

Indexing of 3D Models Based on Graph of Surfacic Regions

Sylvie Philipp-Foliguet
ETIS, CNRS ENSEA
Université Cergy-Pontoise
F-95000 Cergy-Pontoise
sylvie.philipp@ensea.fr

Matthias Fuzier
ETIS, CNRS ENSEA
Université Cergy-Pontoise
F-95000 Cergy-Pontoise
matthias.fuzier@ensea.fr

Michel Jordan
ETIS, CNRS ENSEA
Université Cergy-Pontoise
F-95000 Cergy-Pontoise
michel.jordan@u-cergy.fr

Philippe-Henri Gosselin
ETIS, CNRS ENSEA
Université Cergy-Pontoise
F-95000 Cergy-Pontoise
gosselin@ensea.fr

ABSTRACT

We present a search engine dedicated to 3D object databases. The originality of the method is to represent models by adjacency graphs of surfacic regions. After segmentation of the 3D surface of a model, regions are described by various shape descriptors. The similarity between graphs is computed by kernels on graphs computed from kernels on walks. These kernel functions take into account both the similarity between regions and their spatial relationship. The search engine performs interactive research in a database from a query object, by using semi-supervised classification. The system is applied to a database of 3D high resolution artwork models. We show that a graph representation outperforms a global description of the objects, when using the same descriptors.

Categories and Subject Descriptors

H.3 [Information Systems]: Information Storage and Retrieval; H.3.1 [Information Storage and Retrieval]: Content Analysis and Indexing

General Terms

Algorithms

Keywords

3D shape indexing; kernel on graphs; 3D region graphs; artwork database;

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 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 has been a very active research field for a few years, and concerns textual documents, images, videos, and more recently 3D models. Content-based document retrieval systems are usually composed of two steps: (i) an off-line step which performs the document indexing by computing descriptors and features gathered in a signature; (ii) an on-line step, in which the user performs a search in the database with a search engine working on the signatures. By means of signature comparison, the system ranks the database documents 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 interactive 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)*, allows the user to perform text queries, 2D sketch queries, and to compare 3D models through some 3D shape descriptors [7]. The *3D Search Tool* from the University of Thessaloniki is based on the 3D generalized Radon transform and make comparisons within a 2,000 model database [5]; the results are only based on geometric comparisons, without learning, leading to some mis-classifications of the database. Another promising scheme has been proposed recently in [12], it is based on Fourier series computed on closed curves representing the 3D model and seems robust to noise. The European Network of Excellence *Aim@Shape* 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-truths [22]. Ohbuchi *et al.* [15] proposed a retrieval system based on multiresolution global features, which retrieves object categories from a single example. Interested readers will find detailed surveys of 3D indexing and retrieval methods in [2, 21].

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

3DOR'10, October 25, 2010, Firenze, Italy.

Copyright 2010 ACM 978-1-4503-0160-2/10/10 ...\$10.00.

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 content-based indexing, retrieval and visualization. These tools are firstly dedicated to historians and archeologists, who will be able to find, display and compare artworks in a few clicks, but also to museum visitors, who could have the opportunity to ask a database in front of a statue and thus obtain a lot of additional information.

We propose to use local shape descriptors computed on regions of the surface. We first segment the surface into its main (and if possible semantic) parts, using the local curvature and watershed cuts. Shape descriptors are then computed for each region of the surface partition. The problem of object classification is now reduced to a problem of inexact graph matching. We propose to use kernels on graphs computed from kernels on walks to compute the similarity between these graphs.

We extended our search engine RETIN [17], already used for image and video retrieval, to a new search engine dedicated to 3D objects, called RETIN-3D. Our query consists in a 3D model, and the search engine retrieves 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.

In this paper 3D models are represented by adjacency graphs of surfacic regions. The major contribution is the use of graph kernels to perform object classification and retrieval. We show that graph representation outperforms global description of the objects. The method is applied to a database of archeological objects which were digitized with a high resolution (between 30,000 and 300,000 vertices per model); the database is called EROS-3D and provided by the French Museum Center for Research and Restoration (C2RMF, Le Louvre, Paris). This databas contains about 750 models of greek vases, terracota gallo-roman figurines, moulds, fragments, *etc.* We focus on a particular application for historians concerning two model categories, representing Mother-Divinity and Venus.

In section 2, the models are represented by a valued graph of regions obtained by segmentation of the surface. In section 3, we address the database classification by mean of valued graph matching. In section 4, experiments compare global description and local descriptions associated with spatial relationships through adjacency graph of regions.

2. OBJECT REPRESENTATION AS ADJACENCY GRAPH OF SURFACIC REGIONS

The 3D surface segmentation algorithm was published in [1]. We only remind here the main steps. A detailed version will also appear in [18]. The algorithm is based on the local curvature estimation, followed by a watershed process providing a partition of the 3D surface.

2.1 3D surface segmentation

If κ_1 and κ_2 are the principal (minimum and maximum) curvatures at each point of a surface, the *mean curvature* is defined by $H = (\kappa_1 + \kappa_2)/2$ and the inverse curvature by $H_{inv} = \arctan(-H + \pi/2)$. H_{inv} takes only positive values, high values in concave zones and low values in convex zones.

In fig. 1 the values of these scalar functions are displayed

in gray scale for an object of our database (the whiter the grey level, the higher the curvature value).

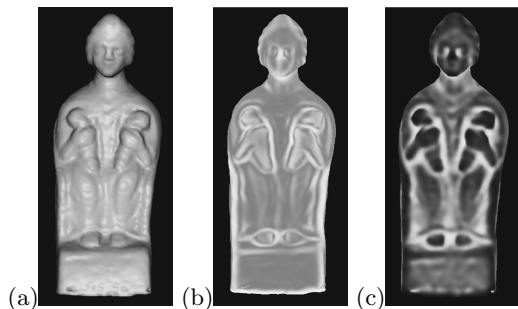


Figure 1: Curvature scalar functions: (a) 3D object; (b) max curvature; (c) inverse curvature H_{inv} .

Watersheds were recently investigated [4], in the framework of graphs whose edges are weighted by a cost function. 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) which share edge e . 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.

Let (x, y) be the pair of points such that $e = \{x, y\}$, we assign the mean of principal curvatures (κ_1 and κ_2) of x and y to e . Considering then the scalar curvature functions explained above (H or H_{inv}), we obtain a map from the set of edges of the graph into \mathbb{R} that we denote by F , and that will represent the curvature between each two adjacent triangles 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 more accurate segmentation: the borders of the regions are constituted by sides of the meshes.

Unlike CAD models or artificial models that are often used in 3D model warehouses, our artwork models do not exhibit regular surfaces, due to their high resolution digitization. In order to classify artwork models, we are interested in partitioning a mesh into its most significant regions. Due to the high number of curvature minima, the watershed cut produces a strong over segmentation. To solve this problem we used the component tree [13] to remove the less significant minima from F , according to criteria based on height, area or volume of the regions.

We show on figure 2 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 *max curvature* map ($\max(\kappa_1^2, \kappa_2^2)$); (b-d) are the results on three different figurines of the database. 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.

2.2 Local feature computation

In a previous work [18], 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.

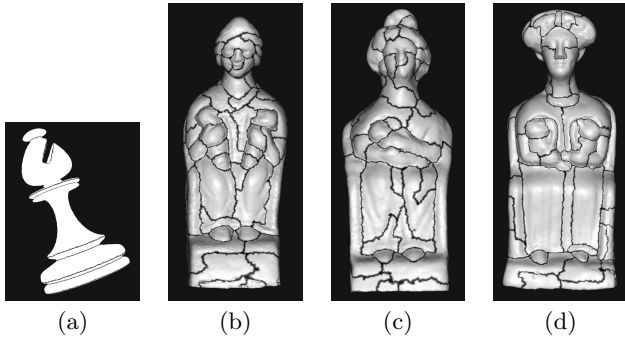


Figure 2: Segmentation of (a) an artificial object (chess piece) and (b-d) several figurines of the database.

EGI [8] and CEGI [9] are projections of the model onto the Gaussian sphere, where each point of the sphere is valued with the total area of the object faces having the same orientation. CEGI take in account the face orientation, in order to discriminate convex and concave surfaces. Cord histograms are defined as distributions of lengths of the cord (the line between the center of the model and a point on the model surface) and angles of the cord with respect to the first and second axis of the model [16].

We also computed some other descriptors, such as AD/AAD [14] and SPRH [24]: AD is a 2D histogram of the angles and distances computed on randomly chosen pairs of surface points. The AAD feature moreover takes into account unoriented or inconsistently oriented surfaces and thus is more robust. SPRH computes a 4-dimensional geometric feature from pairs of 3D surface points, aiming to capture local and global shape aspects. All these features can be used to characterize the global shape of the model through their statistical distribution, stored in histograms.

To catch the local shape of the models, we reduce the feature computation to each region of the surface instead of a computation on the whole model. AD/AAD and SPRH features, due to their conception, do not fit to a local computation. Finally, the set of descriptors available for each region is as follows: (i) EGI, (ii) CEGI, (iii) Cord1D and Cord2D histograms.

3. GRAPH MATCHING

Once objects are represented by valued relational graphs, the problem of comparing objects can be considered as a problem of inexact graph matching [6, 20]. The problem is twofold: firstly find a similarity measure between graphs of different sizes and secondly find the best match between graphs in an “acceptable” time in order to perform on-line classification. Kernel functions can be seen as similarity functions, which can be used with many classifiers and especially SVM [19].

To capture the structure of a graph, kernels on graph can be defined from kernels of walks [10, 23].

3.1 Kernel on graphs based on kernels on walks

A walk h in a graph $G = (V, E)$ is a sequence of vertices of V linked by edges of E : $h = (v_0, v_1, \dots, v_n)$, $v_i \in V$. We will note $H(\cdot)$ a function which maps a graph G to a given set of walks (for example all walks without cycles).

Kashima *et al.* [10] compared two graphs G and G' by a kernel comparing all possible walks of same length on both graphs:

$$K_K(G, G') = \sum_{h \in H(G)} \sum_{h' \in H(G')} K_C(h, h') p(h|G) p(h'|G') \quad (1)$$

with $p(h|G)$ the probability of finding walk h in graph G and $|h|$ the length of h *i.e.* its number of edges.

This class of kernel is used in the framework of graphs of molecules, where the similarity between vertices is binary, a vertex (an atom) is or is not the same as the vertex of the other graph. But when the similarity between two vertices takes real values, this function tends to bury the similarities between walks in the sum. In [23], the kernel is computed by counting the number of common walks (only the edges are labeled). The problem of these kernels is that they use all random walks of the graphs and their number may be infinite (especially if cycles are allowed). This leads to a high computational complexity, and if this is acceptable with graphs with symbolic values, it is unaffordable with our attributed graphs. To deal with this problem, which is important in semi-supervised learning, kernel K_{max} takes the maximum of all similarities of all walks of same length.

$$K_{max}(G, G') = \max_{h \in H(G)} \max_{h' \in H(G')} K_C(h, h') \quad (2)$$

A similar kernel was used in [17] without the restriction of walks of similar lengths to perform image retrieval.

In between both formulas, a kernel on graphs was proposed in [11] for image retrieval tasks. It is faster to compute than Kashima kernel, since it limits the number of walks. In order to be sure to consider each vertex at least once, each vertex v_i is the beginning of one walk h_{v_i} . And in order to reduce the search in graph G' , the most similar walk of same length in G' is searched from the most similar vertex $s(v'_i)$ of V' :

$$K_L(G, G') = \frac{1}{|V|} \sum_{i=1}^{|V|} \max K_C(h_{v_i}, h'_{s(v_i)}) + \frac{1}{|V'|} \sum_{i=1}^{|V'|} \max K_C(h_{s(v'_i)}, h'_{v'_i}) \quad (3)$$

with:

$$\begin{cases} h_{v_i} \text{ is a walk of } G \text{ whose first vertex is } v_i \\ h'_{s(v_i)} \text{ is a walk of } G' \\ \text{whose first vertex is the most similar to } v_i \end{cases}$$

K_K adds the similarities of all walks of all lengths, it takes into account all vertices and all edges but with a lot of redundancy. K_{max} equals the similarity of the two walks the most similar; as these walks do not necessary go through all vertices, some of them do not intervene in the similarity. K_L takes into account all vertices, since it takes the average of the best similarities for walks starting from each vertex.

Concerning the kernels on walks, several K_C were proposed (sum, product, *etc.*) [11]. We tested all these kernels and the best results were obtained with the following one, where e_j denotes edge (v_{j-1}, v_j) :

$$K_C(h_{v_i}, h') = K_v(v_i, v'_0) \times \sum_{j=1}^{|h|} K_e(e_j, e'_j) \times K_v(v_j, v'_j)$$

For minor kernels K_v and K_e , we used Gaussian kernel with χ^1 distance for K_v which is the most adapted for histogram features and only binary values for K_e (1 if there is an edge, 0 if not).

3.2 Graph matching algorithm

The computational complexity depends on the number of walks of $H(G)$, on the similarity function and on the kernel on walks. K_K kernel requires an exhaustive (Eq. (1)) comparison of all walks, since it performs the sum of all similarities between walks. If an incomplete solution is sufficient, the only way to reduce the computation with this kernel is to bound the length of the walks. On the contrary, to compute K_{max} kernel (Eq. (2)), only a part of the walks needs to be compared, the search of the maximum can be easily obtained by the branch and bound algorithm, and an incomplete solution is also reachable through a pruning of the search tree. K_L (Eq. (3)) can be computed without computing all walk comparisons.

A search tree is used to represent all the walk comparisons. This representation allows to perform the computation with the *branch and bound* algorithm, whose recursion well suits the recursion of walks.

For two graphs $G = (V, E)$ and $G' = (V', E')$ and a generative function of set walks H , our search tree is composed of:

- a root
- nodes: each node represents a match of two vertices $n = (v, v') \in V \times V'$.
- a link between two nodes $n_1 = (v_1, v'_1)$ and $n_2 = (v_2, v'_2)$ means that there is an edge $e_{1,2}$ between v_1 and v_2 in $H(G)$, and an edge $e'_{1,2}$ between v'_1 and v'_2 in $H(G')$.

A path in the search tree starts from the root and is a succession of matched vertices. It corresponds to the match of two walks. The main interest of the search tree is that it does not need to be built completely, it can be pruned during its building. For instance, for a search of a maximum, only the useful branches are built.

The branch and bound algorithm aims at finding optimal solutions to problems whose goal is to search the maximal value of a function. It is especially adapted to solve the search of the maximum for functions whose bounds can be predicted on a given subset (in our case, a subset of walks).

In our case, the function is the kernel function K_C which can be recursively written:

$$K_C(h_{i+1}, h'_{i+1}) = K_C(h_i, h'_i) \times K_e(e, e') K_v(v, v')$$

with: $h_{i+1} = h_i$ prolonged by e and v .

4. EXPERIMENTS: OBJECT RETRIEVAL

RETIN-3D is a search engine dedicated to 3D objects. It can be used to browse a database or to retrieve a category of objects. RETIN-3D includes a user-friendly interface (*cf.* fig. 3). On the right part of the screen, one object can be displayed. It can be zoomed, turned, *etc.* It can be displayed as a plain meshed model or with attributed meshes. For example in fig. 3, each mesh is colored according to its distance to the center of the object.

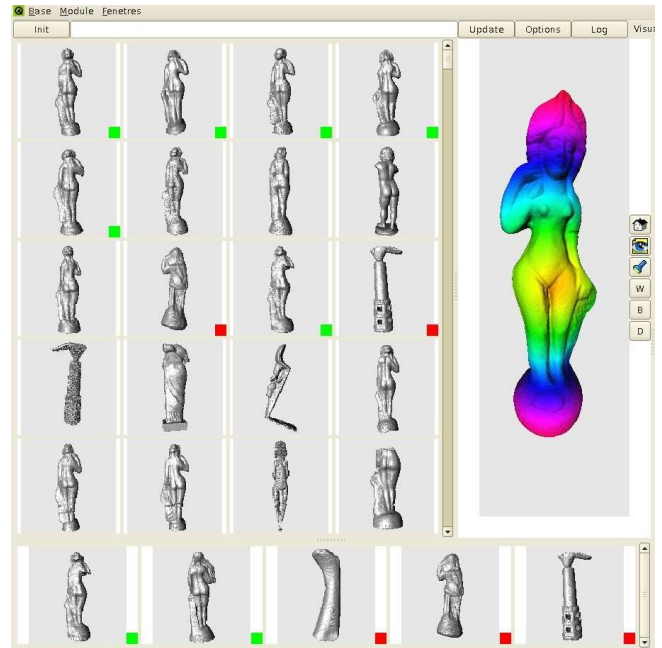


Figure 3: RETIN-3D user interface. Left: the 3D models are ranked from top to bottom by their classification rate, top left is the model query; relevant (resp. irrelevant) objects are annotated with a green (resp. red) mark. Right: a Venus statue colored according to distances to the object center. Bottom: the active learning panel.

The system proposes 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 includes 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” and interactive.

To achieve such a classification task in two classes with very few examples, we used Support Vector Machines (SVM) [3], 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. As the learning is performed with very few examples, RETIN-3D works with an active learning scheme [3]. The “active learning panel” on the bottom of the screen (*cf.* fig. 3) 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 thus performed online and according to the user expectations, hence it gives a great flexibility to the system.

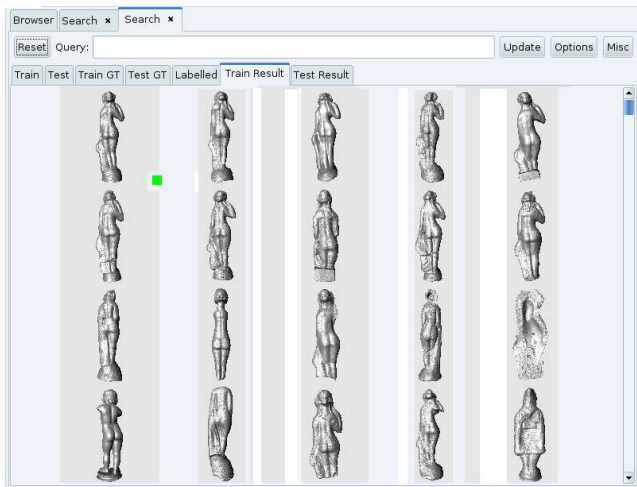


Figure 4: Example of a retrieval session with a Venus statue as query. Models are ranked according to their similarity with the query (top left model). Similarity with graph kernel.

We compared object retrieval using a global representation and a representation by graphs of regions, using the same descriptor. We first used Cord1D descriptor with 256 bins, on one hand computed on the whole model and on the other hand computed on each region of the surface after segmentation. When using the global descriptor, we need 5 positive examples and 5 negative examples, to retrieve 14 Venus statues amongst the 20 first retrieved objects (*cf.* fig. 3). With the graph representation and the kernel of Eq. 3, we only need one positive example to retrieve 17 Venus statues amongst the 20 first retrieved objects (*cf.* fig. 4). It is remarkable that among the artwork pieces returned by the system, we find some broken pieces, and also Venus moulds. The database includes many broken pieces, moulds and fragments. Therefore it is difficult to build a ground truth, thus we do not present any MAP or ROC curves.

In a second experiment, we used CEGI descriptor with 256 bins, computed both on the entire models and on their regions after segmentation. We compared both representations on Mother-divinity figurines. After two feedback steps with 2 models annotated as relevant and 7 as irrelevant, we obtained results of fig. 5 for search on global models, and 6 for search on regions.

With the same protocol and various descriptors, we obtained the following number of mother-divinity models among the 30 first retrieved objects (*cf.* table 1), showing the retrieval improvement that the kernel on graphs brings.

We did the same experiments with only 3 relevant and 3 irrelevant models (table 2), leading to similar conclusions.

5. CONCLUSION

We have shown that a representation of artwork objects by a graph of surfacic regions is a better solution than a global representation. This representation by graph of regions can be obtained by a segmentation of the mean curvature map and simple region features. To compare graphs of regions, we used kernel functions which take into account both the similarity between regions and their adjacency. Our search

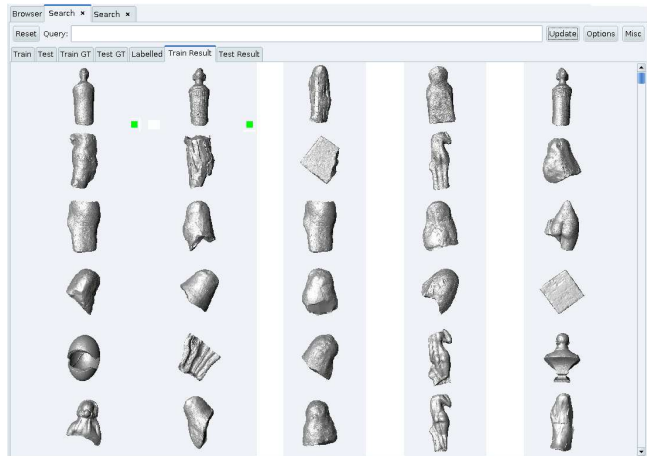


Figure 5: Mother-divinity figurines, global description with the CEGI descriptor (256 bins). 2 models were annotated as relevant and 7 as irrelevant. 4 models are relevant in the 30 first retrieved.

Table 1: Number of relevant models (mother-divinities) in the 30 first retrieved. 2 models of the database were annotated as relevant and 7 as irrelevant.

Descriptor	Global	Graphs
CEGI	4	23
Cord1D	14	22
Cord2D	9	12
EGI	17	22

engine, associated with a user-friendly interface allows the retrieval or the classification of an artwork database into various categories.

Other kernel functions, coding more precisely the relative positions (such as above or left to) could be added to the kernel on edges. Another application of the representation by graph is the matching of a fragment with a complete model.

6. REFERENCES

- [1] M. Alcoverro, S. Philipp-Foliguet, M. Jordan, L. Najman, and J. Cousty. Region-based 3d artwork indexing and classification. In *Proceedings of the IEEE 3DTV-Con Conference*, Istanbul, May 28-30 2008.
- [2] B. Bustos, D. A. Keim, D. Saupe, T. Schreck, and D. V. Vranić. Feature-based similarity search in 3D object databases. *ACM Comput. Surv.*, 37(4):345–387, 2005.
- [3] 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.
- [4] J. Cousty, G. Bertrand, L. Najman, and M. Couprie. Watershed cuts. In *8-th International Symposium on Mathematical Morphology*, pages 301–312, 2007.
- [5] P. Daras, D. Zarpalas, D. Tzovaras, and M. Strintzis. Efficient 3-D Model Search and Retrieval Using

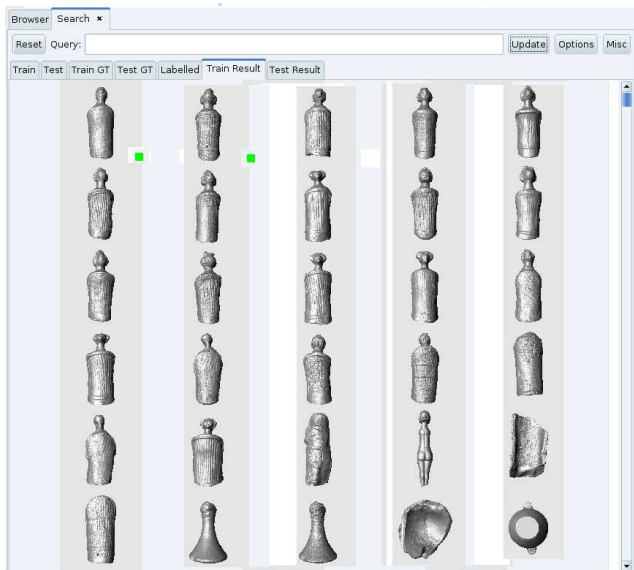


Figure 6: Mother-divinity figurines, CEGI descriptor (256 bins) computed on surfacic region graphs. 2 models were annotated as relevant and 7 as irrelevant. 23 models are relevant in the 30 first retrieved.

Table 2: Number of relevant models (mother-divinities) in the 30 first retrieved. 3 models of the database were annotated as relevant and 3 as irrelevant.

Descriptor	Global	Graphs
CEGI	9	26
Cord1D	9	21
Cord2D	11	9
EGI	20	24

Generalized 3-D Radon Transforms. *IEEE Trans. on Multimedia*, 8(1):101–114, Feb. 2006.

- [6] D. Emms, R. C. Wilson, and E. R. Hancock. Graph matching using the interference of continuous-time quantum walks. *Pattern Recognition*, 42(5):985–1002, 2009.
- [7] T. Funkhouser, P. Min, M. Kazhdan, J. Chen, A. Halderman, D. Dobkin, and D. Jacobs. A Search Engine for 3D Models. *ACM Transactions on Graphics*, 22(1):83–105, Jan. 2003.
- [8] B. Horn. Extended Gaussian Images. *Proceedings of the IEEE*, 72(12):1671–1686, dec. 1984.
- [9] S. 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.
- [10] H. Kashima and Y. Tsuboi. Kernel-based discriminative learning algorithms for labeling sequences, trees and graphs. In *International Conference on Machine Learning (ICML)*, page 58, Banff, Alberta, Canada, 2004.
- [11] J. Lebrun, S. Philipp-Foliguet, and P.-H. Gosselin. Image retrieval with graph kernel on regions. In *19th ICPR International Conference on Pattern Recognition*. IEEE, USF, IEEE, dec 2008.
- [12] E. A. Lmaati, A. E. Oirrak, D. Aboutajdine, M. Daoudi, and M. N. Kaddiou. A 3-D Search engine based on Fourier series. *Computer Vision and Image Understanding*, 114(1):1–7, 2010.
- [13] L. Najman and M. Couprie. Building the component tree in quasi-linear time. *IEEE Trans. on Image Processing*, 15(11):3531–3539, nov. 2006.
- [14] R. Ohbuchi, T. Minamitani, and T. Takei. Shape-Similarity Search of 3D Models by using Enhanced Shape Functions. In *Theory and Practice of Computer Graphics 2003 (TPCG 2003)*, pages 97–104, 2003.
- [15] R. Ohbuchi, A. Yamamoto, and 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*, pages 31–40, Augsburg, Germany, 2007.
- [16] 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.
- [17] S. Philipp-Foliguet, J. Gony, and P.-H. Gosselin. FRBIR: an image retrieval system based on fuzzy region matching. *Computer Vision and Image Understanding*, 113(6):693–707, June 2009.
- [18] S. Philipp-Foliguet, M. Jordan, L. Najman, and J. Cousty. Artwork 3D Model Database Indexing and Classification. *Pattern Recognition*, 2010. To appear.
- [19] B. Schölkopf and A. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, 2002.
- [20] A. Shokoufandeh, L. Bretzner, D. Macrini, M. Fatih Demirci, C. Jönsson, and S. Dickinson. The representation and matching of categorical shape. *Computer Vision and Image Understanding*, 103(2):139, 2006.
- [21] J. W. Tangelder and R. C. Veltkamp. A Survey of Content Based 3D Shape Retrieval Methods. *Multimedia Tools Appl.*, 39(3):441–471, 2008.
- [22] R. C. Veltkamp and F. B. ter Haar. SHREC2008: 3D Shape Retrieval Contest. In *Shape Modeling and Applications*, pages 215–263, 2008.
- [23] S. V. N. Vishwanathan, N. N. Schraudolph, I. R. Kondor, and K. M. Borgwardt. Graph kernels. *Journal of Machine Learning Research*, 10, 2009.
- [24] E. Wahl, U. Hillenbrand, and G. Hirzinger. Surflet-Pair-Relation Histograms: A Statistical 3D-Shape Representation for Rapid Classification. *Int. Conf. on 3D Digital Imaging and Modeling*, 2003.