by any other discrete watershed definition:

- **Consistency.** Watershed cuts are equivalently defined by their regions (catchment basins) through a steepest descent property or by the borders of the regions (the watershed divide) through the drop of water principle (*i.e.* a drop of water falling on the divide must have the possibility to flow down towards several minima of the function).

- **Optimality.** Watershed cuts are equivalent to graph cuts obtained by solving an instance of one of the most studied combinatorial optimization problem, namely the minimum spanning tree problem.

- **Efficiency.** Watershed cuts can be computed thanks to efficient linear time algorithms (with respect to the size of the input graph).

According to these properties, the method proposed in this section differs from the existing approaches based on similar ideas (such as [42, 47, 46]) in two main aspects. Firstly, the resulting segmentations are proved to be globally optimal. Secondly, these optimal segmentations are obtained in linear time. Note also that watershed cuts can be placed in a unifying framework that includes graph cuts [48] and random walks [49]. In this framework, watershed cuts can be written as an MRF-like style energy optimization algorithm [50] that allows for computing an exact global multi-labels optimum. As it is usual in that kind of framework, the quality of the segmentation depends on the detail of the energy definition. In this paper, we propose to design the energy from local curvature computation and to filter this energy by using the component tree to obtain a robust energy landscape, well adapted to obtaining the most significant regions for classification.

3.1.1. 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 cosine and sine of the angle between the first principal direction and this direction.

After computation thanks to the Trimesh algorithm [34] on a 3D mesh, we obtain the values κ_1 and κ_2 on each vertex of the mesh. These values increase with the convexity of the surface. They decrease into negative values on concave zones, getting low absolute values on flat zones. Considering the combination of the principal curvatures κ_1 and κ_2 on a surface we have *convex zones* when both values are great positive, *concave zones* when both are great negative and *saddle zones* when one value is great positive and the other great negative, meaning convexity in one direction and concavity in the other. The *flat zones* have both values low.

As the curvature map is used to partition the surface by using a watershed operator, a single scalar function is desired with high values at the boundaries between the regions that we want to separate. Several approaches can lead to such a height function by combining values κ_1 and κ_2. Mainly, the choice depends on the desired further applications.

The approach we adopted is to use a *pseudo-inverse* H_{inv} of the *mean curvature* $H = (\kappa_1 + \kappa_2)/2$, in the form: $H_{inv} = \frac{1}{\pi}\left(\arctan(-H) + \frac{\pi}{2}\right)$. This

function has the behavior of the inverse of the mean curvature but takes always positive values. It gives high values to concave zones and low values to convex zones. We also consider a *max curvature* as $M = \max(\kappa_1^2, \kappa_2^2)$, that gives high values on 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.

We have used this different treatments of the principal curvatures and, for the art objects we deal with, the H_{inv} function is the one with which we obtained the best results, while the max curvature M gives better results for manufactured objects.

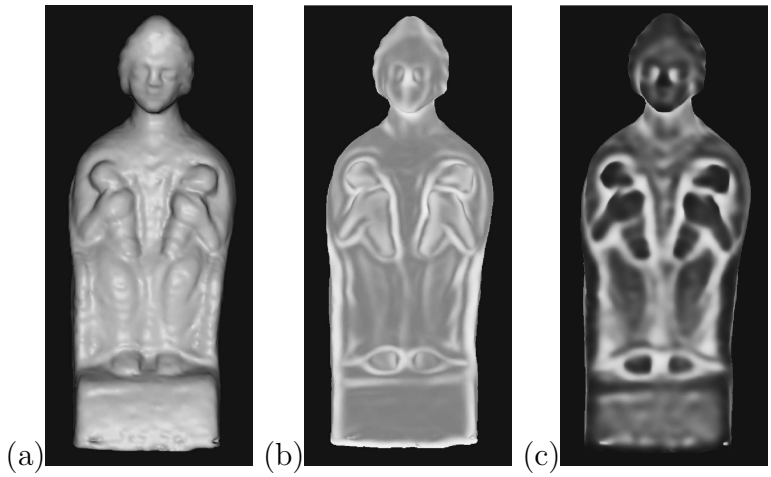


Figure 6: Curvature scalar functions: (a) 3D object; (b) max curvature; (c) pseudo-inverse curvature.

Fig. 6(b) and 6(c) illustrate these two scalar functions for the sculpture 6(a). Small values are black, while large values are white.

3.1.2. Watershed cuts on curvature meshes

Many approaches [35, 36] have been proposed to define and/or compute the watershed of a vertex-weighted graph corresponding to a grayscale image. The digital image is seen as a topographic surface: the gray level of a pixel becomes the elevation of a point, the basins and valleys of the topographic surface correspond to dark areas, whereas the mountains and crest lines correspond to light areas.

In recent papers [44], we investigate 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 the definition of watershed cuts in edge-weighted graphs [44]. We establish the consistency (with respect to characterizations of the catchment basins and dividing lines) 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 framework 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.

Watershed cuts can be extended [37] to *simplicial complexes*, and especially meshes. Consider a 3D surface mesh M (composed of *triangles*, *sides of the 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 $G = (V, E)$ with one vertex for each face of M and an edge connecting two vertices if the corresponding two faces share a side (fig. 7.b).

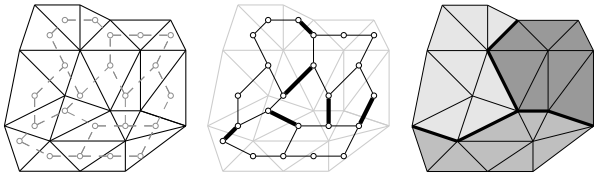


Figure 7: (a) A triangle mesh. (b) Segmentation on edges of the graph (in bold). (c) Segmentation on the mesh.

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 3.1.1, we have computed the curvature values in each point of the mesh. We denote them as κ_{1x}, κ_{2x} and κ_{1y}, κ_{2y} for the points x and y respectively. Then we compute for each e in M, the mean of κ_1 and κ_2 at x and y: $\kappa_1 = (\kappa_{1x} + \kappa_{1y})/2$ and $\kappa_2 = (\kappa_{2x} + \kappa_{2y})/2$. Considering then the scalar curvature functions explained in section 3.1.1, we then obtain a map from E into $\mathbb{R}$ that we denote by f, and that represents the curvature between each two adjacent faces of the mesh. With such a map, we can

compute a watershed cut that leads to a mesh segmentation. The cuts are thus performed on edges of the mesh, leading to a natural and accurate segmentation: the borders of the regions are constituted by sides of triangles of the mesh (in bold in fig. 7.c).

3.1.3. Filtering

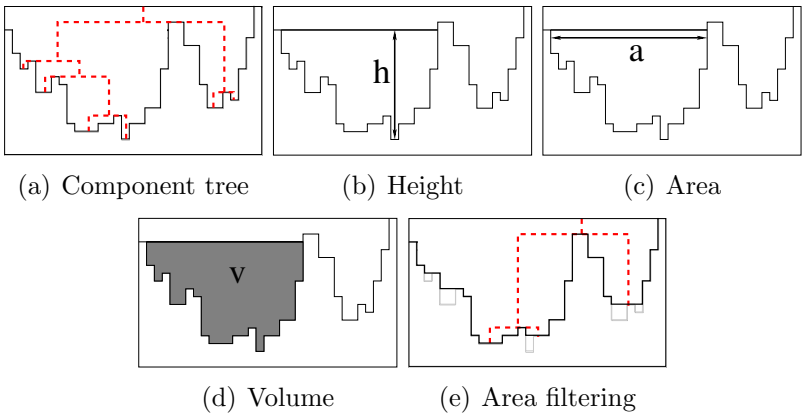
In order to classify artwork meshes, we are interested in partitioning a mesh into its most significant regions. Due to high number of minima, watershed cut produces strong over segmentation since it associates a region to each minimum. Hence, we need to filter the map f so that the watershed cut is guaranteed to produce only the needed regions.

In mathematical morphology, a powerful tool to solve this problem consists of using the component tree of a map (see an illustration in fig. 8a). The level sets of an edge-weighted graph are the sets of edges whose weights are below a given threshold. The connected components of the level sets, thanks to the inclusion relation, can be organized in a tree structure, that is called the *component tree* [38], each connected component of any level set being a node of the tree. This tree, under several variations, has been used in numerous applications. Various algorithms have been proposed in the literature for computing the component tree. The fastest one [39], considering the worst-case complexity, is based on Tarjan's union-find procedure and is quasi-linear. This algorithm is also the fastest for real (*double*) data such as those given by a curvature map [52].

Using the component tree for image filtering consists in removing from f the "less significant" lobes, *i.e.* the "negligible" components of the component tree. To make such an idea practicable, it is necessary to quantify the relative importance of each node of the component tree. We can do that by computing some attributes for each node. Among the numerous attributes that can be computed, three are natural: the height, the area, and the volume (fig. 8b, c and d). The function depicted in black in fig. 8d is the result of an area filtering (of size 3) applied to the function of fig. 8a.

3.2. Local features

In § 2.1, 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 Cord1D and Cord2D features. We also computed for each region a new curvature region histogram, which is a 2D histogram based on each principal curvature $\{\kappa_1, \kappa_2\}$ pair. Thus we



(a) Component tree (b) Height (c) Area

(d) Volume (e) Area filtering

Figure 8: Illustration of respectively a component tree (dashed lines), the height, the area and the volume of a component and an area filtering.

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)* Cord1D and Cord2D, *(iv)* 2D curvature histograms. EGI and Cord2D features, which are computed in reference to the center of the model, allow to take into account not only the 3D shape of a region but also its position relatively to the model.

3.3. Surface region bags

We again use our search engine RETIN to perform the classification of the objects after segmentation into regions. We use local features computed on regions for the object representation. An object is thus summarized by a set or "bag" of vectors describing the regions. The SVM classifier can then be applied through specific kernels.

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 [40] and modified in [41] 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 (usually a Gaussian kernel).

3.4. Partial shape matching results

3.4.1. Segmentation results

We display on figure 9 some results of the 3D surface segmentation: fig. 9(a) is computed on a CAD-design object and is the result of a watershed cut on a M curvature map; fig. 9(b)–9(d) show the segmentation results for two industrial CAD models and one model from the Princeton Shape Benchmark (each model contains between 100 and 1,500 vertices); fig. 9(e)–9(g) are the results on three different figurines of the EROS-3D database. For CAD models such as industrial pieces or chess bishop, the surface 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 figurines. Of course, as watershed regions have to be closed in any cases, some frontiers must go through flat areas. These frontiers have no semantic significance. The volume criterion was used to filter the component tree (minimum volume is set to 0.5% of each model volume) in the case of archeological objects. The artificial objects were filtered by mean of the area criterion.

3.4.2. 3D model classification

We computed region features for the whole EROS-3D database. EGI where computed with 32 (EGI:1) and 128 (EGI:2) bins, CEGI with 64 (CEGI:1) and 256 (CEGI:2) bins, Cord1D with 384 bins, and Cord2D with 256 bins. We performed the classification tests using the same protocol as in section 2. Figure 10 shows the MAP values for various feature vectors, either computed on the whole model (green bars) or on surface regions (yellow bars).

Figure 10 shows a real improvement of classification performances when using region attributes, especially EGI and 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 if the category includes full objects, as well as fragments and moulds (*cf.* fig. 10(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

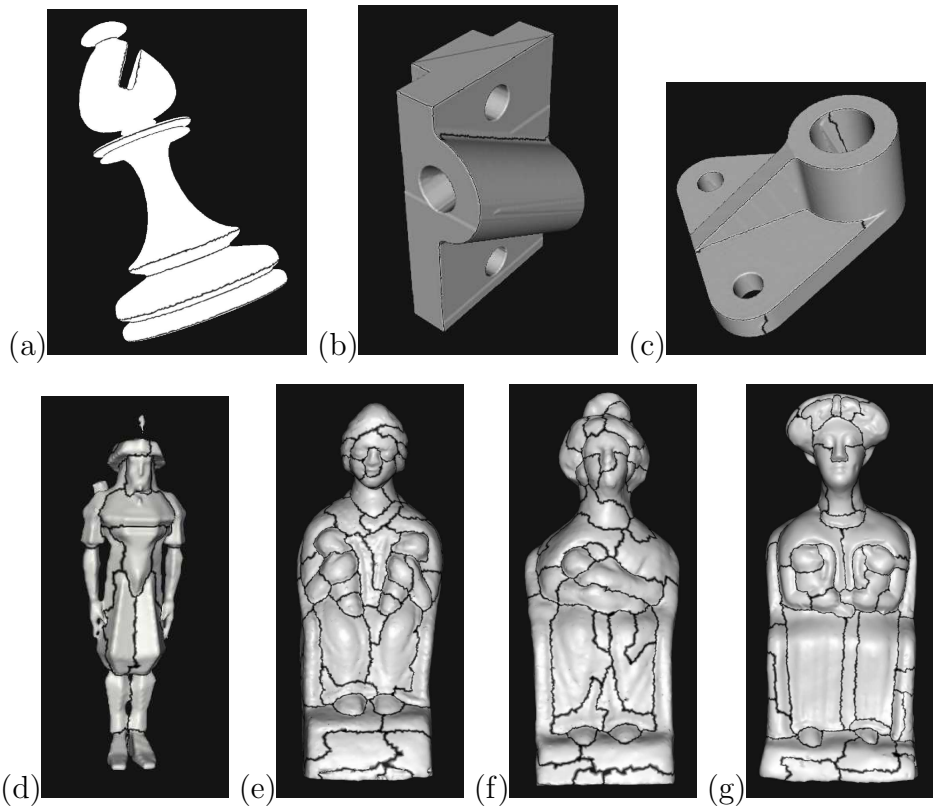
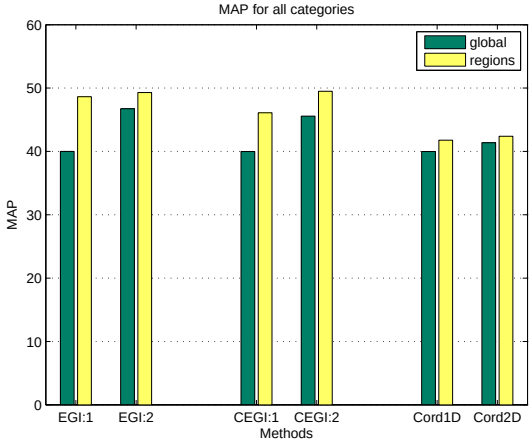
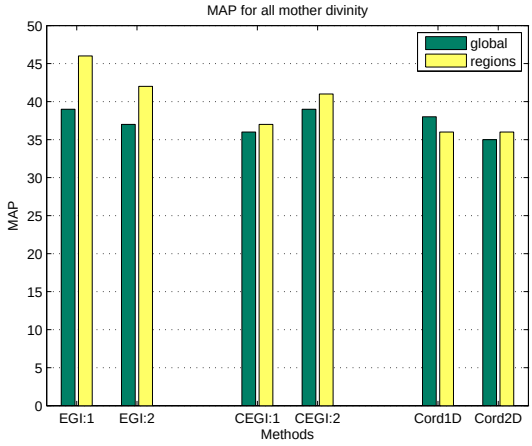


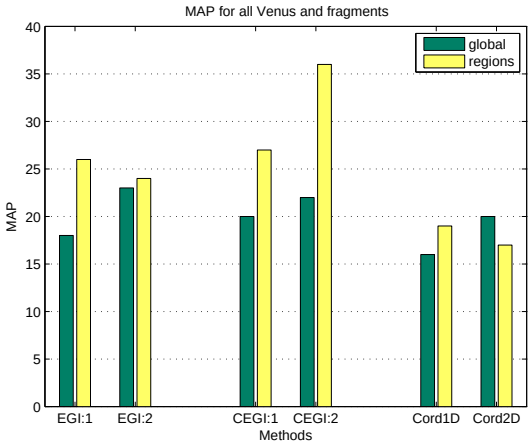
Figure 9: Segmentation of (a) an artificial object (chess piece); (b)–(c) two industrial pieces; (d) a model of the Princeton shape benchmark; (e)–(g) three Mother-Divinities of the EROS-3D database.



(a)



(b)



(c)

Figure 10: 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.

when considering the whole statue or a fragment) do not lead to better performances.

Figure 11 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.

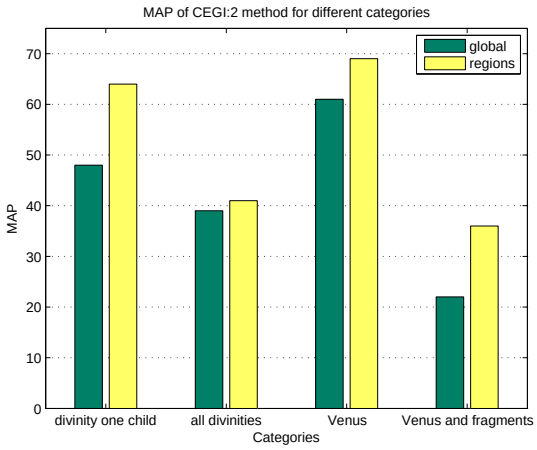


Figure 11: Comparison of classification results on the EROS-3D database, using CEGI feature vectors for different categories.

4. Conclusion

We presented in this paper a 3D model search engine dedicated to artwork 3D models. In this framework of high resolution models, we found that among the most used shape descriptors, a 2D cord histogram (length and angle) is the most efficient to build rough categories of objects. If a more accurate discrimination of the objects, concerning only a part of the objects is needed, we showed that a good solution is to segment the 3D surface.

We proposed a watershed cut algorithm applied on the dual graph of the 3D mesh, leading to a 3D surface segmentation along the *max* curvature lines. Shape descriptors are then computed for each surface region of the segmentation.

The combination of an accurate surface segmentation and local shape descriptors significantly improves the database classification. EGI and CEGI

descriptors better perform the discrimination of objects only differing by some details. The final user (a museum curator, for instance) may be particularly interested in partial shape matching, for example to classify fragments or to classify objects according to minor details.

The active learning strategy, developed in our search engine RETIN-3D is thus an important tool for the final user.

Acknowledgments:
This project was conducted under the French research agency (ANR) contract ANR-05-MMSA-0001-05. We particularly thank Pr. C. Lahanier, G. Aitken, and the C2RMF (Le Louvre, Paris) for the 3D model collection EROS-3D used in this paper. We also thank D. Gorisse, M. Alcoverro and J. Gony who contributed to parts of this paper.

[1] T. Funkhouser, P. Min, M. Kazhdan, J. Chen, A. Halderman, D. Dobkin, and D. Jacobs. A Search Engine for 3D Models. *ACM Trans. on Graphics*, 22(1):83–105, Jan. 2003.

[2] P. Daras, D. Zarpalas, D. Tzovaras, and M.G. Strintzis. Efficient 3-D model search and retrieval using generalized 3-D radon transforms. *IEEE Trans. on Multimedia*, 8(1):101–114, Feb. 2006.

[3] R.C. Veltkamp and F.B. ter Haar. SHREC2007: 3D Shape Retrieval Contest. Technical report, Utrecht University, The Netherlands, 2007. http://www.cs.uu.nl/groups/MG/multimedia/publications/.

[4] R. Ohbuchi, A.Yamamoto, J.Kobayashi. Learning semantic categories for 3D Model Retrieval.. In *Proc. of the ACM Int. Multimedia Conference, Proc. of the int. workshop on Multimedia Information Retrieval (MIR) 2007*, Augsburg, Bavaria, Germany. pp.31-40, ACM Press, USA.

[5] 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*, Kos Island, Greece, 7-9 May 2007.

[6] B.K.P Horn. Extended Gaussian Images. *Proc. of the IEEE*, 72(12):1671–1686, dec. 1984.

[7] S.B. Kang and K. Ikeuchi. The complex EGI: a new representation for 3D pose determination. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 16(3):249–258, March 1994.

[8] T. Zaharia and F. Prêteux. Shape-based retrieval of 3D mesh models. In *IEEE Int. Conf. on Multimedia and Expo (ICME'2002)*, Lausanne, Switzerland, 2002.

[9] M. Kazhdan, B. Chazelle, D. Dobkin, T. Funkhouser, and S. Rusinkiewicz. A Reflective Symmetry Descriptor for 3D Models. *Algorithmica*, 38(1):201–225, 2003.

[10] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin. Shape Distributions. *ACM Trans. on Graphics*, 21(4):807–832, oct. 2002.

[11] E. Wahl, G. Hillenbrand and G. Hirzinger. Surflet-pair-relation histograms: a statistical 3D-shape representation for rapid classification. In *Proc. of 3-D Digital Imaging and Modeling 3DIM 2003.*, pp.474–481, oct. 2003.

[12] R. Ohbuchi, T.Minamitani, and T .Takei. Shape-similarity search of 3D models by using enhanced shape functions. In *Int. J. of Computer Applications inTechnology (IJCAT)*, 23(3/4/5):70-85, 2005.

[13] P. Papadakis, I. Pratikakis, S. Perantonis, and T. Theoharis. Efficient 3D Shape Matching and Retrieval using a Concrete Radialized Spherical Projection Representation. *Pattern Recognition Journal*, 40(9):2437–2452, Sept. 2007.

[14] M. Ben-Chen and C. Gostman. Characterizing Shape Using Conformal Factors. In *Eurographics Workshop on 3D Object Retrieval*, Crete, Greece., April 2008.

[15] T. Zaharia and F. Prêteux. 3D shape-based retrieval within the MPEG-7 framework. In *SPIE Conference 4304 on Nonlinear Image Processing and Pattern Analysis*, volume XII, pp.133–145, San Jose, 2001.

[16] T. Tung and F. Schmitt. The augmented multiresolution Reeb graph approach for content-based retrieval of 3D shapes. *International Journal of Shape Modeling (IJSM)*, 11(1):91–120, June 2005.

[17] S. Biasotti, D. Giorgi, M. Spagnuolo, and B. Falcidieno. Reeb graphs for shape analysis and applications. *Theoretical Computer Science*, 392(1-3):5–22, 2008.

[18] N. Iyer, Y. Kalyanaraman, K. Lou, S. Jayanti, and K. Ramani. A Reconfigurable, Intelligent 3D Engineering Shape Search System Part I: Shape Representation. In *ASME DETC'03, 23rd Computers and Information in engineering (CIE) Conf.*, Chicago, Illinois, 2003.

[19] T.F. Ansary, J.-P. Vandeborre, S. Mahmoudi, and M. Daoudi. A Bayesian Framework for 3D Models Retrieval Based on Characteristic Views. In *Int. Symp. on 3D Data processing, Visualization and Transmission, 3DPVT'04*, Thessaloniki, Greece, Sep. 6-9 2004. IEEE.

[20] D.V. Vranic. *3D Model Retrieval*. PhD thesis, University of Leipzig, 2004.

[21] J.W.H. Tangelder and R.C. Veltkamp. A Survey of Content Based 3D Shape Retrieval Methods. In *Proc. of Shape Modeling Int., SMI'04*, pp.145–156, Genova, Italy, 2004.

[22] N. Iyer, S. Jayanti, K. Lou, Y. Kalyanaraman, and K. Ramani. Three-dimensional shape searching: state-of-the-art review and future trends. *Computer-Aided Design*, 37(5):509–530, April 2005.

[23] B. Bustos, D.A. Keim, D. Saupe, T. Schreck, and D.V. Vranic. Feature-Based Similarity Search in 3D Object Databases. *ACM Computing Surveys*, 37(4):345–387, 2005.

[24] J.W.H. Tangelder and R.C. Veltkamp. A survey of content based 3D shape retrieval methods. *Multimedia Tools and Applications*, 39(3):441–471, 2008.

[25] E. Paquet, A. Murching, T. Naveen, A. Tabatabai, and M Rioux. Description of Shape Information for 2-D 3-D Objects. *Image Communication Journal*, 16:103–12, 2000.

[26] M. Mousa, R. Chaine, and S. Akkouche. Frequency-Based Representation of 3D Point-Based Surfaces Using the Spherical Harmonics. In *ICCVG'06, Int. Conf. on Computer Vision and Graphics*, sep 2006.

[27] M. Mousa, R. Chaine, S. Akkouche, and E. Galin. Efficient spherical harmonics representation of 3D objects. In *15 th Pacific Graphics*, pp.248–257, oct 2007.

[28] M. Cord, P.-H. Gosselin, and S. Philipp-Foliguet. Stochastic exploration and active learning for image retrieval. *Image and Vision Computing*, 25:14–23, 2007.

[29] R. Gal and D. Cohen-Or. Salient geometric features for partial shape matching and similarity. *ACM Trans. on Graphics*, 25(1):130–150, 2006.

[30] P. Shilane and T. Funkhouser. Distinctive Regions of 3D Surfaces. *ACM Trans. on Graphics*, 26(2), 2007.

[31] T. Funkhouser and Ph. Shilane. Partial Matching of 3D Shapes with Priority-Driven Search. In *Symp. on Geometry Processing*, 2006.

[32] 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.

[33] N. Gelfand, N. J. Mitra, L. J. Guibas, and H. Pottmann. Robust Global Registration. In *Eurographics Symp. on Geometry Processing*, pp.197–206, 2005.

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

[35] L. Vincent and P. Soille. Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(6):583–598, June 1991.

[36] G. Bertrand. On topological watersheds. *J. of Mathematical Imaging and Vision*, 22(2-3):217–230, 2005.

[37] G. Bertrand, M. Couprie, J. Cousty, and L. Najman. Lignes de partage des eaux dans les espaces discrets. In L. Najman and H. Talbot, editors, *Morphologie Mathématique 1 : approches déterministes*, Traité IC2, chapter 4, pp.123–150. Hermès Science Publications, 2008.

[38] P. Salembier, A. Oliveras and L. Garrido. Anti-extensive Connected Operators for Image and Sequence Processing. *IEEE Trans. on Image Processing*, 7(4):555–570, April 1998.

[39] L. Najman and M. Couprie. Building the component tree in quasi-linear time. *IEEE Trans. on Image Processing*, 15(11):3531–3539, nov. 2006.

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

[41] 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.

[42] A.P. Mangan and R.T. Whitaker. Partitioning 3d surface meshes using watershed segmentation. *IEEE Trans. on Visualization and Computer Graphics*, 5(4):308–321, Oct-Dec 1999.

[43] M. Attene, S. Katz, M. Mortara, G. Patane, M. Spagnuolo, and A. Tal. Mesh segmentation — a comparative study. In . *IEEE Int. Conf. on Shape Modeling and Applications, 2006. SMI 2006*, pp.14–25, June 2006.

[44] J. Cousty, G. Bertrand, L. Najman, and M. Couprie. Watershed cuts: Minimum spanning forests and the drop of water principle. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 31(8):1362–1374, Aug. 2009.

[45] J. Cousty, G. Bertrand, L. Najman, and M. Couprie. Watershed cuts: Thinnings, shortest-path forests and topological watersheds. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 32(5):925–939, 2010.

[46] A. Jagannathan and E. L. Miller. Three-dimensional surface mesh segmentation using curvedness-based region growing approach. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(12):2195–2204, 2007.

[47] D. L. Page, A. F. Koschan, and M. A. Abidi. Perception-based 3d triangle mesh segmentation using fast marching watersheds. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2:27, 2003.

[48] Y. Boykov and M.-P. Jolly. *Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images*. In *ICCV*, pp.105–112, 2001.

[49] L. Grady. Random walks for image segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 28(11):1768–1783, 2006.

[50] C. Couprie, L. Grady, L. Najman, and H. Talbot. Power watersheds: a new image segmentation framework extending graph cuts, random walker and optimal spanning forest. In *12th International Conference on Computer Vision (ICCV'09)*, September 2009.

[51] J. Cousty, G. Bertrand, M. Couprie, and L. Najman. Collapses and watersheds in pseudomanifolds. In *Combinatorial Image Analysis - IWCIA 2009*, Lecture Notes in Computer Science, pp.397–410. Springer, Nov. 2009. Wiederhold, Petra and Barneva, Renata P., eds.

[52] C. Berger, T. Géraud, R. Levillain, N. Widynski, A. Baillard and E. Bertin. Effective Component Tree Computation with Application to Pattern Recognition in Astronomical Imaging. In *Proc. of the IEEE Int. Conf. on Image Processing*. San Antonio, Texas, USA, September 16-19, 2007.