The Visual Computer manuscript No.
(will be inserted by the editor)

Spectral Feature Selection for Shape Characterization and Classification

S. Marini · G. Patané · M. Spagnuolo · B. Falcidieno

Received: date/Accepted: date

Abstract This paper proposes a framework for selecting the Laplacian eigenvalues of 3D shapes that are more relevant for shape characterization and classification. We demonstrate the redundancy of the information coded by the shape spectrum and discuss the shape characterization capability of the selected eigenvalues. The feature selection methods used to demonstrate our claim are the AdaBoost algorithm and Support Vector Machine. The efficiency of the selection is shown by comparing the results of the selected eigenvalues on shape characterization and classification with those related to the first k eigenvalues, by varying k over the cardinality of the spectrum. Our experiments, which have been performed on 3D objects represented either as triangle meshes or point clouds, show that working directly with point clouds provides classification results that are comparable with respect to those related to surface-based representations. Finally, we discuss the stability of the computation of the Laplacian spectrum to matrix perturbations.

Keywords Shape characterization · feature selection · shape classification · point clouds · Laplacian spectrum

1 Introduction

Shape classification and retrieval are crucial tools in organizing and interacting with databases of 3D models and in getting a picture on the knowledge, or semantics, underlying the models. The performance of classification and retrieval strongly depends on the effectiveness of the shape descriptors, the comparison method, and the indexing techniques [11]. Even though several methods for shape comparison have been proposed [15, 52], only few methodologies address the issue of identifying descriptions that capture the shape features shared by a class of models [31, 36]. In this context, we propose a novel approach for shape characterization and classification based on the relevant information coded by the Laplacian spectrum of 3D shapes represented as point clouds.

The spectrum of the Laplace-Beltrami operator provides a descriptive and large feature vector, which characterizes the input shape and has been applied to shape matching due to its isometry-invariance, robustness to local noise and sampling, shape-intrinsic definition, and multi-scale organization. The first use of the Laplacian spectrum for shape matching was proposed in [48], where two shapes are compared by measuring the Euclidean distance between the vectors defined by the first 50 eigenvalues with smaller magnitude.

While there is an evidence of the close relation among shape features and eigenvalues, the best way to use the spectrum for shape characterization has not been identified yet [47]. Inspired by the earlier work presented in [30, 54], the idea behind the proposed approach is to consider the Laplacian eigenvalues, of either triangle meshes or point clouds, as shape descriptors and identify the most relevant information coded in the spectrum, capable of best characterizing the most relevant features of a given class of 3D objects. This information is then used for classification.

These goals are achieved by exploiting the properties of the feature selection and the classification capabilities of the AdaBoost algorithm [20] and Support Vector Machine [12, 23]. Then, the selected features are assessed with appropriate Bootstrap and Cross-Validation techniques. The result of the study is a new approach capable of automatically associating to classes of 3D objects the subset of the spectrum that

CNR-IMATI
Via De Marini 6
Genova, Italy
Tel.: +39-010-6475691
Fax: +39-010-6475691
E-mail: {marini,patane,spagnuolo,falcidieno}@ge.imati.cnr.it

is more relevant to characterize the intra-class similarity and discriminate among different classes.

Even if AdaBoost and Support Vector Machines are well-known algorithms for feature selection, our work is the first attempt at the identification of those sub-parts of the shape spectrum that discriminate among different classes of models. More precisely, our final aim is to select a subset of eigenvalues, which represents each class by means of those features that characterize the class members and that are discriminative with respect to non-member 3D objects.

In [35], we have shown that statistical methods are appropriate to correlate subsets of the spectrum to classes of 3D shapes and to have a grasp on the semantics captured by the eigenvalues. Starting from these results, we now verify that this statement also applies to point clouds and that the eigenvalues selected by the AdaBoost algorithm and Support Vector Machines effectively characterize the members of a given class of 3D objects. Furthermore, we prove that working directly with point clouds provides classification results that are comparable with respect to those related to surface-based representations.

Since the graph Laplacian of a point cloud converges to the Laplace-Beltrami operator of the underlying manifold, we compute the corresponding eigenvalues without applying the mesh Laplacian discretization based on cotangent and FEM weights. Our intuition is that feature selection based on the eigenvalues of the graph Laplacian of the input point cloud corresponds to feature selection based on the geometric structure of the underlying manifold. In this case, the computation of the Laplacian eigenvalues is based on manifold learning techniques [7] and is robust to the geometric/topological noise in the point cloud. For instance, the topological noise might be introduced when the local shape at a point is recovered using its k-nearest neighbor.

The paper is organized as follows. Sect. 2 briefly recalls previous work on shape comparison and Sect. 3 introduces the spectral analysis for surfaces. Then, Sect. 4 presents the proposed feature selection approach, the data set used for the experiments, the AdaBoost algorithm, Support Vector Machines, and the comparison with previous work. Sect. 5 discusses the characterization and the classification capabilities of the selected eigenvalues. Finally, Sect. 6 provides closing remarks on results and outlines future work.

2 Related work

In the following, we review previous work on feature selection, spectral and point-based shape descriptors.

Feature selection The AdaBoost algorithm [21], which is a statistical tool for feature extraction from 2D images [54], has been used [30] to select relevant views of 3D objects with respect to the light field descriptor [16]. Other classifiers based on semi-supervised learning, dimensionality reduction, and probability have been successfully exploited for shape classification. For instance, in [25] Support Vector Machine is used to cluster 3D models with respect to semantic information. In [26,42], shape classifiers are obtained as a linear combination of individual classifiers and using non-linear dimensionality reduction. In [51], relevant local shape descriptors are selected through a multivariate Gaussian distribution and collected to define a priority-driven search for shape retrieval.

Point sets and point-based shape descriptors Point-sampled surfaces, generated either by scanning real 3D objects with optical devices or sampling implicit and parametric functions, are discrete models of surfaces with an arbitrary genus and a generally high number of points. Point sets became an alternative to polygonal meshes, due to the simplicity of dealing with complex 3D shapes as point clouds and using points as rendering primitives [28,46,49,57]. The lack of connectivity and the atomic definition of point clouds provide a built-in multi-scale surface representation [45], thus avoiding to process the connectivity of polygonal meshes.

Point sets are widely used for ray tracing [2], surface reconstruction [39,56], sampling [3], simplification [44], segmentation [6], spectral analysis [41], machine learning [9, 53], progressive rendering and streaming [19]. In this context, only few methodologies address the comparison of point clouds, without explicitly computing the underlying mesh connectivity. For instance, in [34] point clouds are compared using the histograms of pairwise diffusion, geodesic, and curvature weighted distances. Even though other comparison methods based on mesh representations of 3D shapes might be extended to point clouds, a low attention has been payed to this problem.

In [50], the eigenvectors corresponding to the Laplacian eigenvalues of smaller magnitude are used to define a shape representation that is invariant to isometric transformations. Then, these signatures are compared using a modification of the $D2$-distribution [43], which is based on a set of histograms that capture the variation of distances among points within a set of spherical cells centered at the origin of a k-dimensional space. In [38], *Local Linear Embeddings* are constructed on eigenspaces of affinity matrices and matched by using the *Expectation-Maximization* framework. Instead of using the spectrum itself, in [27] non-rigid objects are matched using spectral embeddings, which are derived from the eigenvectors of affinity matrices computed considering geodesic distances.

3 Spectral descriptors of 3D shapes

Given an input surface $\mathscr{M}$, the *Laplacian spectrum* of $\mathscr{M}$ is defined as the set of solutions (λ, f) of the following *eigenvalue problem*:

$$\text{find } f : \mathscr{M} \to \mathbb{R} \text{ such that } \Delta f = \lambda f, \quad \lambda \in \mathbb{R},$$

where Δ is the Laplace-Beltrami operator. The intuition behind spectral shape comparison is that the Laplacian eigenvalues are meaningful for the description of the input surface $\mathscr{M}$ due to their intrinsic definition and invariance with respect to isometric transformations.

In the discrete setting, we approximate the surface $\mathscr{M}$ with a point cloud $\mathscr{P}$ and the Laplace-Beltrami operator with a sparse matrix. More precisely, a point-based representation of a surface $\mathscr{M}$ is a finite set $\mathscr{P}$ of points sampled on $\mathscr{M}$. The surface $\mathscr{M}$ underlying $\mathscr{P}$ is commonly estimated using the *moving least-squares* [3,4,33] and the *implicit* [1] *approximations*. Given a point set $\mathscr{P} := \{\mathbf{p}_i\}_{i=1}^n$, in the *k-nearest neighbor graph* $\mathscr{G}$ of $\mathscr{P}$ each point $\mathbf{p}_i \in \mathscr{P}$ is associated with its k nearest points in $\mathscr{P}$, which constitute the *neighbour* $\mathscr{N}_{\mathbf{p}_i} := \{\mathbf{p}_{j_s}\}_{s=1}^k$ of $\mathbf{p}_i$. For dense point sets, this graph provides sufficient information to approximate the local geometric and topological structure of $\mathscr{M}$ without meshing the whole point set. The computation of $\mathscr{G}$ requires $O(n \log n)$ time [5].

According to [7,8,18], on the point cloud $\mathscr{P} := \{\mathbf{p}_i\}_{i=1}^n$ the *normalized graph Laplacian matrix* $L_{nr} := (L_{nr}(i,j))_{i,j=1}^n$ is defined as

$$L_{nr}(i,j) := \begin{cases} 1 & i = j, \\ -W(i,j)/\alpha_i & \mathbf{p}_j \in \mathscr{N}_{\mathbf{p}_i}, \\ 0 & \text{else}, \end{cases} \qquad (1)$$
$$\begin{cases} W(i,j) := \exp\left(-\frac{\|\mathbf{p}_i - \mathbf{p}_j\|_2^2}{h^2}\right), \\ \alpha_i := \sum_{j \in \mathscr{N}_{\mathbf{p}_i}} W(i,j). \end{cases}$$

Starting from this discretization and following the graph theory terminology, we introduce different normalizations of the graph Laplacian matrix by rewriting the normalized graph Laplacian matrix as $L_{nr} = D^{-1}(D - W) = D^{-1}L_{un}$, where $L_{un} := D - W$ is the *un-normalized graph Laplacian matrix* and W is the weight matrix, whose elements are $W(i,j)$ if (i,j) is an edge of $\mathscr{G}$ and zero otherwise. As detailed in Sect. 3.1, the symmetry of L_{un} is important to guarantee the stable computation of its eigenvalues with respect to perturbations of the Laplacian matrix and the input surface. Finally, we recall that the matrices L_{nr} and L_{un} are positive semidefinite.

To reduce the dependency of the Laplacian eigenmaps representation from the density of the data points, in [29] the Gaussian weights are normalized with an estimate of the point density and the Laplacian matrix is updated with these new weights. Therefore, the new Laplacian matrix that replaces (1) is built in two phases as follows

$$\tilde{L}(i,j) := \begin{cases} \frac{W(i,j)}{\alpha_i \alpha_j} & \mathbf{p}_j \in \mathscr{N}_{\mathbf{p}_i}, \\ 0 & \text{else}, \end{cases}$$

$$L(i,j) := \begin{cases} -1 & i = j, \\ \frac{\tilde{L}(i,j)}{\sum_{k \in \mathscr{N}_{\mathbf{p}_i}} \tilde{L}(i,k)} & \mathbf{p}_i \in \mathscr{N}_{\mathbf{p}_i}, \\ 0 & \text{else}. \end{cases}$$

In this case, in the limit of large sampled points and small scales the eigenvectors of the new Laplacian matrix converge to those of the Laplace-Beltrami operator on $\mathscr{P}$. An alternative discretization of the Laplacian matrix is described in [32]. We remind that the vector $\mathbf{h}$, $\mathbf{h} \neq \mathbf{0}$, is an *eigenvector* of L related to the *eigenvalue* λ if and only if $L\mathbf{h} = \lambda\mathbf{h}$. Finally, we assume that the eigenvalues $(\lambda_i)_{i=1}^n$ have been increasingly reordered.

3.1 Eigenvalue localization and perturbation analysis

In the following, we provide an estimation of the localization of the eigenvalues with respect to the different normalizations; some of them generalize well-known bounds derived in the context of graph theory [17,40]. Then, we focus on the stability of their computation.

Eigenvalue localization To bound the eigenvalues of the Laplacian matrix, we remind that for any matrix A the following relation holds

$$\begin{aligned} &\max\{|\,\lambda\,|, \lambda \in \text{spectrum}(A)\} \\ &\leq \min\left\{ \max_{i=1,\ldots,n}\left\{\sum_{j=1}^n |\,A(i,j)\,|\right\}, \max_{j=1,\ldots,n}\left\{\sum_{i=1}^n |\,A(i,j)\,|\right\}\right\}. \end{aligned}$$

Therefore, applying the last relation to the normalized graph Laplacian we get that

$$\begin{aligned} &\max_{i=1,\ldots,n}\left\{\sum_{j=1}^n |\,L_{nr}(i,j)\,|\right\} \\ &= \max_{i=1,\ldots,n}\left\{|\,L_{nr}(i,i)\,| + \sum_{j\neq i} |\,L_{nr}(i,j)\,|\right\} \\ &= 2. \end{aligned}$$

In a similar way, we derive the following bound for the unnormalized graph Laplacian

$$\begin{aligned} \lambda_{n-1}(L_{un}) &= \max_{i=1,\ldots,n}\left\{|\,d(i)\,| + \sum_{j\neq i} |\,W(i,j)\,|\right\} \\ &= 2 \max_{i=1,\ldots,n}\{d(i)\}, \end{aligned}$$

where $d(i) := \sum_{j\neq i} |W(i,j)|$.

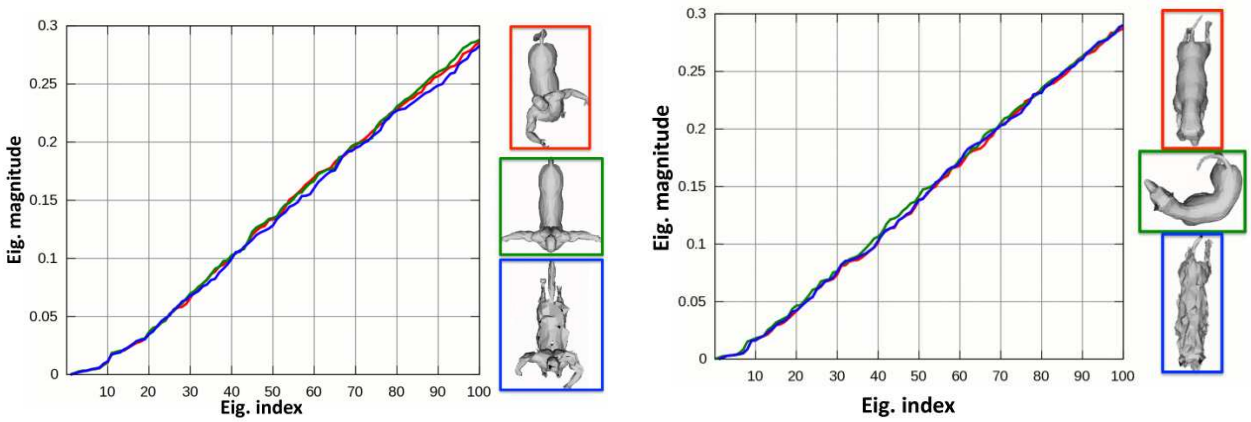


Fig. 1 Variation (y-axis) of the first 100 Laplacian eigenvalues (x-axis) of the un-normalized graph Laplacian of point clouds belonging to different classes of the SHREC 2010 data set. Deformed shapes with a different sampling density share an analogous variation of the corresponding spectra.

Perturbation analysis In the following, we identify the properties of the Laplacian matrix that mainly influence the computation of the spectrum and guarantee a low conditioning number of the corresponding eigenvalues. Assuming that the sampling density of the input surface is coherent with the shape details that must discriminate similar shapes, our experiments have shown that the spectrum of the graph Laplacian is not strongly affected by the noise and shape sampling. For instance, in Figure 1 resampled and almost isometrically deformed surfaces have an almost identical spectrum.

To this end, we focus our attention on the stability of the eigenvalue computation with respect to perturbations of the Laplacian matrix by distinguishing between symmetric/unsymmetric structures and single/multiple eigenvalues. More precisely, we show that each eigenvalue of a symmetric Laplacian matrix L is always well-conditioned. If L is un-symmetric (e.g., the un-normalized graph Laplacian matrix), then we have experimentally verified that the un-symmetric structure does not necessarily imply the eigenvalue sensibility. If L has an eigenvalue λ of multiplicity m, $m \geq 2$, then a perturbation ε to L might produce a change of order ε^m in λ and this amplification becomes more and more evident while increasing the multiplicity of the eigenvalue.

Since any self-adjoint matrix L can be expressed as a linear combination $L = \sum_{i=1}^{n} \lambda_i \mathbf{x}_i \mathbf{x}_i^T$ of its eigenvalues and eigenvectors $\{(\lambda_i, \mathbf{x}_i)\}_{i=1}^{n}$, perturbations of L are strictly related to changes of its spectrum. Given an arbitrary matrix $L \in \mathbb{R}^{n \times n}$ and indicating with L^T its transpose, we recall that the nonzero vectors $\mathbf{x} \in \mathbb{R}^n$ that satisfy $L\mathbf{x} = \lambda \mathbf{x}$ and $\mathbf{x}^T L = \lambda \mathbf{x}^T$ are defined respectively as the *right* and *left eigenvectors* of L related to the eigenvalue λ. It follows that the left eigenvector of L related to the eigenvalue λ is the right eigenvector of L^T with respect to λ.

Chosen any matrix E, we perturb the matrix L by εE, with $\varepsilon \to 0$, and compute the eigenpair $(\lambda(\varepsilon), \mathbf{x}(\varepsilon))$ of the new problem

$$(L + \varepsilon E)\mathbf{x}(\varepsilon) = \lambda(\varepsilon)\mathbf{x}(\varepsilon), \quad \mathbf{x}(0) = \mathbf{x}, \quad \lambda(0) = \lambda. \tag{2}$$

The size of the derivative of $\lambda(\varepsilon)$ indicates the variation that $\lambda(\varepsilon)$ undergoes when the matrix L is perturbed in the direction (E, ε). By differentiating (2), we obtain

$$(L + \varepsilon E)\mathbf{x}'(\varepsilon) + E\mathbf{x}(\varepsilon) = \lambda'(\varepsilon)\mathbf{x}(\varepsilon) + \lambda(\varepsilon)\mathbf{x}'(\varepsilon),$$

and therefore $L\mathbf{x}'(0) + E\mathbf{x} = \lambda'(0)\mathbf{x} + \lambda\mathbf{x}'(0)$. Let us take the inner product of both members of this last equation with the left eigenvector $\mathbf{y}$ associated with λ (i.e., $\mathbf{y}^T L = \lambda \mathbf{y}^T$); then, we get

$$\lambda'(0) = \frac{\mathbf{y}^T E \mathbf{x}}{\mathbf{y}^T \mathbf{x}}, \text{ and } |\lambda'(0)| \leq \|E\|_2 \frac{\|\mathbf{x}\|_2 \|\mathbf{y}\|_2}{|\langle \mathbf{x}, \mathbf{y} \rangle_2|} = \frac{\|E\|_2}{\alpha(\lambda)}, \tag{3}$$

with $\alpha(\lambda) := \frac{|\langle \mathbf{x}, \mathbf{y} \rangle_2|}{\|\mathbf{x}\|_2 \|\mathbf{y}\|_2} \leq 1$. Note that the previous expression is defined if λ is a simple eigenvalue; in fact, under this assumption the left and right eigenvectors cannot be orthogonal. Then, the *conditioning number* of the simple eigenvalue λ of L is defined as

$$\text{cond}(\lambda) := \frac{\|\mathbf{x}\|_2 \|\mathbf{y}\|_2}{|\langle \mathbf{x}, \mathbf{y} \rangle_2|} = \frac{1}{\alpha(\lambda)} \geq 1, \tag{4}$$

and it is independent of the normalization of the left and right eigenvectors. If X is the matrix whose columns are the right eigenvectors of L, then the rows of $Y^T := X^{-1}$ are the left eigenvectors of L and $Y^T X = I$ with I identity matrix. It follows that $\langle \mathbf{x}, \mathbf{y} \rangle_2 = 1$ and

$$\begin{aligned} \text{cond}(\lambda) &= \|\mathbf{x}\|_2 \|\mathbf{y}\|_2 \\ &\leq \|X\|_2 \|Y^T\|_2 \\ &\leq \|X\|_2 \|X^{-1}\|_2 \\ &= \kappa_2(X); \end{aligned}$$

i.e., the conditioning number $\kappa_2(X)$ of the eigenvector matrix X is an upper bound for the conditioning number of each

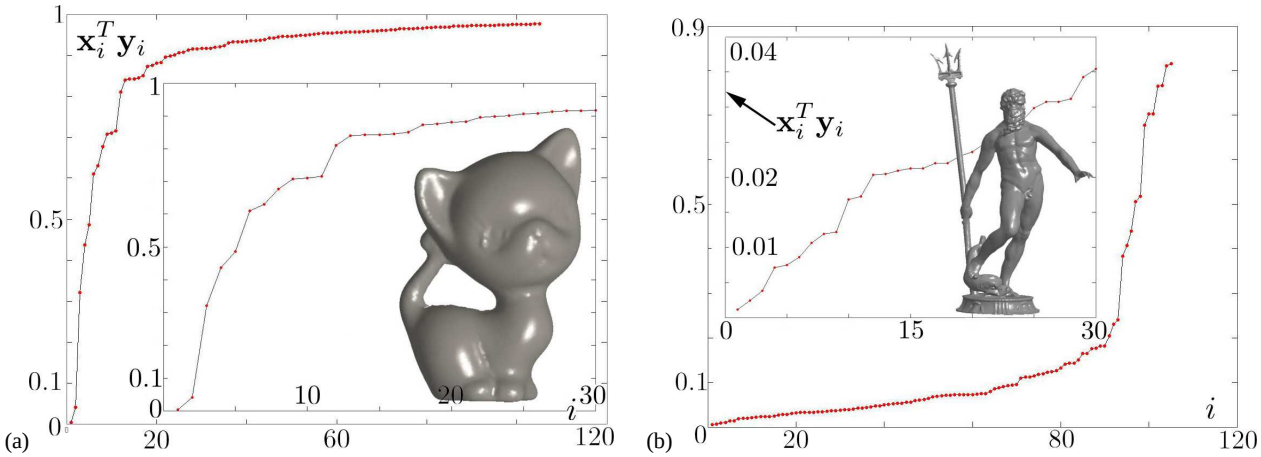


Fig. 2 Plot of $\alpha(\lambda_i) := |\,\mathbf{x}_i^T\mathbf{y}_i\,|$, where $\mathbf{x}_i$ and $\mathbf{y}_i$ are the left and right eigenvector related to the same eigenvalue λ_i of the normalized graph Laplacian; in this test, we considered the first 200 eigenfunctions and each picture shows a zoom-in on the scalar product of the first 30 eigenvectors. In (a) and (b), the minimum of $\{\alpha(\lambda_i)\}_i$ is 0.0036 and 0.0012, respectively. From this example, we see that the un-symmetric structure of the Laplacian matrix does not imply the eigenvalue sensibility.

eigenvalue. In particular, if X is an orthogonal matrix (e.g., L is symmetric) then $\text{cond}(\lambda) = 1$, $\forall\lambda \in \text{spectrum}(L)$. Therefore, each eigenvalue of the un-normalized graph Laplacian matrix is always well-conditioned. Under the assumption that

$$\|\mathbf{x}\|_2 = \|\mathbf{y}\|_2 = 1 \text{ and } \|E\|_2 = 1,$$

we get $|\lambda'(0)| \leq \frac{1}{|\mathbf{y}^T\mathbf{x}|}$ and the upper bound is attained with $E := \mathbf{y}^T\mathbf{x}$. Rewriting the relation (3) as

$$(\lambda(\varepsilon) - \lambda) \approx \varepsilon\|E\|_2\text{cond}(\lambda), \quad \varepsilon \to 0,$$

shows that a perturbation of order ε in L produces an amount $\varepsilon\|E\|_2\text{cond}(\lambda)$ of variation in λ. Therefore, if $\text{cond}(\lambda)$ is large or equal to one then λ is ill- or well- conditioned, respectively.

Since the normalized graph Laplacian is not symmetric, we compute the left and right eigenvectors of L associated with λ and estimate its conditioning according to (4). Figure 2 shows the variation of the conditioning number of the first 200 eigenvalues of the un-symmetric Laplacian matrix with mean-value weights. From this example, it follows that an un-symmetric structure of L does not necessarily imply the eigenvalue sensibility. Since the un-normalized graph Laplacian is symmetric, the left and right eigenvectors are the same and $\text{cond}(\lambda) = 1$; indeed, the computation of the eigenvalues is always stable.

Multiple Laplacian eigenvalues If λ_k is an eigenvalue of L with multiplicity m_k, then we are not guaranteed that its left and right eigenvectors are not orthogonal and the previous discussion does not apply. In this case, the characteristic polynomial p_L of L is written as

$$p_L(\lambda) := \det(L - \lambda I) = (\lambda - \lambda_k)^{m_k} q(\lambda), \quad q(\lambda_k) \neq 0,$$

with q polynomial of degree $n - m_k$. Then, a perturbation in L of size ε results in a change of $p_L(\lambda)$ of order $O(\varepsilon)$ and

$$(\lambda - \lambda_k)^{m_k} = \frac{O(\varepsilon)}{q(\lambda)} \quad \leftrightarrow \quad \lambda = \lambda_k + O(\varepsilon^{\frac{1}{m_k}}).$$

This relation implies that a perturbation $\varepsilon := 10^{-m_k}$ produces a change of order 0.1 in λ_k and this amplification becomes more and more evident while increasing the multiplicity of the eigenvalue. In this case, the eigenspace $\mathscr{F}_{\lambda_k}$ associated with λ_k has multiple dimension and different eigenvectors can be selected as basis of $\mathscr{F}_{\lambda_k}$.

Normalization of the Laplacian spectrum for shape comparison For triangulated surfaces, the FEM Laplacian eigenvalues become invariant to shape scales by normalizing the spectrum of the input shape with its area. For point clouds, rescaling the points of $\mathscr{P}$ by a factor α the entries of the un-normalized graph Laplacian matrix $L_{un}^{t,\mathscr{P}}$ of the new point cloud $\mathscr{Q} := \{\alpha\mathbf{p}_i\}_{i=1}^n$ satisfy the relation $L_{un}^{t,\mathscr{Q}} = L_{un}^{t/\alpha^2,\mathscr{P}}$; indeed, the time component is rescaled from t to t/α^2. A similar discussion applies to the normalized graph Laplacian matrix. Therefore, we guarantee the invariance of the Laplacian spectrum with respect to shape rescaling by normalizing each point cloud before computing its Laplacian eigenvalues.

To measure the variation of the eigenvalues between the reference point cloud $\mathscr{P}$ and its deformation $\mathscr{P}'$, let us indicate with $\lambda := (\lambda_i)_{i=1}^k$ and $\lambda' := (\lambda_i')_{i=1}^k$ the vector of the first k eigenvalues of $\mathscr{P}$ and $\mathscr{P}'$, respectively. Then, we measure the normalized error between these spectra as

$$E_\infty := \|\lambda - \lambda'\|_\infty / \|\lambda\|_\infty.$$

Even though the Laplacian spectrum characterizes geometric and topological features of 3D shapes in a way that is

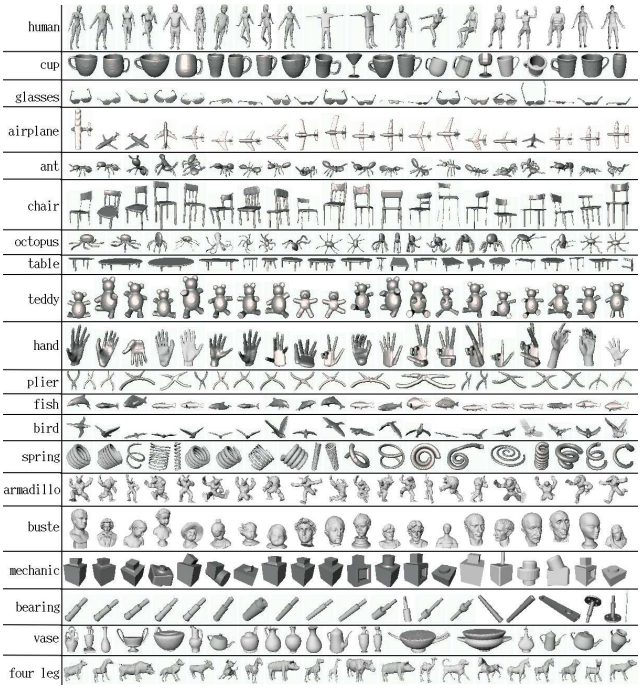


Fig. 3 SHREC 2007 data set.

not unique, previous work [48,55] has also shown that the spectrum is capable of distinguishing dissimilar shapes.

4 Feature selection for shape characterization and classification

In the literature, few works tackle the problem of similarity by using the shape spectrum. Starting from the good retrieval results in [48], we investigate whether the selection of a particular set of Laplacian eigenvalues is capable of characterizing a specific class of shapes and improve the classification performance. Furthermore, the prohibitive cost needed to compute the complete spectrum and the redundancy of the extracted information stress the importance of identifying relevant information from the shape spectrum. Indeed, our work investigates the problem of selecting a bunch of eigenvalues that characterize the members of a given class of shapes and that maximize the classification performance among several classes of shapes. To this end, we investigate the utility of the AdaBoost algorithm (Sect. 4.1) and the Support Vector Machine technique (Sect. 4.2) as methods for spectral feature selection and classification. The shape characterization capabilities of the selected features are shown through the comparison with the traditional approach based on the first k eigenvalues of smaller magnitude, by varying k on the cardinality of the computed spectrum (Sect. 4.3).

In general, feature selection addresses the problem of finding the most compact and informative set of features, thus improving the efficiency or data storage and processing [23]. In our approach, eigenvalues are considered relevant whether they can be used to discriminate among the 3D models of a given class and other shapes that do not belong to such a class. In this way, the selection of the relevant eigenvalues is transformed into a binary classification problem as follows. Each model of a given class is represented by a subset of eigenvalues selected from the corresponding spectrum. Then, a binary classification function, based on the selected eigenvalues, is defined such that it returns "*true*" if an unknown query model is attributed to the given class and "$false$" otherwise. The selected eigenvalues represent the shape features that are shared by the class members and maximize the discrimination with respect to the non-member models.

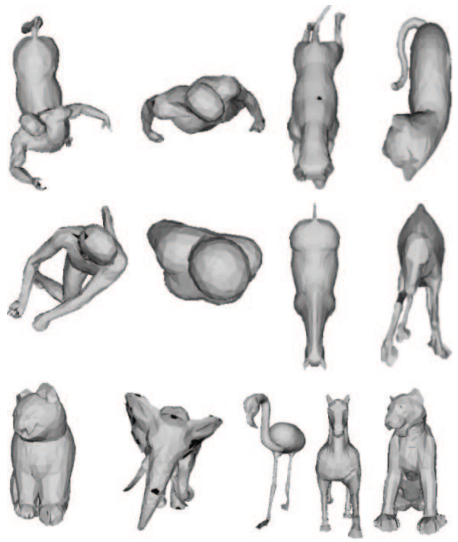


Fig. 4 Representatives of the 13 classes of the SHREC 2010 data set.

Experiments have been performed by using two data sets from the SHape Retrieval Contest[1]: the SHREC 2007 [22, 37] and the SHREC 2010 [13,14] data sets. The SHREC 2007 data set (Figure 3) contains 400 watertight triangle meshes grouped into 20 classes, with 20 models per class. The main differences among shapes of different classes are localized at coarse features; e.g., cups, chairs, or springs. Other classes share similar coarse features and differ just for details; e.g., humans, teddy bears, and armadillos. In this case, we have considered both polygonal meshes and point clouds uniformly sampled from the model surfaces.

The SHREC 2010 data set contains 728 models, which are represented as triangle meshes and are grouped into 13 classes (Figure 4). Note that these meshes are not necessarily watertight or manifold. Members of each class are obtained through transformations of different strength (five degrees) of the class representative model: the higher the de-

[1] `http://www.aimatshape.net/event/SHREC/`

Fig. 5 Transformations of the human shape used in the tests (shown in strength 5, left to right): null, isometry, topology, sampling, local scale, scale, holes, micro holes, noise, shot noise. Image extracted from [13,14].

gree, the stronger the transformation. The transformations are (Figure 5): null transformation, isometry (non-rigid and almost inelastic deformations), topology (welding of shape vertices resulting in different triangulation), micro holes and big holes, global and local scaling, additive Gaussian noise, shot noise, down-sampling (less than 25% of the original points), partial occlusion, and mixed transformation. Differently from SHREC 2007, the classes of the SHREC 2010 data set are strongly overlapped. Each shape of the SHREC 2010 data set has been represented as a point cloud; then, the eigenvalues of the corresponding un-normalized graph Lapalcian matrix (Sect. 3) have been computed.

4.1 The AdaBoost algorithm

The AdaBoost algorithm [20] is a supervised machine learning method for binary classification. It is based on a set of positive and negative examples and exploits a set of weak classifiers to generate a binary classification function (strong classifier) that maximizes the margin between positive and negative examples. The weak classifiers are simple classification functions not necessarily accurate and stable, but efficiently computable [20]. The algorithm iteratively selects the more appropriate weak classifiers and generates a classification function based on the combination of the selected classifiers. This function is capable of classifying an unknown query as belonging to the class of positive or negative examples and it is computed by minimizing the classification error.

In the proposed approach, each shape $\mathscr{P}$, which is represented as a triangle mesh or a point cloud, is coded by its Laplacian spectrum and a weak classifier is defined for each eigenvalue. In this context, the AdaBoost algorithm selects the eigenvalues whose weak classifiers maximize the margin between positive and negative examples. Furthermore, the selected eigenvalues are combined to generate an effective strong classifier.

For shape classification, each class is used in turn as positive example and a subset of the remaining models is considered as negative example (Sect. 5). In this way, we obtain a set of relevant eigenvalues, which characterizes each class of the data set. Among the shape features shared by the class members, we consider the selected eigenvalues as the class descriptions that maximize the distance among models of different classes.

In our experiments, the weak classifier h_k classifies the shape $\mathscr{P}$ by using the kth eigenvalues of its spectrum and it is defined as

$$h_k(\mathscr{P}) = \begin{cases} 1 & \max_{\mathscr{R}\in E^+} d_k(\mathscr{P},\mathscr{R}) \leq \delta, \\ -1 & \text{otherwise,} \end{cases}$$

where E^+ is the set of positive examples, $d_k(\mathscr{P},\mathscr{R})$ is the absolute value of the difference between the kth eigenvalue of the models $\mathscr{P}$ and $\mathscr{R}$, and $\delta := \max_{\mathscr{R},\mathscr{Q}\in E^+} d_k(\mathscr{R},\mathscr{Q})$, is a real number that is associated with the set of positive examples and represents the maximum distance between $\mathscr{R}$ and $\mathscr{Q}$. In the following, we outline the AdaBoost algorithm used by the proposed approach.

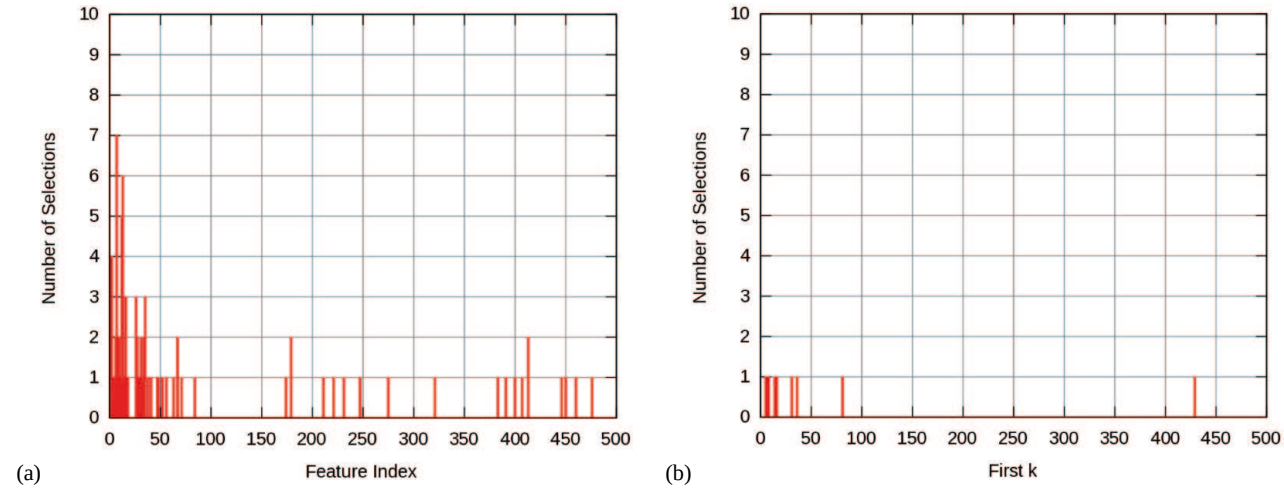
- **Input examples**: $(\mathscr{P}_1,y_1),\ldots,(\mathscr{P}_m,y_m)$, where $\mathscr{P}_i \in D$ and $y_i \in Y = \{+1,-1\}$, $i = 1,\ldots,m$.
- **Initialization**:

  $$w_{0,i} := \frac{1}{2|E^+|} \text{ if } \mathscr{P} \in E^+, \quad w_{0,i} := \frac{1}{2|E^-|} \text{ otherwise,}$$

  where $|E^+|$ and $|E^-|$ are the number of positive and negative examples, respectively.
- **Iteration**: for $t = 1,\ldots,T$,
  - train the weak classifiers by using the weights $w_{t,i}$;
  - select the weak classifier h_t producing the lowest classification error ε_t.
  - **Update of the weights**: $w_{t+1,i} = \frac{1}{Z_t} w_{t,i} e^{-\alpha_t h_t(\mathscr{P})y_i}$, $\alpha_t = \frac{1}{2}\log\frac{1-\varepsilon_t}{\varepsilon_t}$ and Z_t is a normalization factor such that $w_{t,i}$ ranges in $[0,1]$.
- **Strong classifier**:

  $$S = Sign\left(\sum_{t=1}^{T} \alpha_t h_t\right). \tag{5}$$

The algorithm takes as input the set of positive and negative examples, $(\mathscr{P}_1,y_1),\ldots,(\mathscr{P}_m,y_m)$, where $\mathscr{P}_i$ is a model of the considered data set and $y_i \in \{+1,-1\}$ is the label representing a positive and negative example, respectively. AdaBoost iteratively trains the weak classifiers associated with



(a) (b)

Fig. 6 Eigenvalues of the Laplacian spectrum selected through 10 Bootstrap iterations using (a) the AdaBoost algorithm and (b) the first Laplacian eigenvalues. The $y-$axis of (a,b) represents the number of times the eigenvalues have been selected. The $x-$axis in (a,b) represents the eigenvalue indices and the value of k, respectively.

the eigenvalues repeatedly in T iterations. During these iterations, the algorithm maintains a set of weights over the positive and negative examples. In particular, $w_{t,i}$ represents the weight of the example $\mathscr{P}_i$ at the iteration t. Once the weights have been initialized at the first step, they are iteratively updated on the basis of the incorrectly classified examples. In particular, at each iteration the weak classifier that produces the minimum classification error is selected and the error is used to update the weights of the input examples. Then, the weights of misclassified examples are increased and the weights of the correctly classified examples are reduced. This strategy forces the algorithm to focus only on hard examples.

4.2 The Support Vector Machine

Besides AdaBoost, Support Vector Machines (SVMs) [12, 23] are effective approaches for pattern classification and feature selection. Basically, SVMs belong to the category of kernel methods, which generate nonlinear decision boundaries among classes of patterns through methods designed for linear classifiers. SVMs can also be exploited for feature selection through the Recursive Feature Elimination (RFE) algorithm [10,23]. This method iteratively removes features that correspond to components of the SVM weight vector that are smallest in absolute value; such features provide a lower contribution to the classification and are therefore removed [24]. The RFE method involves three main steps: (i) train the classifier; (ii) compute the ranking criterion for all features; (iii) remove the feature with smallest ranking criterion.

The main difference between SVMs and AdaBoost is that the former relies on the definition of the most appropriate kernel to maximize the margin, while the latter achieves analogous results by using a fast greedy algorithm based on weak classifiers. Moreover, SVMs require to solve a quadratic programming problem, while the AdaBoost algorithm is based on linear programming. Finally, the experiments have been performed using the Python toolbox PyML[2].

4.3 The first k eigenvalues approach

In [48], the first k eigenvalues with smaller magnitude are used as shape descriptor to compute 3D shapes represented as triangle meshes. A typical classification scheme that exploits the first k eigenvalues is based on the distance

$$\tilde{d_s}(q,C) = \min_{m \in C} d_s(q,m), \tag{6}$$

between the query q and the class C, where s is the spectrum subsequence defined as $s := (1,\ldots,k) \subseteq (1,\ldots,k_{\max})$ with $k_{\max}$ the cardinality of the spectrum, and d_s is the distance between the query model q and the class C with respect to s. A general definition for the query-class classification scheme

$$q \mapsto \overline{C} \iff \overline{C} = \arg\min_{C \in D} \tilde{d_s}(q,C), \tag{7}$$

The average classification error $\varepsilon \in [0,1]$ is computed as the number of wrongly classified queries divided by the total number of queries. Based on Eq. (6) and Eq. (7), the first k approach selects the spectrum subsequence s that minimizes the classification error ε.

[2] `http://sourceforge.net/projects/pyml/`

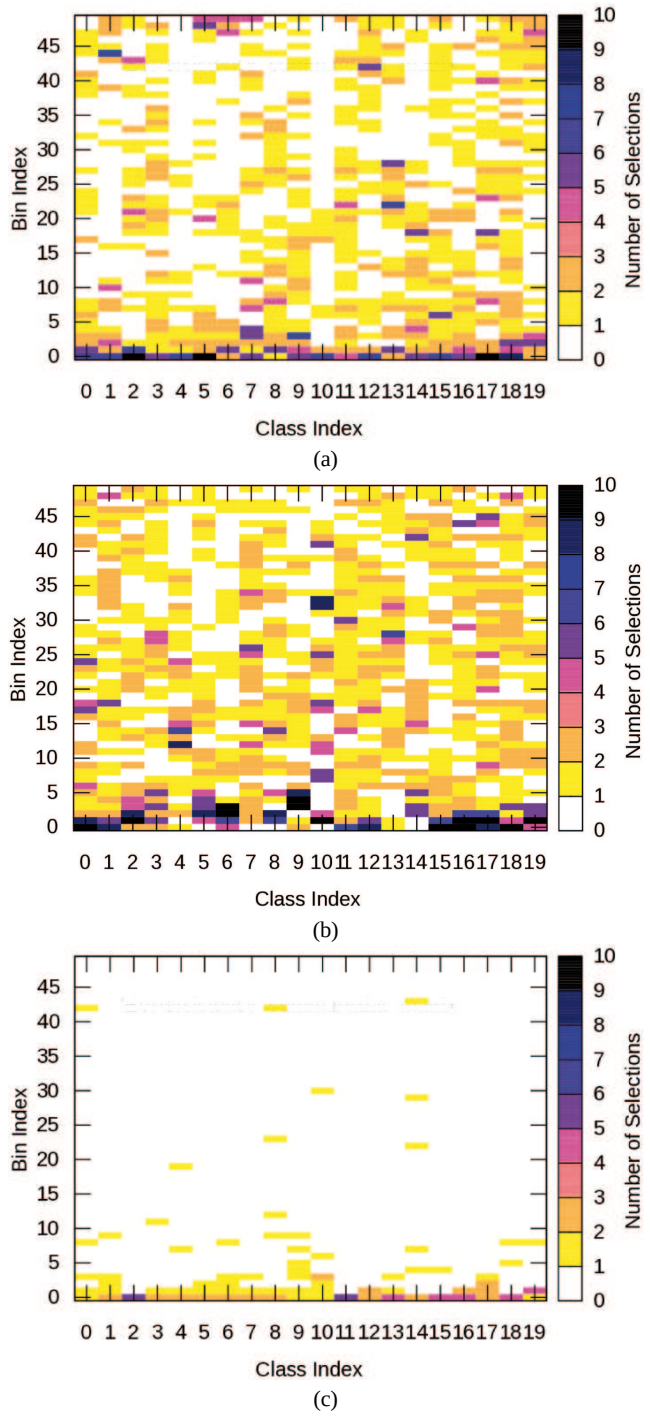


Fig. 7 Selected eigenvalues using (a) the AdaBoost algorithm, (b) the Support Vector Machine, and (c) the first k eigenvalues. Each column of the heat-map matrix represents a SHREC 2007 class. The color of each matrix element represents the number of times the eigenvalues have been selected within 10 iterations of Bootstrap Cross-Validation. The color bar provides the map between colors and number of selections. The $x-$ and $y-$axis represent the class index and the spectrum bin, respectively.

5 Shape characterization and classification

The most relevant features that characterize a class of shapes can be captured by a properly selected subset of their Laplacian eigenvalues. As discussed in Sect. 4, a binary classifier based on a set of eigenvalues can be used to discriminate among the members of the given class and the remaining shapes of the input data set. The set of eigenvalues that maximizes the margin between class members and non-members represents the features shared by the class models. To identify and validate the most representative subset of the Laplacian spectrum, the Bootstrap Cross-Validation assessment methodology has been applied by splitting the data set into training and validation set. The training set is used to select the relevant features and the validation set is used to assess the classification performance.

The Bootstrap methodology is a general random resampling with replacement and it has been used to generate different training and validation sets for Cross-Validation. In our experiments on shape characterization (Sect. 5.1) and classification (Sect. 5.2), the training set is obtained through the Bootstrap strategy by randomly sampling a subset of shapes of a given class and the same number of models among the remaining models.

5.1 Shape characterization

The eigenvalues that characterize the shape features of a given class are obtained by applying the AdaBoost strong classifier in Eq. (5) or the first s classifier described in Eq. (7), which are trained on the positive and negative examples of the training set. The training process selects the set of relevant eigenvalues that maximize the margin between positive and negative examples. Different iterations of the Bootstrap method produce different training sets used for selecting the eigenvalues. Each training set contains some representatives of a given class and very few representatives of the other classes; some classes might not be represented at all. In this way, the selected eigenvalues mainly depend on the class members. The eigenvalues that are selected several times within all the Bootstrap iterations are those that best characterize the class of shapes.

Figure 6(a) shows the eigenvalues selected by AdaBoost for the SHREC 2007 class of humans through 10 Bootstrap iterations. In this experiment, all the models have been uniformly remeshed to 10K vertices and for each shape we have computed the first 500 eigenvalues of FEM Laplacian matrix [48,55]. Since several eigenvalues are never selected (e.g., the eigenvalues with indices between 100 and 150), we conclude that they are less discriminative for the characterization of the class of humans. Those eigenvalues that have been selected just once (e.g., the eigenvalues with indices between 200 and 400) depend on the specific training

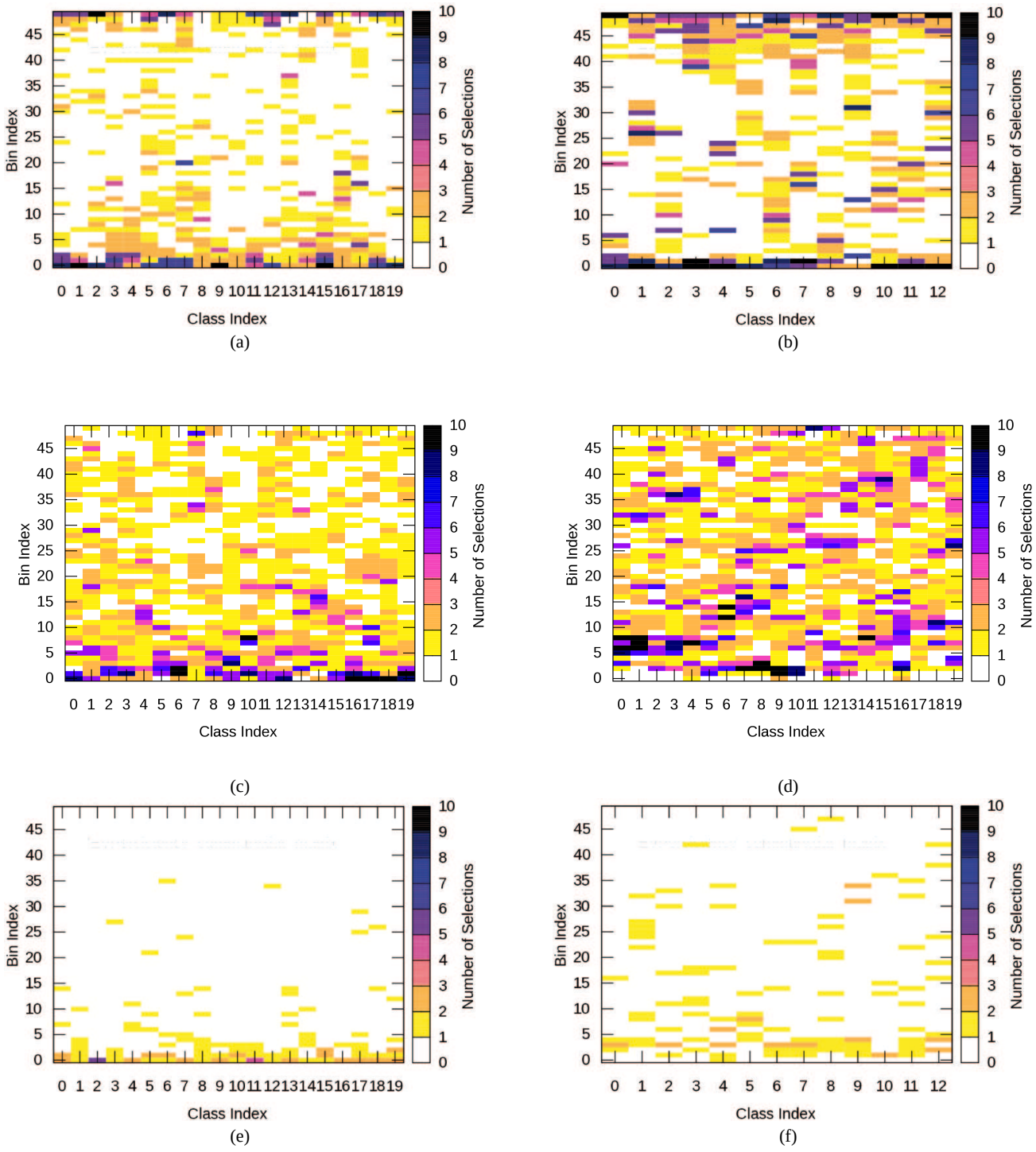


Fig. 8 Selected eigenvalues related to the point sets of the (first column) SHREC 2007 and (second column) SHREC 2010 data sets, with respect to (a,b) the AdaBoost algorithm, (b,c) Support Vector Machine, and (e,f) the first k eigenvalues. In all cases, the shapes are represented as point clouds. Each column of the heat-map matrices represents a shape class. The color of each matrix element represents the number of times the eigenvalues have been selected within 10 iterations of the Bootstrap Cross-Validation. The color bar at right side of the pictures is the map between colors and number of selections. The $x-$ and $y-$axis represent the class index and the spectrum bin, respectively.

set generated by the corresponding Bootstrap iteration and they are not used for shape characterization. Other eigenvalues have been selected several times within the 10 Bootstrap iterations (e.g., several eigenvalues with indices between 0 and 50) and their persistence suggests that they characterize the class of humans.

An analogous experiment has been performed by using the classifier based on the first k eigenvalues defined by Eq. (7). For each class of shapes the value of k is selected during the training phase. Figure 6(b) shows that the selected value for k changes at each Bootstrap iteration. This means that the sequences based on the first k eigenvalues strongly depends on the training set; indeed, such sequences are less representative for the class of human shapes. These experiments also show that (i) a set of selected eigenvalues is suitable to characterize the most relevant shape features and (ii) features that are not individually relevant may become relevant in the context of others [23].

Figure 7 corresponds to a heat-map matrix, which shows the selected features associated with each class of the SHREC 2007 data set through 10 iterations of the Bootstrap Cross-Validation. Each column of the matrix represents a class of the data set and the spectrum has been discretized into 50 bins, each of them representing 10 eigenvalues. The color associated with each entry represents the largest number of times that at least one eigenvalue belonging to the corresponding bin has been selected and the side bar gives the mapping between colors and selections' number. Figure 7(a) shows the results obtained with the AdaBoost algorithm after 20 iterations over a spectrum consisting of 500 eigenvalues. Since several matrix elements have a dark color, especially corresponding to small bin indices, we conclude that, for all the data set classes, AdaBoost is capable of identifying a set of eigenvalues that characterize the shape features shared by the class members. Moreover, the average number of selected eigenvalues for each class is 9. This means that a very small number of eigenvalues (about the 2% of the input spectrum) characterize the shape features shared by a given class. Figure 7(b,c) summarizes the results obtained with the Support Vector Machine and the first k eigenvalues approach by varying k between 1 and 500.

Similar experiments have been performed on the SHREC 2007 and SHREC 2010 data sets, whose shapes are represented as point clouds with 10K points. Figure 8 shows the characterization capability of the AdaBoost algorithm and Support Vector Machine on point clouds; furthermore, the results with respect to the first k eigenvalues is analogous to the corresponding experiment on triangle meshes. In this case, the eigenvalues that characterize the shape features shared by the class members have been selected among eigenvalues of both low and high magnitude. This behavior means that coarse features coded by point clouds are not sufficient to characterize class members and detailed features are necessary to improve the results.

	SHREC 2007		**SHREC 2010**
	Mesh 10KV	**Point Cloud**	**Point Cloud**
AdaBoost	**0.17**	**0.21**	0.27
First k	0.26	0.29	**0.19**
RFE	0.32	0.31	0.31
SVM	0.26	0.26	0.23
First 10 eigs	0.27	0.30	0.30
First 20 eigs	0.27	0.30	0.28
First 30 eigs	0.27	0.31	0.28
First 40 eigs	0.28	0.32	0.28
First 50 eigs	0.28	0.32	0.28

Table 1 Classification error on the SHREC 2007 and SHREC 2010 data sets, whose shapes have been represented as triangle meshes and point clouds with 10K points, respectively. Positive examples consists of the 25% of the class population.

5.2 Shape classification

The eigenvalues selected to characterize relevant shape features can also be exploited for classification. To this end, we will use the AdaBoost Strong Classifier, the Support Vector Machine, the Recursive Feature Elimination, and the classifier (7) for classification within the Bootstrap Cross-Validation framework. For each Bootstrap iteration, a training and a validation set are generated. According to the characterization experiments, the training set is used for selecting relevant eigenvalues and the shapes of the validation set are classified using the AdaBoost Strong Classifier or the first k eigenvalues. The classifier performances are evaluated with respect to classification accuracy (i.e., the lower is the classification error, the higher is the accuracy) and stability (i.e., small changes in the training data do not lead to large changes in the resulting classifier).

In our experiments, we have used both a small set of positive examples (Table 1) consisting of the 25% of the class population (5 and 14 shapes from the SHREC2007 and SHREC2010 data sets, respectively) and a large set of positive examples (Table 2) consisting of the 75% of the class population (15 and 42 shapes from the SHREC2007 and SHREC2010 data sets, respectively) and the same number of negative examples. Tables 1 and 2 summarize the classification error with respect to the eigenvalues selected by the AdaBoost algorithm, the RFE approach, and the first k eigenvalues approach (first three rows). Furthermore, we include the classification performances of the SVM classifier and the first $10-20-30-40$ and 50 eigenvalues for all the classes of the input data set. The Gaussian kernel has been used for both the SVM and RFE; the kernel width and the soft-margin parameter have been selected through the bootstrap resampling method.

	SHREC 2007		SHREC 2010
	Mesh 10KV	**Point Cloud**	**Point Cloud**
AdaBoost	**0.17**	0.20	0.29
First k	**0.17**	0.22	**0.13**
RFE	0.24	0.22	0.24
SVM	0.19	**0.18**	0.16
First 10 eigs	0.19	0.24	0.21
First 20 eigs	0.18	0.23	0.19
First 30 eigs	0.19	0.23	0.19
First 40 eigs	0.19	0.23	0.19
First 50 eigs	0.19	0.24	0.20

Table 2 Classification error on the SHREC 2007 and SHREC 2010 data sets, whose shapes have been represented as triangle meshes and point clouds with 10K points, respectively. Positive examples consists of the 75% of the class population.

The AdaBoost Strong Classifier yields good performance both on the triangle meshes and the point clouds of the SHREC 2007 data set, especially with a small training set (Table 1). Moreover, the stability of the selected eigenvalues guarantee that the strong classifier exploits the eigenvalues that characterize the most persistent shape features of the class (Figs. 7 and 8). Indeed, the classifier uses almost the same selected features for all the iterations of the Bootstrap Cross-Validation. The bad performance obtained for the SHREC2010 data set is due to the fact that each class consist of several modifications of the same shape model and many classes are overlapped. Indeed, AdaBoost is not capable of discriminating among them. Finally, we notice that the selection process based on the shape classes improves the classification performances obtained with the first k Laplacian eigenvalues especially with a small number of examples. In fact, at each iteration the AdaBoost algorithm selects the features capable of improving the classification performance; on the contrary, the RFE method eliminates the features that are less influent in the classification process. This behavior explains why the RFE decreases the performance of the SVM classifier.

6 Conclusions and future work

In this paper, we have shown that for a given class of 3D shapes it is possible to select a subset of the Laplacian eigenvalues that characterize the most relevant shape features shared by the class members. To run the feature selection and shape classification, which apply to 3D shapes represented as triangle meshes and point clouds, a small set of positive and negative examples is generally needed as training set. As future work, we will investigate how the selected eigenvalues can be exploited for the definition of more effective tools for shape similarity and classification.

Acknowledgements Special thanks are given to the anonymous reviewers for their valuable comments, which helped us to improve the presentation and content of the paper. We are grateful to Alexander and Micheal Bronstein for providing the SHREC 2010 data set. The SHREC2007 data set is courtesy of the AIM@SHAPE repository. This work has been partially supported by the FP7 CA FOKUS K3D.

References

1. A. Adamson and M. Alexa. Approximating and intersecting surfaces from points. In *Symp. on Geometry Processing*, pages 230–239, 2003.
2. A. Adamson and M. Alexa. Ray tracing point set surfaces. In *IEEE Shape Modeling International*, pages 272–282, 2003.
3. M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, and C. T. Silva. Point set surfaces. In *Proc. of Visualization*, pages 21–28, 2001.
4. N. Amenta and Y. J. Kil. Defining point-set surfaces. In *ACM Siggraph*, pages 264–270, 2004.
5. S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Y. Wu. An optimal algorithm for approximate nearest neighbor searching fixed dimensions. *Journal of the ACM*, 45(6):891–923, 1998.
6. M. Attene and G. Patanè. Hierarchical structure recovery of point-sampled surfaces. *Computer Graphics Forum*, (29):1905–1920, 2010.
7. M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computations*, 15(6):1373–1396, 2003.
8. M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7:2399–2434, 2006.
9. M. Belkin, J. Sun, and Y. Wang. Constructing Laplace operator from point clouds in $\mathbb{R}^d$. In *Proc. of the Symposium on Discrete Algorithms*, pages 1031–1040, 2009.
10. A. Ben-Hur, C. Soon Ong, S. Sonnenburg, B. Schoelkopf, and G. Raetsch. Support vector machines and kernels for computational biology. *PLoS Comput Biologgy*, 4(10):e1000173, 10 2008.
11. S. Biasotti, B. Falcidieno, P. Frosini, D. Giorgi, C. Landi, S. Marini, G. Patanè, and M. Spagnuolo. 3D Shape Description and Matching Based on Properties of Real Functions . In *Eurographics 2007 - Tutorials*, pages 949–998, Prague, 2007.
12. B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In *Proc. of the ACM Workshop on Computational Learning Theory*, pages 144–152, 1992.
13. A. M. Bronstein, M. M. Bronstein, B. Bustos, U. Castellani, M. Crisani, B. Falcidieno, L. J. Guibas, V. Murino I. Kokkinos, I. Isipiran, M. Ovsjanikov, G. Patanè, M. Spagnuolo, and J. Sun. Shrec 2010: robust feature detection and description benchmark. *Eurographics Workshop on 3D Object Retrieval*, 2010.
14. A. M. Bronstein, M. M. Bronstein, U. Castellani, B. Falcidieno, A. Fusiello, A. Godil, L.J. Guibas, I. Kokkinos, Z. Lian, M. Ovsjanikov, G. Patanè, M. Spagnuolo, and R. Toldo. Shrec 2010: robust large-scale shape retrieval benchmark. *Eurographics Workshop on 3D Object Retrieval*, 2010.
15. B. Bustos, D. A. Keim, D. Saupe, T. Schreck, and D. V. Vranić. Feature-based similarity search in 3D object databases. *ACM Computing Surveys*, 37(4):345–387, 2005.
16. D.-Y. Chen, X.-P. Tian, Y. Shen, and M. Ouhyoung. On visual similarity based 3D model retrieval. *Computer Graphics Forum*, pages 223–232, 2003.
17. F. R. K. Chung. *Spectral graph theory*. American Mathematical Society, 1997.

18. R. R. Coifman and S. Lafon. Diffusion maps. *Applied and Computational Harmonic Analysis*, 21(1):5–30, July 2006.
19. S. Fleishman, D. Cohen-Or, M. Alexa, and C. T. Silva. Progressive point set surfaces. *ACM Transactions on Graphics*, 22(4):997–1011, 2003.
20. Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *Proc. of Computational Learning Theory*, 1995.
21. Y. Freund and R. E. Schapire. A short introduction to boosting. In *In Proc. of the International Joint Conference on Artificial Intelligence*, pages 1401–1406, 1999.
22. Paraboschi L. Giorgi D., Biasotti S. Watertight models track. Technical Report 09/07, 2007.
23. I. Guyon, S. Gunn, M. Nikravesh, and L. Zadeh, editors. *Feature Extraction, Foundations and Applications*, volume 207 of *Studies in Fuzziness and Soft Computing*. Springer, 2006.
24. I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46:389–422, March 2002.
25. S. Hou, K. Lou, and K. Ramani. SVM-based semantic clustering and retrieval of a 3D model database. In *Journal of Computer Aided Design and Application*, pages 155–164, 2005.
26. S. Hou and K. Ramani. A probability-based unified 3D shape search. In *Conference on Semantic and Digital Media Technologies*, pages 124–137. Lecture Notes in Computer Science, 2006.
27. V. Jain and H. Zhang. A spectral approach to shape-based retrieval of articulated 3D models. *Computer Aided Design*, 39:398–407, 2007.
28. A. Kalaiah and A. Varshney. Modeling and rendering of points with local geometry. *IEEE Transactions on Visualization and Computer Graphics*, 9(1):30–42, 2003.
29. S. Lafon, Y. Keller, and R. R. Coifman. Data fusion and multicue data matching by diffusion maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(11):1784–1797, 2006.
30. H. Laga and M. Nakajima. A boosting approach to content-based 3D model retrieval. In *Proc. of Computer Graphics and Interactive Techniques*, pages 227–234, 2007.
31. H. Laga and M. Nakajima. Supervised learning of salient 2d views of 3D models. *The Journal of the Society for Art and Science*, 7(4):124–131, 2008.
32. C. Lange and K. Polthier. Anisotropic smoothing of point sets. *Computer Aided Geometric Design*, 22(7):680–692, 2005.
33. D. Levin. Mesh-independent surface interpolation. *Geometric Modeling for Scientific Visualization*, 3:37–49, 2003.
34. M. Mahmoudi and G. Sapiro. Three-dimensional point cloud recognition via distributions of geometric distances. *Graphical Models*, 71:22–31, 2009.
35. S. Marini, G. Patanè, M. Spagnuolo, and . Falcidieno. Feature selection for enhanced spectral shape comparison. In *Eurographics Workshop on 3D Object Retrieval*, 2010.
36. S. Marini, M. Spagnuolo, and B. Falcidieno. Structural shape prototypes for the automatic classification of 3D objects. *IEEE Computer Graphics and Applications*, 27(4):28–37, 2007.
37. Biasotti S. Marini S., Paraboschi L. Shape retrieval contest 2007 (SHREC07): Partial matching track. Technical Report 10/07, 2007.
38. D. Mateus, F. Cuzzolin, R. Horaud, and E. Boyer. Articulated shape matching using locally linear embedding and orthogonal alignment. *IEEE International Conference on Computer Vision*, pages 1–8, 2007.
39. B. Mederos, L. Velho, and L. H. de Figueiredo. Moving least squares multiresolution surface approximation. *SibGrapi*, pages 19–26, 2003.
40. B. Mohar and S. Poljak. Eigenvalues in combinatorial optimization. *Combinatorial and graph-theoretical problems in linear algebra*, 23(98):107–151, 1993.
41. M.Pauly and M. Gross. Spectral processing of point-sampled geometry. In *ACM Siggraph*, pages 379–386, 2001.
42. R. Ohbuchi and J. Kobayashi. Unsupervised learning from a corpus for shape-based 3D model retrieval. In *Proc. of the Workshop on Multimedia Information Retrieval*, pages 163–172, 2006.
43. R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin. Shape distributions. *ACM Transactions on Graphics*, 21(4):807–832, 2002.
44. M. Pauly, M. H. Gross, and L. Kobbelt. Efficient simplification of point-sampled surfaces. In *IEEE Visualization*, 2002.
45. M. Pauly, L. P. Kobbelt, and M. Gross. Point-based multiscale surface representation. *ACM Transactions on Graphics*, 25(2):177–193, 2006.
46. H. Pfister, M. Zwicker, J. van Baar, and M. Gross. Surfels: Surface elements as rendering primitives. In *ACM Siggraph*, pages 335–342, 2000.
47. M. Reuter, . Biasotti, D. Giorgi, G. Patanè, and M. Spagnuolo. Discrete laplace-beltrami operators for shape analysis and segmentation. *Computers & Graphics*, 33:381–390, 2009.
48. M. Reuter, F.-E. Wolter, and N. Peinecke. Laplace-Beltrami spectra as Shape-DNA of surfaces and solids. *Computer-Aided Design*, 38(4):342–366, 2006.
49. S. Rusinkiewicz and M. Levoy. Qsplat: a multiresolution point rendering system for large meshes. In *ACM Siggraph*, pages 343–352, 2000.
50. R. M. Rustamov. Laplace-Beltrami eigenfunctions for deformation invariant shape representation. In *Proc. of the Symposium on Geometry processing*, pages 225–233, 2007.
51. P. Shilane and T. Funkhouser. Selecting distinctive 3D shape descriptors for similarity retrieval. In *Proc. of Shape Modeling and Applications*, page 18, 2006.
52. J. W. Tangelder and R. C. Veltkamp. A survey of content based 3D shape retrieval methods. *Multimedia Tools and Applications*, 39(3):441–471, 2008.
53. J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.
54. K. Tieu and P. Viola. Boosting image retrieval. *International Journal of Computer Vision*, 56(1-2):17–36, 2004.
55. B. Vallet and B. Levy. Spectral geometry processing with manifold harmonics. *Computer Graphics Forum 27(2)*, 2008.
56. H. Xie, J. Wang, J. Hua, H. Qin, and A. Kaufman. Piecewise C^1 continuous surface reconstruction of noisy point clouds via local implicit quadric regression. In *IEEE Visualization*, page 13, 2003.
57. M. Zwicker, H. Pfister, J. van Baar, and M. Gross. Ewa splatting. *IEEE Transactions on Visualization and Computer Graphics*, 8(3):223–238, 2002.