Eurographics Workshop on 3D Object Retrieval (2013)
S. Biasotti, I. Pratikakis, U. Castellani, T. Schreck, A. Godil, and R. Veltkamp (Editors)

Learning Kernels on Extended Reeb Graphs for 3D shape classification and retrieval

V. Barra[1,2] and S. Biasotti[3]

[1]Clermont-Université, Université Blaise Pascal, LIMOS, BP 10448, F-63000 CLERMONT-FERRAND
[2]CNRS, UMR 6158, LIMOS, F-63173 AUBIERE
[3]Istituto di Matematica Applicata e Tecnologie Informatiche 'E. Magenes', CNR, Italy

Abstract

This paper addresses 3D shape classification and retrieval in terms of supervised selection of the most significant features in a space of attributed graphs encoding different shape characteristics. For this purpose, 3D models are represented as bags of shortest paths defined over well chosen Extended Reeb graphs, while the similarity between pairs of Extended Reeb graphs is addressed through kernels adapted to these descriptions. Given this set of kernels, a Multiple Kernel Learning algorithm is used to find an optimal linear combination of kernels for classification and retrieval purposes. Results are comparable with the best results of the literature, and the modularity and flexibility of the kernel learning ensure its applicability to a large set of methods.

Categories and Subject Descriptors (according to ACM CCS): Computer Graphics [I.3.6]: Methodology and Techniques—Information storage and retrieval [H.3.3]: Information search and Retrieval—

1. Introduction

The fast advancement of tools for acquisition and storage of 3D models has led to a rapid increase of the number and size of the models available on the internet and domain-specific databases. Organizing these 3D models is becoming an acute issue for numerous applications, including CAD, medical imaging, molecular biology, architecture or game design. In this scenario, one fundamental problem is how to select and combine different features. Often objects that are in a same class share characteristics that are only implicitly available: the challenge is then to recognize from a large set of features, the subset that better characterizes the class itself. Shape features are often of different scales and therefore their combination is not a matter of linear combination. In this paper we propose an effective 3D shape classification and retrieval method that selects the most significant shape features from a larger set. The construction of the representative set can be regarded as a machine learning task that uses a supervised learning technique to capture the high-level semantic concepts of the classes. To deal with a semantic representation able to couple global and local features, we adopt a shape description that combines the overall shape structure (coded in a topological graph) with a local geometric description (the spherical harmonic indices of the shape parts).

The 3D shape classification and retrieval problem is addressed as a search in a space of attributed graphs encoding different shape characteristics through a similarity measure able to handle both the graph structure and the geometric attributes associated to nodes and edges. The most efficient linear combination of features is then computed using supervised learning. More precisely, 3D models are represented by bags of shortest paths defined over a set of Extended Reeb graphs (ERGs) computed from a set of functions. For each of these functions, the similarity between pairs of corresponding ERGs are computed using a so called kernel, implicitly defining the similarity between the models. Given this set of kernels, a Multiple Kernel Learning algorithm is used to find an optimal linear combination for classification and retrieval purposes. This linear combination can be class-specific or computed for a whole database. Results are comparable with the literature, and the modularity and flexibility of the kernel learning ensure its applicability to a large set of methods.

The paper is organized as follows. Section 2 reviews 3D shape classification and feature selection methods. Section 3 introduces the elements of our method, i.e. ERGs and kernel on graphs, while section 4 develops the method we propose, base on multiple kernel learning. Section 5 presents and analyses the classification and retrieval perfor-

DOI: 10.2312/3DOR/3DOR13/025-032

mance of the optimal kernel over two benchmark datasets [GBP07, GM08].

2. State of the art

For efficient comparison and similarity estimation, 3D models can be represented with a set of meaningful descriptors that encode the salient geometric and topological characteristics of their shapes. The objects in the database are then ranked according to their distance to the descriptors of the query model, see [BKS*05, TV08] for an overview of 3D shape retrieval methods.

The use of data-driven approaches to learn the salient features of a 3D model and the similarity measure that is suitable to the data set can improve significantly the performance of classification and retrieval algorithms. Even though several methods for shape comparison have been proposed, only few methodologies address the issue of identifying descriptions that capture the shape features shared by a class of models [Lag10, MCHB07]. On one hand, a solution to learn inter-class distances is to manually feed the retrieval system with relevance feedback techniques [NWS07, GFSF10]. On the other hand, to automatically address the selection of class features, machine learning techniques like boosting and support vector machines, have been adopted either using images [TV04] or shape descriptions [HLR05]. For instance the AdaBoost algorithm [FS95] has been used in [Lag10] to select relevant views of 3D objects with respect to the light field descriptor.

Other classifiers based on semi-supervised learning, dimensionality reduction, and probability have been successfully exploited for shape classification. For instance, in [HLR05] Support Vector Machine is used to cluster 3D models with respect to semantic information. In [OK06] shape classifiers are obtained as a linear combination of individual classifiers and using non-linear dimensionality reduction. In [SF06], relevant local shape descriptors are selected through a multivariate Gaussian distribution and collected to define a priority-driven search for shape retrieval. [MPSF11] uses Adaboost and SVM as tools to automatically select the frequencies of the Laplacian spectrum of 3D shapes that are relevant for classification. There a selection of the eigenvalues is used to represent each class by means of those features that characterize the class members and that are discriminative with respect to non-member 3D objects. Unfortunately, there is not an explicit correspondence between the eigenvalues and the geometric meaning of the features selected by the boosting algorithm.

On the contrary, our aim is to automatically build a new composite description from a set of elementary kernels. In recent years, several methods have been proposed to combine multiple kernels instead of using a single one. These different kernels may correspond to using different notions of similarity or may be using information coming from multiple sources (different representations or different feature subsets). In particular [GA11] provides a taxonomy of existing methods for Multiple Kernel Learning and review several algorithms.

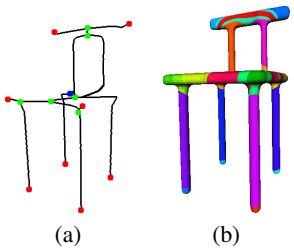


Figure 1: *An ERG (a) and the corresponding object partition (b). Blue=minimum, red=maxima and green=saddles. Different colors in the model (b) represent the different parts (each part corresponds to a node or an edge).*

3. Background

In this section we overview how to extract a graph-based description from a 3D shape, namely an Extended Reeb graph, and to define kernels on these descriptions. In the following, $G = (V, E, \delta_V, \delta_E)$ is an undirected, labelled graph of nodes V and edges E where $\delta_V : V \to \mathbb{R}^p$, $\delta_E : E \to \mathbb{R}^d$ associate to every node and edge numerical attributes. We assume that there are no multiple edges from one node to another in G and that G contains no negative cycles.

3.1. Extended Reeb graph

The Extended Reeb graph (ERG) is a 3D shape descriptor that fulfils the graph requirements on G of being an undirected and labelled graph. Like other methods based on the Reeb graph [Ree46], the ERG reflects in the graph coding the invariance properties of a real function f defined on the shape. When f is differentiable, nodes of the graph correspond to critical points of f, see [BGSF08] for an overview of Reeb graph-based descriptions. The ERG approximates the topological structure of the Reeb graph on the basis a sampling of the image of f. The characterization of f in terms of maxima, minima and saddles depends on the behaviour of f along the level sets: the nodes of the ERG correspond to *areas* instead of critical points while its arcs are detected tracking the evolution of the level sets [Bia05]. Then, we couple the topological information stored in the ERG with a geometric representation of the parts of the model that correspond to nodes and edges of the graph. Figure 1 depicts the ERG and the model parts associated to nodes and edges.

From the storage point of view, each node v of G corresponds to one critical area (a maximum, a minimum or a saddle-like area) and each edge $e = (v_1, v_2)$ is associated to the surface portion bounded by the regions r_1 and r_2 associated to v_1 and v_2, respectively. For each node, the attributes

$\delta_V(v)$ and $\delta_E(e)$ correspond to the spherical harmonic values [FMK*03] of the related parts. If the regions r_1 and r_2 corresponding to v_1 and v_2 are adjacent (i.e., r_1 and r_2 share a boundary element), $\delta_E(e) = 0$. In case of multiple edges $(e_1, \ldots, e_n)$ between v_1 and v_2, the attributes $\delta_E(e_i)$ codes the spherical harmonic indices related to the surface portions (e.g., the rear segments of the chair in Figure 1(b)).

The most interesting aspect of this descriptor is its parametric nature: changing f allows different descriptions of the same surface that highlight different "local" features while preserving the "global" topological structure of the surface. Here, we focus on ten scalar functions, see Section 4.1.

3.2. Graph similarity and kernel on graphs

The problem we face here is to find, given two such graphs G and G', a similarity measure that can be geometric, topological or can manage both of these notions, provided that sufficient information is carried out in the graphs. Several methods are available, most of them derived from graph matching techniques. Recently, several authors proposed as an alternative to define matching techniques based on kernels [KTI03, VSKB10] and used them either for retrieval and indexing [PFJFG10], classification [ABK06] or scene analysis [FSH11]. Analogously, we base the similarity of graphs on the similarity of their representations by bag of paths.

The idea of building kernel between graphs originated from [GFW03] and was further extended by [BK05] and [KTI03] in the case of marginalized kernels. Defining a kernel only from the values of δ_V and δ_E is possible, but it does not manage the graph structure. Then, the main principle is to define a specific kernel for both nodes and edges, and to gather the corresponding results into a high level kernel, defining a dot product between graphs.

Given $G = (V, E, \delta_V, \delta_E)$, a path p is a sub-graph of G defined by a sequence of l nodes $p = (v_1 \cdots v_l)$ such that for each $i, (v_i v_{i+1}) \in E$. The length of p is defined as $l(p) = f(\delta_V(v_i), v_i \in p, \delta_E(e), e \in p)$, where f is a real function defined on the set of nodes and edges. p is said to be the shortest path between v_i and v_j if for all path p' between these two nodes, $l(p) < l(p')$.

The kernel formulation is expressed in terms of bag of paths [KJ03]. Graphs $G = (V, E, \delta_V, \delta_E)$ and $G' = (V', E', \delta_{V'}, \delta_{E'})$ are described as bag of paths H and H', and the similarity $K(G, G')$ between G and G' is assessed through the similarity between H and H', using a predefined kernel K_c on paths :

$$K(G, G') = \sum_{h \in H} \sum_{h' \in H'} K_c(h, h') P(h|G) P(h'|G')$$

where $P(h|G)$ (respectively $P(h'|G')$) is the probability of walking along h (resp. h') on G (resp. G'). The K_c function is computed from two basic kernels K_V and K_E defined on V and E. In the case of Gaussian kernels:

$$\forall (v, v'), K_V(v, v') = e^{-\frac{1}{2}\left(\delta_V(v) - \delta_{V'}(v')\right)^T \Sigma_V \left(\delta_V(v) - \delta_{V'}(v')\right)}$$
$$\forall (e, e'), K_E(e, e') = e^{-\frac{1}{2}\left(\delta_E(e) - \delta_{E'}(e')\right)^T \Sigma_E \left(\delta_E(e) - \delta_{E'}(e')\right)}$$

where $\Sigma_V = diag\left(\frac{1}{\sigma_i}\right)$ (resp. Σ_E) is diagonal, and σ_i is the bandwidth of the i^{th} label of v (resp e).
Then if $h = (v_1 \cdots v_l), h' = (v'_1 \cdots v'_l)$, with $v_{i-1} v_i = e_i$ and $v'_{i-1} v'_i = e'_i$: $K_c(h, h') = K_V(v_1, v'_1) \prod_{i=2}^{l} K_V(v_i, v'_i) K_E(e_i, e'_i)$

The probabilities $p(h|G)$ can easily be computed with $P(h|G) = p_S(v_1) \prod_{i=2}^{l} p_T(v_i|v_{i-1}) p_E(v_l)$, where p_S is the probability of initial visit, p_E is the ending probability and p_T is the transition probability between nodes in G. In theory, it is thus possible to compute probability values even for a large number of nodes, and [KTI03] proposed a recursive scheme and a convergence criteria allowing an efficient computation. Nevertheless, if it is possible to define an infinity of paths in an unoriented graph, some of them are not relevant and do not need to be explored. We propose then to filter out the set of all paths and use only the shortest paths of length less or equal to a maximal length L_{max}. The resulting kernels are easy to compute, retain expressivity and are still positive definite. Finally, considering shortest paths between vertices naturally prevents from tottering phenomenon.
We propose to replace the set of all paths by the set of shortest paths of maximal length L_{max} and to compute K by only retaining the best comparison values for all shortest paths, leading to the max matching kernel formulation [SRB07]:

$$K(G, G') = \frac{1}{2}\left[\hat{K}(H, H') + \hat{K}(H', H)\right]$$

where $\hat{K}(H, H') = \frac{1}{|H|} \sum_{h \in H} \max_{h' \in H'} K_c(h, h')$, and H (resp. H') is the set of shortest paths in G (resp. G') of maximal length L_{max}. But since this definition does not induce a definite positive kernel, we use an approximation proposed in [HB04]:

$$\max_{h' \in H'} K_c(h, h') \approx \sum_{h' \in H'} K_{dc}(h, h')$$

with $K_{dc} = exp\left(-\frac{d_c(h,h')^2}{2\sigma^2}\right)$, a kernel defined by the distance d_c induced by K_c, $\sigma > 0$.

4. Learning kernels from functions

Our approach automatically correlates kernels with respect to the classes of a database. If we think to each kernel as an user's filter of the features of a dataset, the selection of the kernels that better hold retrieval implicitly defines the class complexity and the invariants that characterize it. In our framework it is thus possible to compute a class-specific function (a kernel) and for each class to learn a kernel using MKL and a one class SVM. The set of descriptions is not

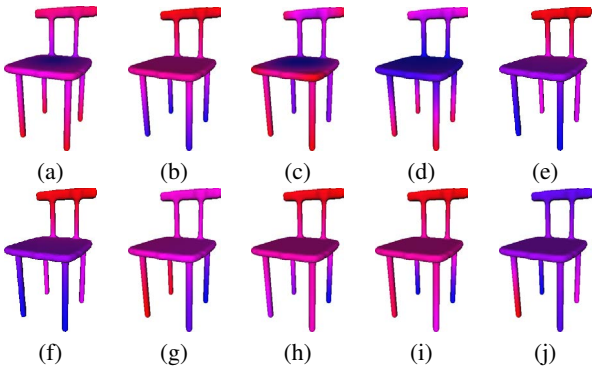


Figure 2: *The set of real functions in our framework. Colors represent the function, from low (blue) to high (red) values.*

necessarily orthogonal and admit redundancy and overlapping in the feature space [TV04].

4.1. Real functions

The role of a single function f is to convey the most significant shape information and act as a filter of the features that will be stored in the shape descriptor [BDF*08]. The set $F = \{f : S \to \mathbb{R} | f continuous\}$ is infinite, anyway an appropriate selection of the functions is necessary to make the descriptor suitable for shape matching issues: for instance, the function f has to be invariant from object rotation, translation and scaling. In the large number of functions available in the literature, we are considering:

- the distance from the barycentre B of the object, $Bar(p) = d_E(p,B)$, $p \in S$ and d_E represents the usual Euclidean distance (Figure 2-a);
- the distance from the axis $\vec{\alpha}$, $MSA(p) = d_E(p,\vec{\alpha})$ where $\vec{\alpha} = \frac{\sum_{p\in S}(p-B)\|p-B\|}{\sum_{p\in S}\|p-B\|^2}$, where B is the barycentre of S (Figure 2-b);
- the function $MSANorm(p) = \|\vec{\alpha} \times (p-B)\|$, $p \in S$, $\vec{\alpha}$ is the same as above and B is the barycentre (Figure 2-c);
- an average of the geodesic distances, $Geodesic(p) = \frac{\sum_{v\in S} g(v,p)}{\max_{v\in S} g(v,p)}$, where $g(v,p)$ is the geodesic distance between a sampled vertex v and p, in the experiments we consider a uniform re-sampling R of S made of 256 vertices and $v \in R$, [HSKK01] (Figure 2-d);
- the first three (ranked with respect to the decreasing eigenvalues), non-constant eigenfunctions of the Laplace-Beltrami operator of the mesh computed according to [BSW08], $LAPL_i, i = \{1,2,3\}$, (Figure 2(e-g));
- a mix of the first three eigenfunctions of the Laplace-Beltrami operator obtained according to the rule: $MIX_{i+j-2} = (LAPL_i)^2 - (LAPL_j)^2$, $i = \{1,2\}$, $j = \{2,3\}$, $i \neq j$ (Figure 2(h-j)).

The choice of these functions is motivated by the different behaviour they have with respect to different aspects, each one reflecting either intrinsic or extrinsic shape features. For instance, the distance from the barycentre highlights the distribution of the object with respect to its barycentre. Therefore such a function is rotation invariant with respect to rotations around the barycentre but sensitive to pose variations. Similarly the distances from the principal shape axis and its orthogonal are independent of axis rotations and independent of axis symmetries. On the contrary the geodesic-based and the Laplacian-based functions are isometry-invariant and therefore pose invariant because they approximate the intrinsic Riemannian metric of the surface [BBK06]. In this way, protrusions and hollows are emphasized, also at different scales in the case of Laplacian-based functions, and the graph representation is independent of the different articulations of the objects.

The mix of the different properties (rigid or isometry invariant) guarantees that different shape features are coded. Although the insertion in the loop of new functions would influence the type of invariance detected from the method without modifying the global framework of our method.

4.2. The simpleMKL algorithm

We focus on the case where the kernel is learnt to be a convex combination of given base kernels. Following [ASYS08], we indeed think that "no single descriptor is capable of providing fine grain discrimination required by prospective 3D search engines". We then design the algorithm such that several structural/feature based kernels can be combined to enhance the retrieval result.

Although there are not large differences in terms of accuracy of the Multiple Kernel Learning algorithms reviewed in [GA11], there is difference between them in complexity as given by the number of stored support vectors, the sparsity of the solution as given by the number of used kernels, and training time complexity. We therefore adopt the *simpleMKL* [RBCG08] algorithm.

Multiple Kernel Learning (MKL) was first introduced in [LCB*04], and was subject to enhancements, extensions and algorithms in e.g [SRSS06, RBCG08]. The idea behind MKL is to look for a different solution of the learning problem: classically the solution is written in the form $f(x) = \sum_{i=1}^{l} \alpha_i^* K(x,x_i) + b^*$, where α_i^* and b^* are some coefficients to be learned from examples x_i and $K(.,.)$ is a given definite kernel associated with a reproducing kernel Hilbert space (RKHS) H. [LCB*04] proposed to consider that $K(.,.)$ is a convex combination of basis kernels K_m

$$K(x,y) = \sum_{m=1}^{M} d_m K_m(x,y), \text{ with } d_m \geq 0, \sum_{m=1}^{M} d_m = 1,$$

where M is the total number of elementary kernels. MKL learnt both the α_i^* and d_m in a single optimization problem,

and [RBCG08] proposed to solve the equivalent problem:

$$\begin{aligned} \underset{\{f_m\},b,\xi,d}{\text{Minimize}} \quad & \frac{1}{2}\sum_m \frac{1}{d_m} \parallel f_m \parallel^2_{\mathrm{H}_m} + C\sum_i \xi_i \\ \text{subject to} \quad & y_i \sum_m f_m(x_i) + y_i b \geq 1 - \xi_i, \forall i, \\ & \xi_i \geq 0, \forall i, \\ & \sum_m d_m = 1, d_m \geq 0, \forall m, \end{aligned}$$

where H_m are the RKHS associated to each $K_m(.,.)$ and each f_m belongs to a different H_m. It is assumed that one looks for a decision function of the form $f(x) = \sum_m f_m(x)$. This problem can be extended to a one class problem, and we used this approach in the following using as K_m the kernels on ERG computed from the functions detailed in section 4.1.

5. Results

The results shown in this Section mainly focus on the capability of the method of learning the most relevant features of each class rather than in the absolute retrieval and classification performance of this specific technique.

5.1. Datasets

We evaluate the performance of our method on two datasets:

- The database of the SHape REtrieval Contest SHREC'07- [GBP07]. The collection is composed of 400 watertight mesh models, subdivided into 20 classes of 20 elements each. Ground truth was manually established so that the classes exhibit sufficient and diverse variation, from pose change to shape variability in the same semantic group.
- The database of 646 watertight models of the SHREC'08- [GM08] contest. These models were classified, and released as training, testing and query data. To reflect both functional and geometric similarity this dataset has three different levels of categorization: from coarse to fine. At the *coarse* level, objects are classified according to semantic criteria, besides their shape; at the *intermediate* level, the classes are subdivided according to both functionality and shape features; while at the *fine* level, the classes are further partitioned on the basis of the geometric similarity. For instance, at the coarse level some objects were classified into the furniture class. At the intermediate level, these same objects were further divided into tables, seats and beds. At the fine level, the objects were classified into chairs, armchairs, stools, sofa and benches.

The ERG graphs were extracted from these models using functions defined in section 4.1 and all labelled using the spherical harmonics on both nodes and edges. In our experimental settings the ERGs are computed automatically dividing the image of the function f in 16 intervals uniformly distributed. The kernel similarity measure was applied on these labelled graphs for the retrieval process. In the following, each model in each dataset was used in turn as a query against the remaining part of the database. For a given query, the goal of the track was to retrieve the most similar objects.

5.2. Parameters estimation

The ERG graphs we processed have a maximal number of nodes less than 100. We conducted several experimental studies on the SHREC databases, using a leave-one-out cross validation procedure, to determine the best L_{max} value, from 0 (only nodes are considered) to the number of nodes in the graph. As the considered path length increased (a structural information was added in the graph description), the retrieval rate and the quantitative indexes increased then decreased when a given path length was reached. Any value of L_{max} ranging in $[\![3 \cdots 7]\!]$ gave both good retrieval results and low complexity, and we performed all our experiments with $L_{max} = 5$, *i.e.* the similarity of graphs is based on bag of shortest paths of maximal length equal to 5.

The other parameters that have to be estimated are related to the matrices Σ_E and Σ_V that appears in the definition of K_V and K_E in Section 3.2. These parameters express the bandwidths σ_i of the i^{th} spherical harmonics of v (resp. e). Accurately defining these bandwidths is crucial since they allow to define the intensity of the corresponding kernel. Several methods can be used to determine these values. However, to be database-specific, we choose a cross validation approach and we apply a leave-one-out procedure dataset to determine the best values for the bandwidths.

Several experiments were performed using as training set a percentage ranging from 10% to 50% of the models in each class. We present in this article the results obtained using half of the elements in each class but roughly the same results were obtained with the 20% of the models. With the 25% of the class, there was no overfitting and most of the other 75% of models was successfully retrieved thus having a good generalization capacity.

5.3. Results on SHREC'07

Each class of the SHREC'2007 database was processed separately using a one class SVM inside MKL. MKL thus faces a 2-class classification problem, and outputs the corresponding optimal convex combination of elementary kernels. In the experiments reported in this Section, the learning step was performed using 10 models per class (out the 20 available for each class). Figure 3 presents the weights of this convex combination for all classes. The choice of the weights mostly depends on the intra-class variability. Anyway, LAPL1 and MIX1 are the most present functions for all the classes because many non-rigid deformations are present in the classes. Rigid functions such as barycentre and MSA are more discriminative when the intra-class variability is smaller.. e.g. armadillo, faces, etc.

Figure 4 shows the optimal kernel matrix provided by the

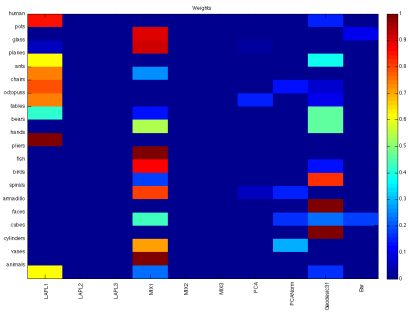


Figure 3: *Weights of the elementary kernels for each class of the SHREC'07 database*

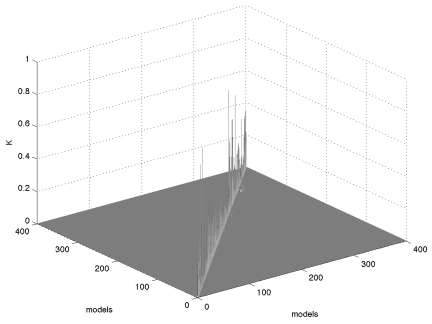


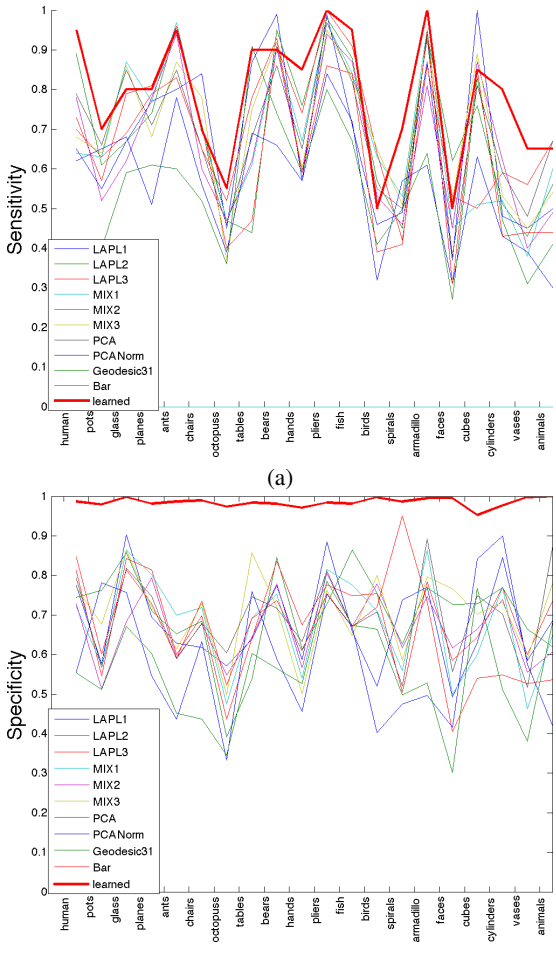
Figure 4: *Kernel matrix on the SHREC'07 database*

weights learned. The overall performance in terms of retrieval and classification rates over the whole SHREC'2007 dataset is comparable or overcome the results on the literature [GBP07] and a flat averaring of the functions [BBar]. In quantitative terms, ADR=8.82, NN= 100, FT=,0.88, ST=0.96 , where by ADR we mean average dynamic recall, NN is the nearest neighbour classifier, FT is the first tier and ST is the second tier.

The experimental results show that the optimal kernel is based on a convex combination of a few single kernels. The comparison of the performance of the optimal kernel with respect to the single ones was done analysing the sensitivity and the specificity over each class. Figure 5-a presents the sensitivity computed for both each elementary kernel and for the learned one (red bold curve). Figure 5-b shows the same results for the specificity. These two quantitative indexes were improved for almost every class, with a dramatic improvement for the specificity.

5.4. Results on SHREC'08

The results in Figure 6 show the choice of the weights for the optimal kernel on the SHREC'2008 dataset. In general, we see that the distance from the barycentre is the most relevant in all the three levels of categorization. Anyway, as far



(a)

(b)

Figure 5: *Sensitivity (a) and specificity (b) on the SHREC'07 database. Results for the elementary and the learned kernels*

as the classification is finer also the average geodesic distance, the distance from the main shape axis (MSA) and the mixed Laplacian-based functions come into the play. From our point of view, this fact is explained from the high intra-class variability of the shapes at the coarse level, where also functional variations are admitted and isometric invariants are not enough to characterize the class. On the contrary at the finest level of detail, a class is often made by a single shape and its non-rigid deformations: in this case, the descriptions based on eigenfunctions or geodesics are invariant to these shape variations and better characterize the intra-class variability.

The optimal kernels can also be computed and Figure 7 presents the kernel obtained from the third level on the training set. This kernel was used for classification purposes on this training set and classification performances were comparable to the one provided in [GM08].

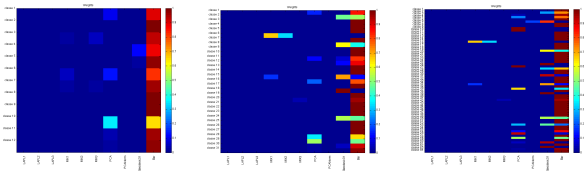


Figure 6: *Weights of the elementary kernels for each class of the SHREC'08 database for the 3 levels of categorization.*

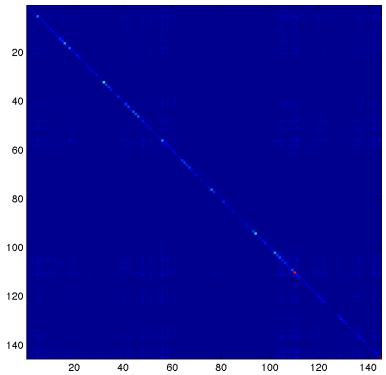


Figure 7: *Optimal kernel of the SHREC'2008 dataset - third level of categorization on the training set.*

In the experiments on the specificity and sensitivity of the learned kernels with respect to the single ones on the SHREC'2008 dataset we noticed a significant improvement in the quantitative results; Figure 8 presents the specificity curves of the single kernels and the learned one (red bold curve).

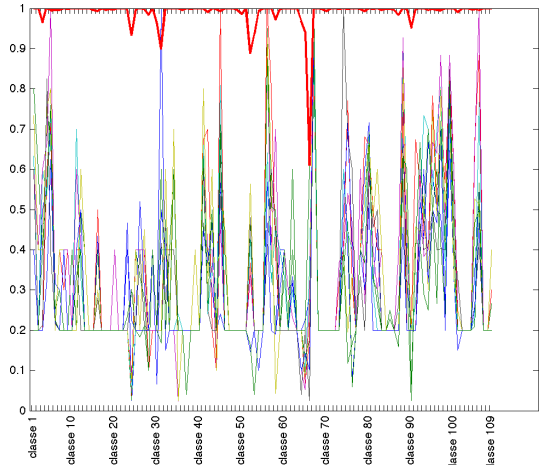


Figure 8: *Specificities of the SHREC'08 dataset - 3rd level of categorization. The learned kernel is the red bold curve.*

6. Conclusions

Combining several information in order to enhance the result of an algorithm is a quite common strategy in computer science. Examples include combination of classifiers (boosting methods) or combination of heterogeneous data (information fusion) in several domains (e.g. medical imaging, remote sensing). We adopt here this kind of strategy by computing several functions on model meshes, that provide complementary and redundant information for the retrieval process. We then deduce similarity measures using kernels on the Extended Reeb Graphs, and combine them to allow a final decision function.

Such a strategy offers several advantages. It is evolutive (it allows other functions to be included in the processing pipeline), modular (only informative subsets of functions can be used) and adaptive (the aggregation rule can be changed, and even learned using a learning set extracted from the original database). Differently from an aggregation of the kernels based on averaging or voting rules like that in [BBar], learning through the MKL strategy explictly yields a selection of the most relevant features without jeopardizing the retrieval and classification performances. These are good with respect to the reference methods, and most of the time outperform the state-of-the-art, with the advantage of being evolutive and modular. In addition, the use of kernel similarity measures based on the representation of graphs in terms of a bag of shortest paths with bounded maximal length allows an efficient computation.

Finally, our experiments highlight that only a fraction of the kernels is necessary to effectively retrieve and classify a shape and this selection is database dependent, varying according the intra- and extra-class variability of the shapes. We plan to further experiment the method on larger and heterogeneous datasets, in order to experimentally validate the assumption that some kernels better capture the "semantic" aspects of the class and others, e.g., geodesics, are better suited to detect intra-class variability aspects.

Acknowledgments This work has been partially developed in the CNR research activity ICT.P10.009, and partially supported by VISIONAIR, European project "FP7 INFRASTRUCTURES" (2011-2015).

References

[ABK06] ASSFALG J., BORGWARDT K. M., KRIEGEL H.-P.: 3dstring: a feature string kernel for 3d object classification on voxelized data. In *Proc. of the 15th ACM Int. Conf. on Information and knowledge management* (New York, NY, USA, 2006), ACM, pp. 198–207. 3

[ASYS08] AKGÜL C. B., SANKUR B., YEMEZ Y., SCHMITT F.: Similarity score fusion by ranking risk minimization for 3d object retrieval. In *EG 3D object retrieval Workshop.* (2008), pp. 41–48. 4

[BBK06] BRONSTEIN A. M., BRONSTEIN M. M., KIMMEL R.:

Efficient computation of isometry-invariant distances between surfaces. *SIAM J. Sci. Comput. 28*, 5 (Sept. 2006), 1812–1836. 4

[BBar] BARRA V., BIASOTTI S.: 3d shape retrieval using kernels on extended reeb graphs. *Pattern Recognition* (to appear). 6, 7

[BDF*08] BIASOTTI S., DE FLORIANI L., FALCIDIENO B., FROSINI P., GIORGI D., LANDI C., PAPALEO L., SPAGNUOLO M.: Describing shapes by geometrical-topological properties of real functions. *ACM Comput. Surv. 40*, 4 (2008), 1–87. 4

[BGSF08] BIASOTTI S., GIORGI D., SPAGNUOLO M., FALCIDIENO B.: Reeb graphs for shape analysis and applications. *Theoretical Computer Science 392*, 1–3 (2008), 5–22. 2

[Bia05] BIASOTTI S.: Topological coding of surfaces with boundary using Reeb graphs. *Computer Graphics and Geometry 7*, 1 (2005), 31–45. 2

[BK05] BORGWARDT K. M., KRIEGEL H.-P.: Shortest-path kernels on graphs. In *Proc. 5th Int. Conf. on Data Mining* (Washington, DC, USA, 2005), IEEE Computer Society, pp. 74–81. 3

[BKS*05] BUSTOS B., KEIM D., SAUPE D., SCHRECK T., VRANIĆ D.: Feature-based similarity search in 3D object databases. *ACM Computing Surveys 37*, 4 (2005), 345–387. 2

[BSW08] BELKIN M., SUN J., WANG Y.: Discrete Laplace operator for meshed surfaces. In *Symposium on Computational Geometry* (Univ. of Maryland, USA, 2008), ACM, pp. 278–287. 4

[FMK*03] FUNKHOUSER T., MIN P., KAZHDAN M., CHEN J., HALDERMAN A., DOBKIN D., JACOBS D.: A Search Engine for 3D Models. *ACM Trans. Graphics 22*, 1 (2003), 83–105. 3

[FS95] FREUND Y., SCHAPIRE R. E.: A decision-theoretic generalization of on-line learning and an application to boosting. In *2nd European Conf. on Comp. Learning Theory* (London, UK, 1995), Springer-Verlag, pp. 23–37. 2

[FSH11] FISHER M., SAVVA M., HANRAHAN P.: Characterizing structural relationships in scenes using graph kernels. *ACM Transactions on Graphics 30* (Aug. 2011), 34:1–34:12. 3

[GA11] GÖNEN M., ALPAYDIN E.: Multiple kernel learning algorithms. *JMLR 12* (July 2011), 2211–2268. 2, 4

[GBP07] GIORGI D., BIASOTTI S., PARABOSCHI L.: *Watertight Models Track*. Tech. Report 09, IMATI, Genova, 2007. 2, 5, 6

[GFSF10] GIORGI D., FROSINI P., SPAGNUOLO M., FALCIDIENO B.: 3d relevance feedback via multilevel relevance judgements. *The Visual Computer 26*, 10 (2010), 1321–1338. 2

[GFW03] GÄRTNER T., FLACH P., WROBEL S.: On graph kernels: Hardness results and efficient alternatives. In *Proc. of the 16th Conf. on Comp. Learning Theory* (Washington, DC, USA, August 2003), vol. 2777 of *LNCS*, Springer, Berlin–Heidelberg, Germany, pp. 129–143. 3

[GM08] GIORGI D., MARINI S.: Shape retrieval contest 2008: Classification of watertight models. In *Shape Modeling International 2008* (June 4-6 2008), IEEE, pp. 219–220. 2, 5, 6

[HB04] HAASDONK B., BAHLMANN C.: Learning with distance substitution kernels. In *26th DAGM Symp. on Pattern Recognition* (Tübingen, Germany, August 2004), vol. 3175 of *LNCS*, Springer, pp. 220–227. 3

[HLR05] HOU S., LOU K., RAMANI K.: SVM-based semantic clustering and retrieval of a 3D model database. *Computer-Aided Design and Applications 2*, 1-4 (2005), 155–164. 2

[HSKK01] HILAGA M., SHINAGAWA Y., KOHMURA T., KUNII T. L.: Topology matching for fully automatic similarity estimation of 3D shapes. In *28th Conf. on Comp. Graphics and Interactive Techniques, SIGGRAPH'01* (Los Angeles, CA, August 2001), ACM Press, ACM Press, pp. 203–212. 4

[KJ03] KONDOR R., JEBARA T.: A kernel between sets of vectors. In *Int. Conf. on Machine Learning* (Washington, DC, USA, 2003), AAAI Press. 3

[KTI03] KASHIMA H., TSUDA K., INOKUCHI A.: Marginalized kernels between labeled graphs. In *20th Int. Conf. on Machine Learning* (Washington, DC, 2003), AAAI Press, pp. 321–328. 3

[Lag10] LAGA H.: Semantics-driven approach for automatic selection of best views of 3d shapes. In *3rd EG Conf. on 3D Object Retrieval* (Aire-la-Ville, Switzerland, 2010), Eurographics Association, pp. 15–22. 2

[LCB*04] LANCKRIET G. R. G., CRISTIANINI N., BARTLETT P., EL GHAOUI L., JORDAN M. I.: Learning the kernel matrix with semidefinite programming. *J. Mach. Learn. Res. 5* (2004), 27–72. 4

[MCHB07] MATEUS D., CUZZOLIN F., HORAUD R., BOYER E.: Articulated Shape Matching Using Locally Linear Embedding and Orthogonal Alignment. In *Work. Non-rigid Registration and Tracking through Learning* (Rio de Janeiro, Brésil, 2007), IEEE Computer Society Press, pp. 1–8. 2

[MPSF11] MARINI S., PATANÈ G., SPAGNUOLO M., FALCIDIENO B.: Spectral feature selection for shape characterization and classification. *The Visual Computer 27*, 11 (2011), 1005–1019. 2

[NWS07] NGUYEN G., WORRING M., SMEULDERS A.: Interactive search by direct manipulation of dissimilarity space. *Multimedia, IEEE Transactions on 9*, 7 (nov. 2007), 1404 –1415. 2

[OK06] OHBUCHI R., KOBAYASHI J.: Unsupervised learning from a corpus for shape-based 3d model retrieval. In *Proc. of the 8th ACM Int. Workshop on Multimedia Information Retrieval* (New York, NY, USA, 2006), MIR '06, ACM, pp. 163–172. 2

[PFJFG10] PHILIPP-FOLIGUET S., JORDAN M., FUZIER M., GOSSELIN P.-H.: Indexing of 3d models based on graph of surfacic regions. In *Proc. of the ACM workshop on 3D object retrieval* (New York, NY, USA, 2010), ACM, pp. 69–74. 3

[RBCG08] RAKOTOMAMONJY A., BACH F., CANU S., GRANDVALET Y.: Simplemkl. *J. of Machine Learning Research 9* (2008), 2491–2521. 4, 5

[Ree46] REEB G.: Sur les points singuliers d'une forme de Pfaff complètement intégrable ou d'une fonction numérique. *Comptes Rendus Hebdomadaires des Séances de l'Académie des Sciences 222* (1946), 847–849. 2

[SF06] SHILANE P., FUNKHOUSER T.: Selecting distinctive 3d shape descriptors for similarity retrieval. In *Shape Modeling International 2006* (Washington, DC, USA, 2006), SMI '06, IEEE Computer Society, pp. 18–. 2

[SRB07] SUARD F., RAKOTOMAMONJY A., BENRSHRAIR A.: Kernel on bag of paths for measuring similarity of shapes. In *15th Eu. Symp. on Artificial Neural Networks* (2007), pp. 355–360. 3

[SRSS06] SONNENBURG S., RÄTSCH G., SCHÄFER C., SCHÖLKOPF B.: Large Scale Multiple Kernel Learning. *Journal of Machine Learning Research 7* (July 2006), 1531–1565. 4

[TV04] TIEU K., VIOLA P.: Boosting image retrieval. *International Journal of Computer Vision 56* (2004), 17–36. 2, 4

[TV08] TANGELDER J. W., VELTKAMP R. C.: A survey of content based 3d shape retrieval methods. *Multimedia Tools and Applications 39* (September 2008), 441–471. 2

[VSKB10] VISHWANATHAN S. V. N., SCHRAUDOLPH N. N., KONDOR R., BORGWARDT K.: Graph kernels. *Journal of Machine Learning Research 11* (2010), 1201–1242. 3