Investigating the Effects of Multiple Factors Towards More Accurate 3-D Object Retrieval

Petros Daras, *Member, IEEE*, Apostolos Axenopoulos, and Georgios Litos

***Abstract*—This paper proposes a novel framework for 3-D object retrieval, taking into account most of the factors that may affect the retrieval performance. Initially, a novel 3-D model alignment method is introduced, which achieves accurate rotation estimation through the combination of two intuitive criteria, plane reflection symmetry and rectilinearity. After the pose normalization stage, a low-level descriptor extraction procedure follows, using three different types of descriptors, which have been proven to be effective. Then, a novel combination procedure of the above descriptors takes place, which achieves higher retrieval performance than each descriptor does separately. The paper provides also an in-depth study of the factors that can further improve the 3-D object retrieval accuracy. These include selection of the appropriate dissimilarity metric, feature selection/dimensionality reduction on the initial low-level descriptors, as well as manifold learning for re-ranking of the search results. Experiments performed on two 3-D model benchmark datasets confirm our assumption that future research in 3-D object retrieval should focus more on the efficient combination of low-level descriptors as well as on the selection of the best features and matching metrics, than on the investigation of the optimal 3-D object descriptor.**

***Index Terms*—3-D object retrieval, descriptor extraction, feature selection, manifold learning, rotation estimation.**

I. Introduction

THE recent innovations in advanced 3-D scanning mechanisms, computer-aided modeling tools, display and rendering devices resulted in an explosion of the number of available 3-D models over the Internet. It is now possible to easily construct complete 3-D geometry models with relatively low cost and time consumption. This increasing amount of 3-D content intensified the need for effective search through the various online media databases. Towards this direction, extensive research has been conducted in the area of 3-D content-based search and retrieval.

3-D object retrieval methods exploit the low-level features (e.g., shape), which are automatically extracted from 3-D objects, in order to retrieve semantically similar objects. Starting from simple, heuristic approaches that detect a few generic geometric features in the surface or the volume of a 3-D model, 3-D content-based search has evolved over the years to include highly complex algorithms that apply sophisticated mathematics so as to detect fine discriminating details and achieve the highest possible accuracy. Recently, though, it has become apparent that further research towards investigating even more complex algorithms can hardly improve the performance of existing state-of-the-art methods. Therefore, much effort is now put into combining already existing 3-D shape descriptors into one unified similarity matching framework, which has been proven to be much more effective than using each descriptor separately. Results of the latest SHREC Contests [1], [2], a worldwide benchmarking initiative for 3-D shape retrieval, have demonstrated that the combination of already existing methods can achieve higher retrieval accuracy than a single highly complex algorithm.

Based on the above facts, the 3-D object retrieval framework, which is presented in this paper, does not focus on the investigation of a new method for descriptor extraction. It analyzes several factors that can significantly improve the performance of existing algorithms and proposes a complete solution for 3-D object retrieval, which can be used as reference for further research in this field. The factors analyzed in this paper include rotation normalization, feature selection, selection of the optimal similarity metric, combination of descriptors and weight optimization and, finally, manifold learning.

A. Background and Related Work

A typical 3-D object retrieval procedure usually consists of the following steps: firstly, a preprocessing phase takes place, where the 3-D model is translated so that its center of mass coincides with the center of a coordinate system and scaled in order to lie within a bounding sphere of radius 1. Rotation normalization is also desired when a rotation-dependent descriptor extraction method is used. The second step involves feature extraction, where low-level descriptors are extracted from the 3-D object, uniquely represent its shape. During search and retrieval, the low-level descriptors of the query object are matched with the low-level descriptors of the objects in a database, using an appropriate dissimilarity metric, and the most similar objects are returned to the user. Apart from the above procedures, a 3-D object retrieval procedure can be enhanced by the following actions: 1) feature selection applied on the extracted low-level descriptors; 2) appropriate selection of the optimal distance measure; 3) combination of more than one low-level descriptors; and 4) re-ranking of the search results using manifold learning approaches. These actions, although optional, can significantly improve the performance of an already existing 3-D object retrieval system, therefore, they should by no means be underestimated.

Manuscript received March 27, 2011; revised August 23, 2011; accepted November 05, 2011. Date of publication November 15, 2011; date of current version March 21, 2012. This work was supported by the EC-funded project I-SEARCH. The associate editor coordinating the review of this manuscript and approving it for publication was Christophe De Vleeschouwer.

The authors are with the Informatics and Telematics Institute, Centre for Research and Technology Hellas, Thessaloniki GR-57001 Greece (e-mail: daras@iti.gr; axenop@iti.gr; gl@iti.gr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2011.2176111

Pose normalization is an essential step before almost every descriptor extraction method. While translation and scale normalization can be easily handled, rotation normalization is not a trivial task. The most popular solution for rotation estimation relies on principal component analysis (PCA) [3], which is based on the computation of the 3-D objects' inertia moments. Despite its wide adoption by several descriptor extraction methods, PCA has been proven to be inefficient, due to the following inherent limitations: 1) there are many cases of 3-D objects where it fails to properly align them, and 2) it does not provide information about the orientation of the principal axes. This led to further extensions of PCA to improve its performance, such as the continuous PCA (CPCA) [4]. CPCA, which is based on the eigenanvalue computation of the 3-D object's covariance matrix, is more stable than the traditional PCA. The limitations of PCA can be overcome by combining it with other rotation normalization methods. In [5], authors apply both PCA and a method based on virtual contact area (VCA) [6], in order to produce two rotated versions of the same object. Then, the volumes of the bounding boxes parallel to principal axes are computed and the rotated object with the minimum bounded volume is chosen.

A more recent approach for rotation estimation was presented in [7], where PCA is combined with a new measure based on rectilinearity. Rectilinearity is defined as the maximum ratio of the surface area to the sum of three orthogonal projected areas of the mesh. After rotation of the 3-D model using both PCA and rectilinearity, three orthogonal 2-D silhouettes are taken from the three principal axes for each rotated model. The rotated model with the minimum sum of valid pixels within the 2-D silhouettes is kept. Another method, which was presented in [8], is looking for symmetries within the 3-D object. It introduced a planar reflective symmetry transform (PRST), which computes reflection symmetry of a 3-D shape with respect to all possible planes. In [9], authors use CPCA as an initial rotation estimation step. Then, plane reflection symmetry is calculated. If the 3-D object has two or three symmetrical planes, the CPCA-generated pose is kept. Otherwise, an additional step is applied, which measures the local translational invariance towards a specific direction along the object. The method achieves good results in many cases where CPCA fails. However, the principal axes are usually returned in arbitrary order, which is not desired for search and retrieval tasks.

Concerning low-level feature extraction from 3-D objects, the existing methods can be classified into four main categories [5]: histogram-based, transform-based, graph-based, view-based and, finally, combinations of the above. The first category includes methods that integrate the local or global features extracted from 3-D objects into histograms [10]. Then, appropriate histogram comparison metrics are employed to measure the shape similarity. Histogram-based methods are, in general, easy to implement but usually they are not discriminating enough to make subtle distinctions between classes of shapes. In transform-based methods, the 3-D object is usually described as a function in the 3-D space. Then, a mathematical transformation is applied to this function, which captures specific geometrical features of it [13], [14], [24]. Finally, 2-D view-based methods consider the 3-D shape as a collection of 2-D projections taken either from canonical or non-canonical viewpoints. Each projection is then described by standard 2-D image descriptors [5], [15], [16]. The results of the last three 3-D Shape Retrieval Contests (SHREC09 [17], SHREC10 [2], SHREC11 [1]) have proven that 2-D view-based methods achieve the highest performance among all other categories of 3-D object retrieval methods. Their only drawback is that they discard valuable 3-D information (due to the self-occlusion).

All the above categories of descriptor extraction methods are suitable for rigid, global-shape 3-D object retrieval. When it comes to non-rigid or partial 3-D object retrieval, graph-based methods [54] provide the optimal solution, since they have the potential of encoding geometrical and topological shape properties in a more faithful and intuitive manner. In [47], a technique, called topology matching, is introduced, which calculates the similarity between polyhedral models by comparing multi-resolutional Reeb graphs (MRGs). This work has been further extended in [12] where the Reeb graph is augmented with geometrical attributes leading to the creation of a flexible multi-resolutional representation, called an augmented Reeb graph. A mesh decomposition method is presented in [48]. The mesh is represented as an attributed graph, which is considered the signature of the object. In [11], a method which combines topological and geometrical information is proposed, which is invariant to geometric transformations of a 3-D object, as well as to the different poses of articulated objects. Recent advances in non-rigid 3-D object retrieval can be found in SHREC'11 Track: Retrieval on Non-rigid 3-D Watertight Meshes [49], where the best methods worldwide are compared in a new benchmark dataset. Similar to articulated shape matching, partial 3-D object retrieval [50], [51] focuses on identifying salient points on the 3-D objects' meshes, followed by a local shape descriptor, which is applied around each interest point. Then, matching subsets of salient points are found so that the local shape dissimilarity and deformation is minimal. The drawbacks of graph-based methods are that they do not generalize easily to all 3-D shape representation formats and they require dedicated matching schemes, thus, they are not suitable for general-purpose 3-D object retrieval tasks.

The 3-D object descriptors extracted by using the above methods are usually represented as high-dimensional vectors. The dimensionality of the descriptor vectors can vary from a few hundreds to tens of thousands real or integer values. As the descriptor size increases, search and retrieval in very large databases becomes prohibitive, since similarity matching of large descriptor vectors requires high response times. This problem can be overcome by using feature selection methods, which have been widely used in pattern analysis in order to select the most significant features of a given descriptor. In 3-D object retrieval, feature selection can significantly reduce the size of a descriptor vector, without affecting its retrieval accuracy, thus, providing a more compact representation, suitable for search in large databases. Feature selection is not a new research area in machine learning [18]. A relatively recent approach, called correlation-based feature selection (CFS) [19] uses a correlation based heuristic to evaluate the worth of features. The method was tested on discrete and continuous class data sets and demonstrated superior performance over similar approaches. A similar approach was presented in [20]. The method is based on a probabilistic algorithm, which uses randomness to guide search in such a way that a correct solution is guaranteed even if unfortunate choices are made.

In 3-D object retrieval, feature selection has been only recently introduced. In [41], an optimized feature selection (OFS) approach is applied to select the optimal descriptor vector among Tchebichef moments, Centroid-based Fourier descriptors, Zernike moments and Silhouette descriptors. In [42], an information-theoretic filter, based on mutual information, is applied on a set of 3-D object descriptors. The selection criterion proposed in [42] determines which features should be discarded first and which ones should be retained in order to keep the mutual information between the features set and the class label high. Finally, a framework for selecting the Laplacian eigenvalues of 3-D shapes that are more relevant for shape comparison and classification is presented in [43]. Three different approaches are compared: the first k eigenvalues, the Hill Climbing technique, and the AdaBoost algorithm.

3-D object retrieval accuracy can be further improved if more than one low-level descriptors are combined in order to produce one unified descriptor vector. In [21], a descriptor for search and retrieval of 3-D objects was presented, which combines two view-based methods (silhouette-based views and depth-based views) with a transform-based method. The combination of descriptors demonstrated better performance than each descriptor separately, in several test databases [21]. Similar conclusions are drawn in [5], where the combination of a transform-based method with a view-based method achieved higher retrieval accuracy than each separate descriptor in three benchmark databases. Combination of different descriptors is usually expressed as a weighted sum of their individual dissimilarities. In most of existing methods, these weights are determined heuristically, which, however, does not guarantee the maximum possible improvement. A more theoretical approach for weight estimation is given in [22], where the contribution of each descriptor to the overall measure is assessed based on an intuitive "*purity*" metric.

Finally, manifold ranking has been adopted to improve the performance of 3-D search and retrieval methods. The use of manifold ranking was based on the concept that low-level descriptors, in general, follow a nonlinear manifold structure, which makes classical Euclidean metrics inappropriate. By properly unfolding this manifold structure, a more representative feature space of lower dimension is achieved. In [23], a manifold learning approach based on Laplacian Eigenmaps was applied to the initial descriptors of several 3-D objects in order to create a new low-dimensional feature space, where each 3-D object is mapped. Similarity matching among 3-D objects is, then, reduced to calculating the L-2 distances of their corresponding low-dimensional features. In SHREC 2010 contest [2], the first two ranked methods were view-based methods using manifold ranking as a post-processing step to improve the retrieval accuracy.

In this paper, a complete framework for 3-D object retrieval is proposed. The framework takes into account all above mentioned factors that affect the retrieval accuracy, namely rotation normalization, combination of low-level descriptors, feature selection, selection of optimal distance measures and manifold ranking. Although most of the methods used in each part of the proposed framework have been already presented in the literature, such an approach that effectively merges all the above factors has not been proposed so far, to the best of our knowledge. Through this study, very useful conclusions can be drawn, regarding the future research directions in 3-D object retrieval.

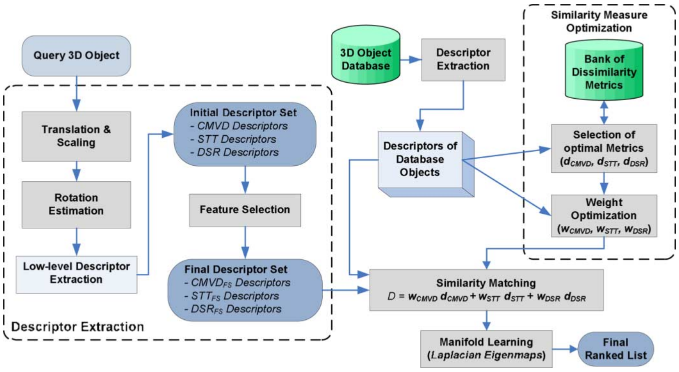
Query 3D Object
Translation & Scaling
Rotation Estimation
Low-level Descriptor Extraction
Initial Descriptor Set
- CMVD Descriptors
- STT Descriptors
- DSR Descriptors
Feature Selection
Final Descriptor Set
- CMVD$_{FS}$ Descriptors
- STT$_{FS}$ Descriptors
- DSR$_{FS}$ Descriptors
Descriptor Extraction
3D Object Database
Descriptor Extraction
Descriptors of Database Objects
Similarity Measure Optimization
Bank of Dissimilarity Metrics
Selection of optimal Metrics (d_{CMVD}, d_{STT}, d_{DSR})
Weight Optimization (w_{CMVD}, w_{STT}, w_{DSR})
Similarity Matching
$D = w_{CMVD}\, d_{CMVD} + w_{STT}\, d_{STT} + w_{DSR}\, d_{DSR}$
Manifold Learning (*Laplacian Eigenmaps*)
Final Ranked List

Fig. 1. Block diagram of the proposed framework.

The rest of the paper is organized as follows: in Section II, an overview of the overall framework is presented, while in Section III, the rotation estimation method, which is introduced in this paper, is analyzed. In Section IV, a short description of the low-level descriptor extraction takes place followed by the proposed feature selection step. Similarity matching, which includes selection of the optimal metric and weight optimization, is described in Section V. In Section VI, the manifold learning approach is given, while Section VII analyzes the experimental results. Finally, conclusions are drawn in Section VIII.

II. Method Overview and Innovations

The proposed framework consists of the following procedures Fig. 1: 1) preprocessing and descriptor extraction, 1) dissimilarity measure optimization, and 3) manifold learning. During pre-processing, the 3-D object is translated, scaled and rotated in order to be aligned to a canonical coordinate frame. Then, low-level feature extraction takes place, where three different methods are applied: the Compact Multi-View Descriptor (view-based) [5], the Spherical Trace Transform (transform-based) [24], and the Depth Silhouette Radical Extent descriptor (combined transform and view-based) [21]. The outcome is an initial set of low-level descriptors. The feature selection step reduces the dimensionality of the initial descriptors, by selecting only the most discriminative descriptors per method.

During dissimilarity measure optimization, the following actions are taken to improve the performance of the low-level descriptors: firstly, the optimal dissimilarity metric per descriptor is selected from a bank of the most widely known metrics and, secondly, the three descriptors are combined to a unified dissimilarity measure, where the weights for each descriptor are optimized using machine learning techniques.

Manifold learning, finally, is used to transform the 3-D object descriptors into a new low-dimensional feature space. In this feature space, each 3-D object of the database is represented by a low-dimensional vector, which is used as its descriptor to retrieve similar objects.

Although most of the constituting parts of the proposed framework have been already presented in previous works, it still provides numerous innovative features, which are presented below:

It introduces a novel method for accurate alignment of 3-D objects: it uses continuous PCA as an initial rotation estimation step, followed by a computation of symmetries in principal planes (0xy, 0xz, 0yz). A similar approach has been also presented in [9] producing accurate results for symmetric models. However, in the proposed method, a metric based on rectilinearity [7] is applied to non-symmetric models, as a next step, which achieves a very good estimation of the correct pose. The combination of the above intuitive properties achieves the most accurate 3-D model alignment that has ever been reported.

It provides an intuitive solution for dissimilarity measure optimization: the dissimilarity between two 3-D objects is calculated by combining the dissimilarities of three different low-level descriptor vectors. For each descriptor, the optimal dissimilarity metric is chosen. Although similar approaches chose the best among a limited set of distance metrics, in this paper, an extensive study on the most well-known distance metrics has been performed, which provides a complete reference for future works. Moreover, the weights for each descriptor have not been arbitrarily chosen, but they are calculated using an appropriately selected optimization method. Such an approach has not been reported so far for weight optimization in 3-D object retrieval.

It provides a complete framework for 3-D object retrieval: the success of the proposed approach is based on the efficient combination of most of the possible factors that affect the retrieval performance. It is the first time, to the best of our knowledge, that rotation estimation, feature selection, dissimilarity measure optimization and manifold learning are combined into a single framework for 3-D object retrieval. Experimental results proved that all of the above factors contribute to achieve the best possible retrieval accuracy.

III. Rotation Estimation

The rotation estimation method introduced in this paper consists of the following steps: firstly, after translation and scaling, CPCA [4] is applied to the input 3-D object to produce a first pose estimation. It has been already proven [9] that CPCA achieves good performance for symmetric models. Then, the reflection symmetry for the three CPCA-coordinate planes (0xy, 0xz, 0yz) is computed. If symmetry is observed in two or three coordinate planes, the transformation is kept as it is and the process terminates. In case symmetry is observed in only one or zero coordinate planes, then, the algorithm proceeds to the correction step based on rectilinearity. The outcome of this step is finally kept as the result rotation estimation.

A. Computing Symmetry

There are several ways to find symmetries in 3-D objects. In [8], a planar reflective symmetry transform (PRST) is introduced. PRST computes reflection symmetry of a 3-D shape with respect to all possible planes. Other related works compute symmetry based on octrees [25], Gaussian Images [26] and generalized moments [27]. In [9], plane reflection symmetry is computed in two ways: the symmetry distance, which is the minimum mean squared distance of a shape to its perfectly symmetric shape, and the symmetry descriptor similarity [28], which is the distance between the shape descriptor of a shape and the shape descriptor of its perfectly symmetric shape.

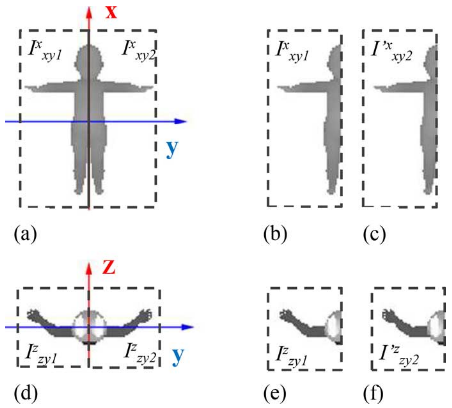

Fig. 2. Computing symmetry for plane 0xz.

In this paper, a method for fast approximate symmetry computation of the CPCA coordinate planes is proposed. It is based on the 2-D symmetry computation of depth images extracted from the three CPCA principal axes. More specifically:

Let I_{xy} be a 2-D depth image extracted from the third principal axis (z) of 3-D model M, which is rotated using CPCA. The image is actually the projection of the 3-D model on plane 0xy, adding depth information. The depth image is depicted in Fig. 2(a). If plane 0xz is a symmetry plane, then the x-axis is a symmetry axis of image I_{xy}. This means that if we take the two half images I^x_{xy1} and I^x_{xy2}, which are divided by x-axis, and flip the second half image, the resulting half images I^x_{xy1} and I'^x_{xy2} should be similar [Fig. 2(b) and 2(c)]. A metric for similarity of these half images is the correlation coefficient:

$$Corr^x_{xy} = \frac{\sum_{i=1}^{n/2}\sum_{j=1}^{n} I^x_{xy1}(i,j)\cdot I'^x_{xy2}(i,j)}{\sqrt{\sum_{i=1}^{n/2}\sum_{j=1}^{n}\left(I^x_{xy1}(i,j)\right)^2}\cdot\sqrt{\sum_{i=1}^{n/2}\sum_{j=1}^{n}\left(I'^x_{xy2}(i,j)\right)^2}}. \tag{1}$$

In order for I^x_{xy1} and I'^x_{xy1} to be similar, correlation coefficient should be greater than a predefined threshold:

$$Corr^x_{xy} > t. \tag{2}$$

Similarly, a 2-D depth image I_{yz} extracted from the first principal axis (x) is the projection of the 3-D model on plane 0yz. If 0xz is a symmetry plane, then the z-axis is a symmetry axis of I_{yz} and the half images I_{yz} and I^z_{yz1} should be similar [Fig. 2(d)–2(f)], i.e.,

$$Corr^z_{yz} > t. \tag{3}$$

Summarizing the above criteria, if 0xz is a symmetry plane, then both inequalities (2) and (3) must hold. The inverse is not mathematically valid but, in practice, it can provide a good estimation for symmetry. In other words, if (2) and (3) hold, then plane 0zy can be regarded as symmetric with a good degree of approximation. Similar inequalities are defined to calculate symmetries for planes 0xy and 0yz. The values of threshold t have been experimentally determined $(t = 0.9)$ [33].

B. Rectilinearity-Based Correction

By applying the symmetry computation method to the three CPCA-coordinate planes, zero to three symmetries may be detected. If two or three of the CPCA-coordinate planes are symmetry planes, the pose estimation made by CPCA is finally kept and the algorithm terminates. If less than two plane symmetries are detected (zero to one), the algorithm proceeds to the correction step based on rectilinearity.

The rectilinearity measure was initially proposed for pose estimation of 3-D objects in [7]. It is based on the maximum ratio of the surface area to the sum of three orthogonal projected areas of the mesh. The rectilinearity measure is invariant under scale translation and rotation and it can be applied to both open and closed meshes as well as to degenerate meshes. The rectilinearity of a 3-D mesh M is given by the following equation:

$$\begin{aligned} R(M) &= \max_{\alpha,\beta,\gamma\in[0,2\pi]} \frac{S(M)}{P(M,\alpha,\beta,\gamma)} \\ &= \max_{\alpha,\beta,\gamma\in[0,\pi/2]} \frac{S(M)}{P(M,\alpha,\beta,\gamma)} \end{aligned} \tag{4}$$

where $S(M)$ is the surface area of the given mesh $P(M,\alpha,\beta,\gamma)$ is the sum of the three projected areas of M and $\alpha,\beta,\gamma \in [0,2\pi]$ are the angles by which the coordinate frame is rotated around x, y, and z axis, respectively. The rectilinearity measure is invariant to flipping of the coordinate axes, thus the search range of the optimal angles (α,β,γ) can be narrowed to $[0,/2\pi]$.

The set of angles (α,β,γ) that maximize (4) results in the most stable pose of the 3-D mesh M, which can provide an adequate pose estimation criterion. This approach has been used in [7] combined with PCA and it performs better than PCA alone. However, it has several drawbacks, which are analyzed below.

The first is that rectilinearity and PCA are applied in parallel and the best rotation estimation among the two is finally chosen. The selection of the best rotation estimation is based on the computation of the number of valid pixels of three silhouettes, projected on the planes 0xy, 0xz, 0yz, for the two rotated meshes. Then, the model that yields the smaller value is selected as the final normalization result. The valid-pixel criterion is not very accurate, though, since the pose with the smaller value of valid pixels is not always the best one. Another shortcoming is that, in [7], the entire range of angles $\alpha,\beta,\gamma \in [0,\pi/2]$ is scanned to find the optimal solution. The problem of maximizing (4) is a nonlinear optimization problem, which requires high computational time. The computational time increases as the number of polygons of the given mesh M does and it may become prohibitively large, especially for real-time search and retrieval tasks. Moreover, the rectilinearity measure does not give any information about the correct order of the principal axes.

The combined pose estimation (CPE) method proposed in this paper can successfully overcome the above limitations. First of all, the rectilinearity-based correction step is applied just after the CPCA method, not in parallel with it. No selection criterion similar to valid-pixel criterion is applied, thus, the result of the correction step is always retained.

Then, depending on the number of detected plane symmetries (zero or one), two different variations of the correction step are realized: If one plane symmetry is detected, the 3-D object is further rotated only around the axis which is perpendicular to the symmetry plane, that is if 0zy is the symmetry plane, the model is rotated only around the x axis. This results in more accurate poses than those produced by successively rotating the model around x, y and z axes. In the case that no plane symmetry is detected, the model is successively rotated around x, y and z axes, which is a computationally expensive procedure. However, in the proposed method, rotation around x, y and z axes is applied only for zero plane symmetries, while in [7] it is applied in all cases.

IV. Descriptor Extraction

After proper pose normalization, a set of geometric features (low-level descriptors) is extracted from the 3-D model, which uniquely represents its shape. In the framework presented in this paper, descriptors have been extracted using three different methods: the Compact Multi-View Descriptor (CMVD) [5], the Spherical Trace Transform (STT) [24], and the Depth Silhouette Radical Extent descriptor (DSR) [21]. The reason for choosing the above descriptors was to combine the properties of both transform-based and view-based methods, which have been proven to achieve better retrieval performance than histogram-based and graph-based methods [1]. However, the framework can be further extended so as to include more types of descriptors.

CMVD, which was initially presented in [5], provides a compact and at the same time efficient low-level descriptor vector. It is based on the matching of multiple 2-D views, which can be extracted from a 3-D object by selecting a set of different viewpoints. In order to be uniformly distributed, the viewpoints are chosen to lie at the 18 vertices of a regular 32-hedron, which is produced by tessellation of octahedron at the first level. The set of uniformly distributed views consists of 2-D binary images of size 128 × 128 pixels. In each image, three rotation-invariant and flip-invariant functionals are applied in order to produce the final set of descriptors per view. These functionals are the 2-D Polar-Fourier Transform, 2-D Zernike moments, and 2-D Krawtchouk moments. A detailed description of the above functions is available in [24]. The dimension N_{CMVD} of the resulting descriptor vector is $N_{CMVD} = N_{\mathrm{FT}} + N_{\mathrm{Zern}} + N_{\mathrm{Kraw}}$, where $N_{\mathrm{FT}} = 78$, $N_{\mathrm{Zern}} = 56$, and $N_{\mathrm{Kraw}} = 78$ are the number of Polar Fourier coefficients, Zernike moments, and Krawtchouk moments, respectively.

STT, on the other hand, is applied to the volume of the 3-D object, which is represented as a binary volumetric function. At a first step, a set of concentric spheres is defined, centered at the mass center of the object. For every sphere, a set of planes which are tangential to the sphere is also defined. The intersection of each plane with the object's volume provides a spline of the object, which can be treated as a 2-D image. Next, 2-D rotation invariant functionals F are applied to this 2-D image, producing a single value. Thus, the result of these functionals when applied to all splines, is a set functions defined on every sphere whose range is the results of the functional. Finally, a rotation invariant transform T is applied on these functions, in order to produce rotation invariant descriptors. Again, for STT, 2-D Krawtchouk moments and 2-D Zernike Moments are used as 2-D functionals. A more detailed description of the extraction of these descriptors is available in [24]. The dimension of descriptor vectors is $N_{\mathrm{STT,Zern}} = 1080$ for the descriptors based on the 2-D Zernike Moments and $N_{\mathrm{STT,Kraw}} = 1080$ for the descriptors based on the Krawtchouk 2-D functional.

The DSR descriptor was introduced in [21]. It combines the Depth Buffer descriptor, the Silhouette descriptor, and the Radical Extent descriptor. At a preprocessing step, CPCA is applied to the 3-D object. In order to extract the 2-D views, the 3-D object is projected perpendicularly on the coordinate hyperplanes. Three silhouette images and six depth buffer images are extracted. In the case of silhouette images, a 1-D FFT transform is applied to the contour which approximates the silhouette. This descriptor is invariant to rotation of the 2-D view. In the case of depth buffer images, a 2-D FFT transform is applied to the depth image. In the Radicalized Spherical Extent descriptor, the 3-D model is described by a spherical function which decomposes the model into a sum of concentric shells and gives the maximal distance of the model from the center of mass as a function of angle and the radius of the equivalent shell. The spherical function is represented by spherical harmonics coefficients. The above descriptors are concatenated in order to form a single descriptor vector for each 3-D object. The dimension of the DSR descriptor is $N_{DSR} = 472$.

A. Applying Feature Selection on the Initial Descriptor Vectors

The descriptor extraction process presented above, results in a set of high-dimensional descriptor vectors per 3-D object. When dealing with relatively small 3-D object databases (e.g., up to few thousands of 3-D models), the descriptor vector size is not a critical issue. When it comes to databases with hundreds of thousands of models, the high dimensionality of descriptor vectors becomes a major scalability issue. Therefore, several attempts have been made to reduce the size of the resulting descriptors.

Feature selection methods have been widely used in machine learning to select the most representative features within a given descriptor. In 3-D object retrieval, a few attempts towards this direction have been recently introduced [42], [43]. In this paper, the following approaches for feature selection have been tested: *Correlation-based Feature Selection (CFS)* [19], *Chi-Square attribute selection* [29], *Consistency-based* [20], *Principal Components* [34], *Relief Attribute Selection* [35], *SVM* [36], *Adaboost* [43], and *Mutual Information*-based feature selection [42]. Among them, CFS, Chi-Square, Adaboost, and Mutual Information proved to be the most appropriate ones. A brief description of these approaches is given below.

Let $D = \{D_1, D_2, \ldots, D_{N_{DESCR}}\}$ be the initial set of attributes (features) before feature selection. The features D_{i} correspond to the low-level descriptors of a 3-D object and N_{DESCR} is the dimensionality of the initial descriptor vector. Feature selection will result in a subset $\boldsymbol{S}$ of $\boldsymbol{D}$, i.e., $S \subset D$, $S = \{f_1, f_2, \ldots, f_k\}$, where k is the number of the selected features f_{i} and $k < N_{DESCR}$. Our goal is to select those attributes f_{i} that differentiate between instances of different classes and have the same value for instances of the same class.

To achieve the latter, the CFS method is based on the hypothesis that "*good feature subsets contain features highly correlated with the class, yet uncorrelated with each other*". More specifically, the CFS algorithm ranks feature subsets according to a correlation-based heuristic evaluation function. The bias of the evaluation function is toward subsets that contain features that are highly correlated with the class and uncorrelated with each other (i.e., low inter-correlation among features). The feature subset evaluation function is given by the following equation:

$$M_S = \frac{k\bar{r}_{cf}}{\sqrt{k + k\,(k-1)\,\bar{r}_{ff}}} \tag{5}$$

where $\boldsymbol{M}_S$ is the heuristic "merit" of a feature subset $\boldsymbol{S}$ containing k features, $\bar{r}_{cf}$ is the mean feature-class correlation $f \in S$, and $\bar{r}_{ff}$ is the average feature-feature inter-correlation. A detailed description on how feature-class and feature-feature correlations are computed is available in [19].

CFS performs a ranking of all possible feature subsets $\boldsymbol{S}$ with respect to the value $\boldsymbol{M}_S$ of (5) and the best set of features (that maximizes $\boldsymbol{M}_S$) is eventually selected. In order to avoid exhaustive enumeration of all possible feature subsets, the *best first* search strategy is adopted, which searches the space of attribute subsets using Hill Climbing with backward technique [43].

Apart from CFS, Chi-Square attribute selection also focuses on selecting the optimal subset $\boldsymbol{S}$ of features from an initial set $\boldsymbol{D}$ of descriptors. The difference here is that instead of measuring a correlation-based evaluation function, the Chi-Square algorithm evaluates the worth of an attribute by computing the value of the chi-squared statistic:

$$\chi^2 = \sum_{i=1}^{N_C} \sum_{j=1}^{k} \frac{(N_{ij} - E_{ij})^2}{E_{ij}} \tag{6}$$

where N_{C} is the number of classes of the training set, k is the number of selected features, N_{ij} is the number of instances of the training set for which feature j belongs to class i, and E_{ij} is calculated as follows:

$$E_{ij} = \frac{\sum_{u=1}^{N_C} N_{uj} \cdot \sum_{u=1}^{k} N_{iu}}{N_t} \tag{7}$$

where N_t is the total number of instances of the training set.

The feature selection method introduced in [42] estimates the mutual information I between the features set and the class label:$I(\mathbf{S}_{N_{Descr}}; \mathbf{C}) = H(\mathbf{S}_{N_{Descr}}) - H(\mathbf{S}_{N_{Descr}}|\mathbf{C})$, where $\mathbf{S}_{N_{Descr}}$ is a matrix of size $N \times N_{Descr}$ and $\boldsymbol{C}$ of size $N \times 1$, N is the number of sample objects and N_{Descr} is the total number of descriptors. The entropies H are estimated using the kNN-based method developed by Leonenko [44]. The method starts with the entire feature set and determines which feature to discard in order to produce the smallest decrease of $I(\mathbf{S}_{N_{Descr}-1}; \mathbf{C})$. The process is, then, repeated for the features of the remaining set, until only one feature is left. A detailed description of the method is available in [42].

AdaBoost has been used for feature selection in 3-D object retrieval in [43]. The goal of AdaBoost is to select those features that maximize the inter-class similarity and the dissimi-

TABLE I
PERCENTAGES OF PERFECT ALIGNMENT FOR THE 720 OBJECTS OF SHREC 2009 GENERIC SHAPE BENCHMARK

.	CPCA	PT-Sym.	CPCA-RECT	Proposed Method
Number of correctly aligned objects (out of 720)	475	510	565	613
Percentage of correct alignment (%)	65,97	70,83	78,47	85,14

larity among classes. Using a set of positive and negative examples together with a large set of features, AdaBoost makes the margin among the training examples as large as possible.

The above techniques for feature selection were applied to reduce the dimensionality of the CMVD, STT, and DSR descriptors and provide a compact representation, keeping at the same time their retrieval accuracy unaffected.

V. SIMILARITY MATCHING

The overall dissimilarity between two 3-D objects A and B can be calculated as the weighted sum of the dissimilarities of each descriptor separately, according to the following equation:

$$\begin{aligned} dis\,(A,B) = & \, w_{CMVD} \cdot dis_{CMVD}\,(A,B) \\ & + w_{STT} \cdot dis_{STT}\,(A,B) \\ & + w_{DSR} \cdot dis_{DSR}\,(A,B) \end{aligned} \tag{8}$$

where dis_{CMVD}, dis_{STT}, and dis_{DSR} is the dissimilarity of the CMVD, STT, and DSR descriptor vectors, respectively, and w_{CMVD}, w_{STT}, and w_{DSR} their corresponding weights. Several distance metrics have been already used in the literature to compute the dissimilarity between a pair of descriptor vectors, such as the Euclidean distance, the Manhattan distance, the Kullback-Leibler divergence, or the Cosine distance. However, in most of the cases, the selection of a specific distance metric is not justified, while a significant amount of existing metrics is neglected. In this paper, an extensive study of the most well-known distance metrics has been performed. The metric that achieved the best retrieval accuracy for each descriptor was eventually chosen.

A. Selection of the Optimal Distance Metric

Selecting the most appropriate dissimilarity metric for each descriptor vector is not that new. In [45], several well-known distance metrics are applied to a range of 3-D shape descriptors in order to select the ones that achieve the highest performance for each descriptor. In this paper, an in-depth survey has been carried out, where the most widely used distance metrics were gathered from the literature and tested with our 3-D object descriptors in terms of retrieval performance. A summary of all metrics, which were tested, is given in Table I. Apart from the ones presented in Table I, the Earth Mover's Distance [30] and the Diffusion Distance [31] were also tested. However, these so-called "*cross-bin*" distances are recommended for histogram comparison and time-varying signals (e.g., sound), thus they did not reach their full potential in our descriptors, while being at the same time considerably slower than the other methods. The assessment was made in terms of precision-recall and the performance of all metrics is presented in the Experimental Results Section.

Regarding the CMVD descriptor, the "*Squared L-1 distance*" was proved to be the optimal metric. Let $\boldsymbol{D}_v$ be the descriptor vector of the v^{th} view, which is extracted according to the CMVD method. The dissimilarity between a corresponding pair of views of two 3-D models A and B, in terms of Squared L-1 distance, is given by

$$d_v\,(A,B) = \sum_{i=1}^{N_{CMVD}} \left| D_v^A(i) - D_v^B(i) \right|^{1/2}. \tag{9}$$

By summing the dissimilarities of the corresponding pairs of views, the CMVD-besed dissimilarity dis_{CMVD} between the models A and B is given by the following equation:

$$dis_{CMVD}\,(A,B) = \sum_{v=1}^{N_V} d_v(A,B) \tag{10}$$

where N_v is the number of uniformly distributed views of the 3-D object.

In the case of the STT descriptor, the best performance was achieved by using the so-called "*X distance*" [24], which is given by

$$dis_{STT}\,(A,B) = 2 \cdot \sum_{i=1}^{N_{STT}} \frac{\left| D_{STT}^A(i) - D_{STT}^B(i) \right|}{\left| D_{STT}^A(i) + D_{STT}^B(i) \right|} \tag{11}$$

where D_{STT} is the STT descriptor of the 3-D object and N_{STT} the dimensionality of the descriptor vector.

Finally, for DSR descriptor, the "X^2 *distance*" [37] was selected as the optimal metric:

$$dis_{DSR}\,(A,B) = 2 \cdot \sum_{i=1}^{N_{DSR}} \frac{\left(D_{DSR}^A(i) - D_{DSR}^B(i) \right)^2}{\left(D_{DSR}^A(i) + D_{DSR}^B(i) \right)} \tag{12}$$

where D_{DSR} is the DSR descriptor of the 3-D object and N_{DSR} the dimensionality of the descriptor vector. It must be noted that in (9) and (11) the descriptor vectors are the compact representations resulted after the feature selection step.

B. Weight Optimization for the Combined Distance Measure

After selecting the optimal metrics for dissimilarities dis_{CMVD}, dis_{STT}, and dis_{DSR} of (8), the weights w_{CMVD}, w_{STT}, and w_{DSR} need to be determined. A variety of intelligent computing algorithms can be adopted to calculate the optimal weights, such as genetic algorithms, artificial neural network, simulated annealing algorithm, ant colony optimization, etc. In this paper, an optimization method, called particle swarm optimization (PSO) [32], was selected since it produced very satisfactory results. PSO is a global optimization algorithm, similar to genetic algorithm, motivated by social behavior of organisms such as bird flocking and fish schooling. It is a computational method that optimizes a problem in which a best solution can be represented as a point or surface in an n-dimensional space. PSO iteratively tries to improve a candidate solution with regard to a given measure of quality (fitness function). PSO establishes a population (swarm) of candidate solutions, known as particles that move around in the search space, and are guided by the best found positions, updated while better positions are found by the particles.

In our approach, the population of candidate solutions is the weights w_{CMVD}, w_{STT}, and w_{DSR}, which can take arbitrary values within the range [1]. The fitness function to be optimized is the average Tier-1 precision, which is calculated on an appropriately selected train dataset. Each 3-D object of this dataset is used as query to retrieve similar objects, using (8) as dissimilarity metric. This produces N_t lists of retrieved results ranked in ascending order, where N_t is the number of 3-D objects of the train dataset. The Tier-1 precision is given by the following equation:

$$P_{T1} = \frac{R^C(K)}{K}, \quad K = |C| - 1 \tag{13}$$

where K is the number of first retrieved objects, $R^C(K)$ is the number of retrieved objects within the K-first, which are of the same class C with the query, and $|C|$ is the number of objects that belong to class C.

The PSO algorithm iteratively converges to a set of weights w_{CMVD}, w_{STT}, and w_{DSR} that maximize the average Tier-1 precision. This optimization procedure resulted in the following weights: $w_{CMVD} = 0.489, w_{STT} = 0.111$, and $w_{DSR} = 0.4$. Experiments presented in Section VII prove that the proposed optimized set of weights managed to significantly improve the retrieval accuracy of existing descriptors and dissimilarity metrics.

VI. Manifold Learning

In general, manifold learning is a technique that aims to identify a low-dimensional manifold, in which the data of a given dataset lie. Even if these data are initially represented as high dimensional vectors in Euclidean feature space, manifold learning achieves nonlinear dimensionality reduction. A manifold learning approach, based on *Laplacian eigenmaps* (LE) [23], is used in this paper to further improve the performance of 3-D object retrieval. It must be noted that LE was selected among several approaches, such as *locally linear embedding* (LLE) [38], *local regression global alignment* (LRGA) [39], and *manifold ranking-based retrieval* (MR) [40]. These methods were also tested, however, a detailed analysis of each method would be out of the scope of this paper.

Let us assume a dataset of $N3-D$ objects. In this dataset, CMVD, STT, and DSR descriptors are extracted for each 3-D object, adding also feature selection to produce more compact descriptors. Furthermore, dissimilarity measure optimization is applied, based on the methods described above.

At a pre-processing stage, an $N \times n$ matrix, $\mathbf{NN}$, is initially created, where each row i represents the n-nearest neighbors of the ith object. The n-nearest neighbors are determined by computing the one-to-all dissimilarity of object i with all N objects of the dataset, using the combined dissimilarity measure of (8), and sorting the results in ascending order.

Using the $\mathbf{NN}$ matrix, the LE-based algorithm creates a low-dimensional feature space, where all 3-D objects are represented as l-dimensional vectors. The algorithm consists of the following steps:

Step 1) Construct the graph G, by connecting nodes i and j with an edge, if object j is among k-nearest neighbors of object i.

Step 2) Produce the $N \times N$ adjacency matrix, $\mathbf{W}$, of G:

$$W_{ij} = \begin{cases} 1, & \text{if object } j \text{ belongs to } k \text{ nearest neighbors} \\ & \quad \text{of object } i \\ 0, & \text{otherwise.} \end{cases} \tag{14}$$

Step 3) Create an $N \times N$ diagonal matrix $H_{ii} = \sum_j W_{ij}$.

Step 4) Create an $N \times N$ Laplacian matrix $\mathbf{L} = \mathbf{H} - \mathbf{W}$.

Step 5) Solve the generalized eigenproblem $\mathbf{LY} = \lambda \mathbf{HY}$ to find the eigenvalues λ and the eigenvectors $\mathbf{Y}$ of $\mathbf{L}$.

Step 6) Sort eigenvalues in an ascending order and keep the l eigenvectors that correspond to the l-first eigenvalues (excluding the first one).

The l selected eigenvectors correspond to l-dimensions of the new low-dimensional feature space, where all database objects are mapped to low-dimensional points. In this feature space, semantically similar 3-D objects are placed close to each other, in terms of Euclidean distance, while objects of different semantic categories are far from each other.

VII. Experimental Results

The proposed method was experimentally evaluated using the SHREC 2009 [17] and SHREC 2011 [1] Generic Shape Benchmark datasets. The first dataset consists of 720 3-D models classified in 40 equally-sized categories. The dataset includes also 80 query models, 2 from each category. The second benchmark dataset consists of 1000 3-D models classified in 50 equally-sized categories. The 3-D object descriptors, in both datasets, were extracted using the CMVD, STT, and DSR methods.

To evaluate the proposed method in SHREC 2009 dataset, each 3-D model of the query set (80 objects) was used as query to retrieve the rest 720 models of the database and a rank list was produced for each query. Regarding the SHREC 2011 benchmark, each object from the database was used as query to retrieve the rest 999 objects. The retrieval performance, in both cases, was evaluated in terms of the well-known "precision-recall", where precision is the proportion of the retrieved models that are relevant to the query and recall is the proportion of relevant models in the entire database that are retrieved in the query.

A. Evaluation of the Rotation Estimation Method

In Fig. 3, some indicative results of rotation normalization using the proposed pose estimation method are depicted. In these categories of 3-D models (chairs, couches, animals) most of the rotation estimation methods fail to produce consistent results. The proposed method, however, achieves very accurate pose estimation even in these categories.

The proposed method was compared with the following state-of-the-art rotation estimation methods: 1) continuous PCA (CPCA) [4], 2) Rectilinearity measure combined with CPCA (CPCA-RECT) [7], and 3) Plane and Translational Symmetry (PT-Symmetry) [9]. In order to have a quantitative measure of their performance, all rotation estimation methods were applied as a preprocessing (pose estimation) step, followed by extraction of the same view-based rotation dependent descriptor. In the results presented below, the CMVD descriptor was used.

A comparison of the above methods, using SHREC 2009 and SHREC 2011 datasets, is given in Fig. 4, respectively. It

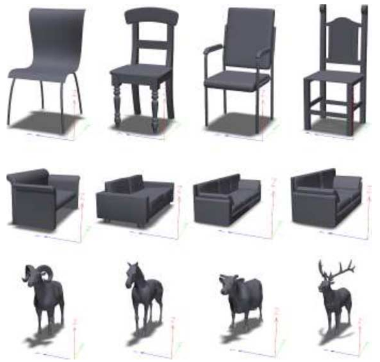

Fig. 3. Results of rotation normalization using the proposed method.

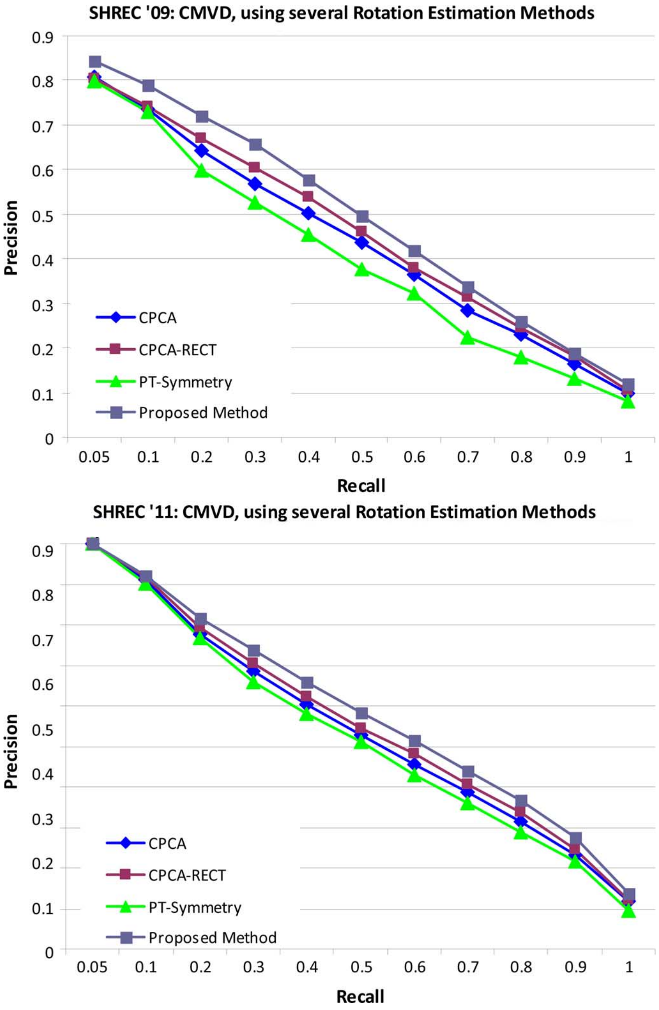

Fig. 4. Comparison of the proposed rotation estimation method with CPCA, CPCA-RECT, PT-Symmetry, using the SHREC '09 and SHREC '11 Generic Shape Benchmarks.

is clear that the proposed method achieves the best rotation estimation. In Table II, the percentages of perfect alignment (similar to what a human would select) for the 720 objects of SHREC 2009 Generic Shape Benchmark are given. The results of the proposed method are compared to the ones of CPCA, CPCA-RECT, and PT-Symmetry. It is clear that the proposed method outperforms all the others, producing better alignment results. The proposed approach achieves perfect alignment in 613 out of the 720 objects, while CPCA, PT-Symmetry, and CPCA-RECT achieve perfect alignment in 475, 510, and 565 objects, respectively. More specifically, PT-Symmetry and CPCA-RECT demonstrate inferior performance because they contain no information about the ordering of the principal axes. This leads to inconsistencies in principal axes in several models of the dataset. On the other hand, CPCA achieves better ordering of principal axes than PT-Symmetry and CPCA-RECT, but it produces worst results in specific categories. The method presented in this paper not only achieves better ordering of principal axes but it is also very robust in most of the categories of the dataset. Table II indicates also that there are cases where all methods fail to return an accurate pose. This is usually observed in 3-D objects with very irregular shapes, such as trees, where it is very hard to detect symmetries or principal axes. There are also specific models in the dataset, which are defective by design. These are outliers and cannot be aligned by any rotation estimation method.

TABLE II
METHODS THAT PARTICIPATED IN SHREC 2011
GENERIC SHAPE RETRIEVAL CONTEST

Method	Authors
BF-DSIFT	T. Furuya and R. Ohbuchi
Expressive Visual Features (EVF)	K. Goto, T. Yanagimachi, T. Furuya and R. Ohbuchi
PANORAMA	P. Papadakis, I. Pratikakis, T. Theoharis
	and S. Perantonis
ENHANCED-PAN	B. Gong, J. Liu, and X. Tang

B. Evaluation of Feature Selection

Feature selection is a supervised learning technique since it requires knowledge of the database classification scheme. In order to assess the performance of the four feature selection methods (CFS, Chi-Square, AdaBoost, and Mutual Information-based feature selection), which were adopted in this paper, an appropriately selected 3-D object dataset was used as training set. Then, the outcome of feature selection was used for testing in SHREC 2009 and SHREC 2011 benchmarks. The training set was constructed by selecting 3-D objects from both SHREC 2009 and SHREC 2011 datasets, resulting in a total of 400 models classified in 20 equally-sized categories. All of these 20 categories are included in both datasets.

The training set was used to select the optimal features from CMVD and STT descriptors. While in STT the entire descriptor vector was used, in CMVD, the descriptor vector of only one view per object was used. More specifically, the most representative view for each class was manually selected.

The results of feature selection in both datasets are presented in Fig. 5. In the case of STT, all feature selection methods achieved very high compression, since the dimensionality of the 2-D Zernike-based descriptors was reduced from 1080 to 122 with CFS, 232 with Chi-Square, 275 with Mutual Information and 318 with AdaBoost, while the dimensionality of 2-D Krawtchouk-based descriptors was reduced from 1080 to 101 with CFS, 230 with Chi-Square, 288 with Mutual Information and 340 with AdaBoost. Moreover, the performance of STT was improved by using CFS in both 2-D Zernike and 2-D Krawtchouk-based descriptors (note that only the CFS-based

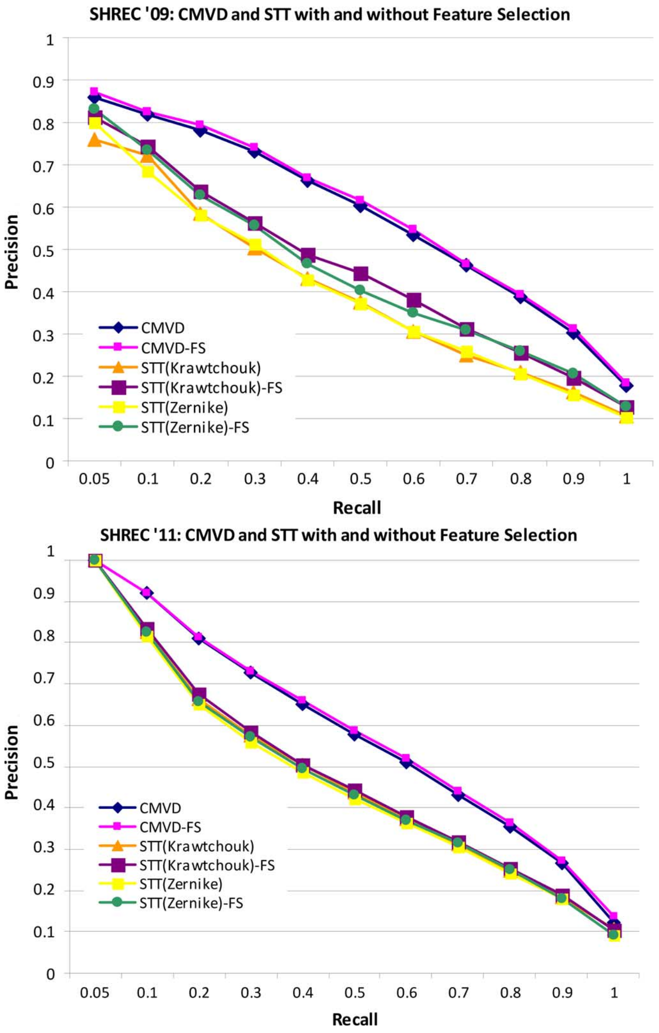

Fig. 5. Results of feature selection for CMVD and STT descriptors, using SHREC ’09 and SHREC ’11 Benchmarks.

feature selection is depicted in Fig. 5, which achieved better performance). Similar improvements are observed in the performance of CMVD as well. However, in this case, not a very high compression was achieved, since 60 (CFS), 64 (Chi-Square), 92 (Mutual Information) and 160 (Adaboost) features were selected out of the 212 descriptors per view (note that only the Chi-square-based feature selection is depicted in Fig. 5, which achieved better performance). The compression may be further improved if, instead of only the representative views, combinations of views per 3-D object are used for feature selection. In this case, different number of features may be selected for each view. This is a more complicated problem and it remains a challenge for future work.

Regarding the DSR descriptor, feature selection did not achieve significant reduction. This is actually not very surprising, taking into account the fact that DSR is already a very compact descriptor. More specifically, DSR is the combination of three separate descriptors: Depth Buffer, which is a view-based descriptor with dimensionality 186, Silhouette descriptor with dimensionality 150 and Ray-based descriptor, a transform-based descriptor with dimensionality 136. The constituting descriptor vectors have been already optimized in order to achieve the best performance with the least possible features [21].

C. Performance Evaluation of the Various Dissimilarity Metrics

In Fig. 6, a comparison of the various dissimilarity metrics for each low-level descriptor is presented. For both functionals of STT, namely Zernike moments and Krawtchouk moments, the following distance metrics returned the best results: X*, Canberra, SQRT(L1),* X^2*, KLD, Jensen* and $L1$. Among these, both X and *Canberra*[46] distances achieved by far the best retrieval performance. It must be noted that since STT descriptors are all positive values, the formulae of X and *Canberra* distance in Table I return exactly the same result. This is the reason why *Canberra* distance is not depicted in the diagram. It is also worth to mention that in [24], where STT was introduced, the $L1$ distance was used as a distance measure. However, other distance measures proved to be more efficient than the metric that was initially proposed.

Regarding CMVD and DSR descriptors, almost all distance metrics returned acceptable results. The Squared L1 distance ($SQRT(L1)$) was slightly better in CMVD, while for DSR, the X^2 distance returned the best retrieval results. It is worth to mention that for each low-level descriptor, different distance metrics achieved the best retrieval accuracy. As an example, the X distance, which achieved the best performance for STT descriptor, was ranked last when applied to the DSR descriptor.

D. Evaluation of Weight Optimization and Manifold Learning

In Fig. 7, the precision-recall diagrams of each individual low-level descriptor (CMVD, STT, DSR), along with their combinations using weight optimization and manifold learning, are presented for SHREC 2009 and SHREC 2011 databases, respectively. It is clear that by merging the single-descriptor dissimilarities into one unified distance measure, the retrieval accuracy of each separate descriptor can be significantly improved. Finally, it is obvious in both datasets that by applying manifold learning to the combined dissimilarity measure the best retrieval results are produced. The improvement is more visible for larger recall values, which means that the system keeps retrieving a high percentage of relevant objects even to higher positions of the ranked list.

E. Comparison With Other State-of-the-Art Methods

The 3-D object retrieval framework presented in this paper was compared with similar state-of-the-art approaches. Although there are plenty of methods available in the literature, we selected only the ones that achieved the best performance in SHREC 2009 and SHREC 2011 Generic Shape Retrieval contests, since they can provide a sufficiently representative sample.

A preliminary version of the proposed framework was ranked first among five state-of-the-art methods that participated in SHREC 2011 Generic Shape Retrieval contest. In the version presented in SHREC 2011 results, the weights of the combined dissimilarity measure have been heuristically determined, while in the current version presented in this paper,

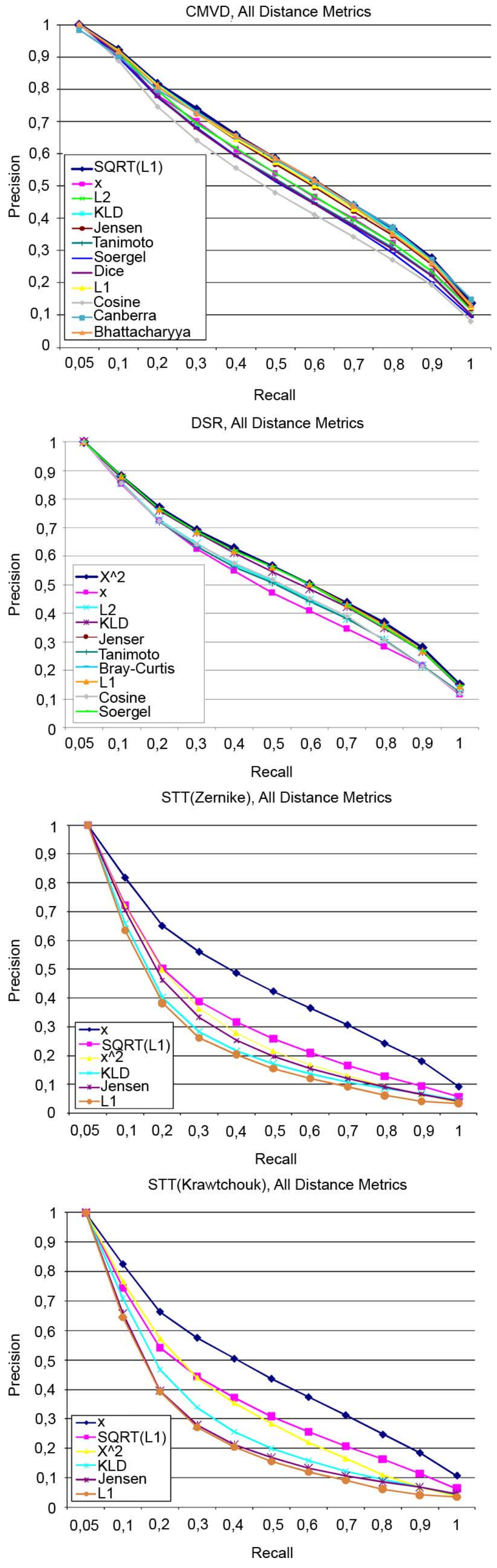

Fig. 6. Evaluation of the various dissimilarity metrics for CMVD, STT, and DSR descriptors.

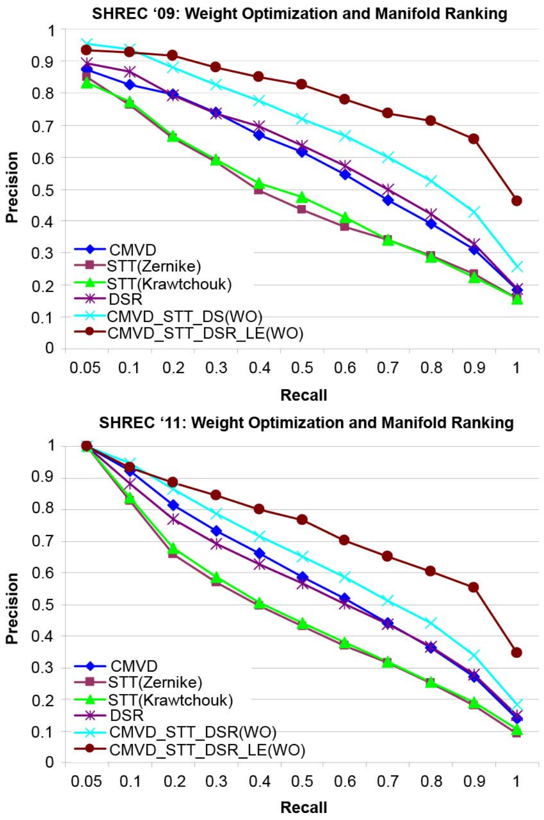

Fig. 7. Evaluation of weight optimization and manifold learning, using SHREC '09 and SHREC '11 Benchmarks.

the PSO optimization method has been applied to estimate the optimal weights. In Fig. 8, both versions are depicted, where it is obvious that the performance is further improved if an appropriate optimization method is used. The rest four methods that participated in SHREC 2011 are presented in Table III.

Although the proposed method did not take part in SHREC 2009 Generic Shape Retrieval Contest, it is depicted in a common precision-recall diagram along with the participating methods, in Fig. 8. The methods are given in Table IV for the sake of completeness.

From the precision-recall diagrams, it is obvious that if the proposed framework had taken part in SHREC 2009 contest, it would have been ranked among the first three methods.

F. Computational Issues

A method for 3-D object retrieval should not only demonstrate high retrieval accuracy but also be adequately fast, which makes it appropriate for online applications. Therefore, a major aspect in the proposed framework is the computation time. In terms of computational efficiency, we focus on the following subparts of the proposed framework: rotation estimation, descriptor extraction, similarity matching and manifold ranking. It is of no interest to present computation times for feature selection and dissimilarity metric optimization, since these processes are performed only once (offline) and the optimal values are eventually kept.

In Table V, the average computation times for rotation estimation, descriptor extraction, similarity matching and manifold ranking processes are presented. These times were obtained by applying the proposed framework to the 1000 3-D models of the SHREC 2011 dataset. More specifically, rotation estimation and descriptor extraction times refer to the average time required

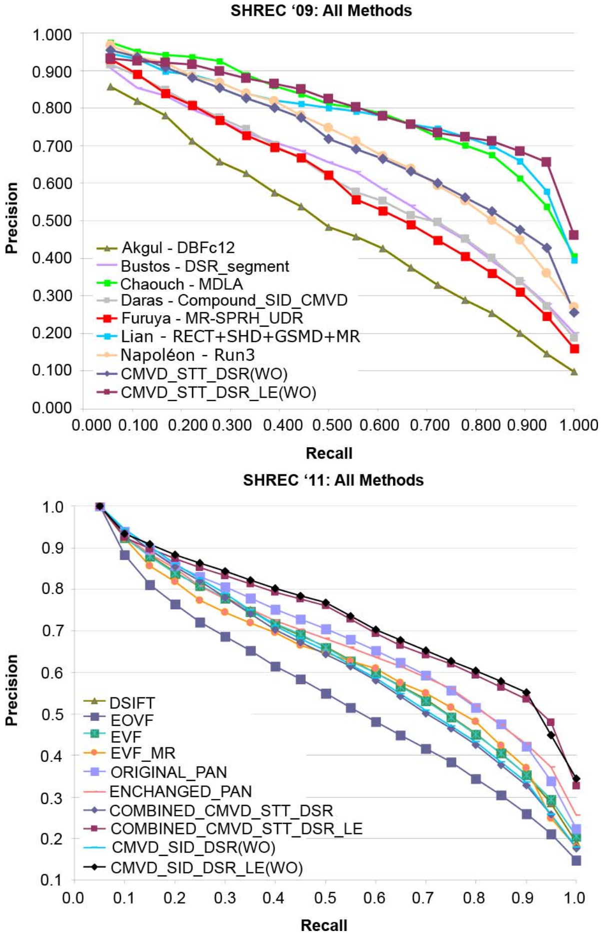

Fig. 8. Comparison of the proposed framework with state of the art, using SHREC '09 and SHREC '11 Benchmarks.

TABLE III
METHODS THAT PARTICIPATED IN SHREC 2009
GENERIC SHAPE RETRIEVAL CONTEST

Method	Authors
MDLA	M. Chaouch and A. Verroust-Blondet
A Composite Shape Descriptor	Z. Lian, P. L. Rosin and X. Sun
Combined SID-CMVD	P. Daras, A. Mademlis and A. Axenopoulos
MCC	T. Napoléon
Density-Based Framework (DBF)	C. Akgül, B. Sankur and Y. Yemez
BF-SIFT	T. Furuya, M. Tezuka and R. Ohbuchi
Local-Global Features from DESIRE	B. Bustos, S. Kreft, T. Schreck and M.Walter

TABLE IV
AVERAGE COMPUTATION TIMES FOR ROTATION ESTIMATION, DESCRIPTOR EXTRACTION, SIMILARITY MATCHING, AND MANIFOLD RANKING PROCESSES

Action		Processing time (msec)
Rotation Estimation	CPCA	102
	Rectilinearity	4336
	Proposed Method	645
Descriptor Extraction	CMVD	1810
	STT	4884
	DSR	1290
Similarity Matching	No Feature Selection	1982
	Feature Selection	441
Manifold Ranking (offline)		21382
Manifold Ranking (online)		16

for processing one 3-D object of the dataset, while similarity matching time corresponds to the time needed to match an object of the dataset (which is used as query) with the remaining 999 objects. Manifold ranking time is divided into two processes: offline and online. The former refers to the time for creating the new low-dimensional feature space, where all database objects are mapped to low-dimensional points, using Laplacian eigenmaps. The latter refers to the time needed to match an object of the dataset (which is used as query) with the remaining 999 objects using the low-dimensional descriptors.

From Table IV useful conclusions can be drawn. First of all, the proposed rotation estimation method is more time consuming than CPCA but it is much faster than the method based on rectilinearity. The reason is that the latter uses an optimization method to find the best rotation around x, y, and z axes, which is quite slow. However, in the proposed method, rotation around x, y, and z axes is applied only for zero plane symmetries, while in case that symmetries are detected, CPCA is kept. In general, the proposed method achieves more stable rotation normalization than CPCA, while the processing time is less than one second, which is acceptable for the pre-processing task.

Regarding descriptor extraction, STT proved to be more time consuming than CMVD and DSR, while it contributes less to the improvement of retrieval performance. Therefore, in real-time search applications, only CMVD and DSR can be used. It is also worth to mention that feature selection achieves significant reduction of the similarity matching time (5 times faster), while retrieval performance is not degraded. Finally, manifold ranking not only achieves better retrieval performance but it also reduces significantly the dissimilarity matching time thanks to the (non-linear) dimensionality reduction. The times were obtained using a PC with a dual-core 2.4-GHz processor and 8 GB of RAM.

VIII. DISCUSSION AND CONCLUSIONS

In this paper, a novel framework for 3-D object search and retrieval was presented. The paper's main focus was not to implement a new highly complex algorithm for descriptor extraction but to investigate how the contribution of various factors, such as rotation normalization, feature selection, descriptor combination, weight optimization and manifold learning, can improve the performance of existing descriptor extraction algorithms. With respect to rotation normalization, the paper introduced a new method for accurate alignment of 3-D objects, which combines the well-known CPCA with two intuitive criteria, the plane reflection symmetry and rectilinearity. Regarding feature selection, several techniques, which have been already used in machine learning, were applied in 3-D object retrieval. The paper provided also an exhaustive list of the most well-known dissimilarity metrics, as well as an approach for dissimilarity measure optimization. The retrieved results were further improved using a manifold learning method based on Laplacian Eigenmaps. Experiments performed on two 3-D object benchmark datasets, namely the SHREC 2009 and SHREC 2011 Generic Shape Benchmarks, demonstrated the superiority and the efficiency of the proposed framework comparing with other state-of-the-art approaches.

TABLE V
SET OF DISSIMILARITY METRICS TESTED AND EVALUATED IN THIS PAPER

Name	Equation	Name	Equation
Manhattan Distance (L1)	$d=\sum_{i=1}^{N}\left\|D^A(i)-D^B(i)\right\|$	Squared L1 Distance (SQRT(L1))	$d=\sum_{i=1}^{N}\left\|D^A(i)-D^B(i)\right\|^{\frac{1}{2}}$
Euclidean Distance (L2)	$d=\sqrt{\sum_{i=1}^{N}\left(D^A(i)-D^B(i)\right)^2}$	Kullback-Leibler divergence (KLD)	$d=\sum_{i=1}^{N}(D^B(i)-D^A(i))\boldsymbol{ln}\frac{D^B(i)}{D^A(i)}$
Cosine Coefficient	$d=\frac{\sum_{i=1}^{N}D^A(i)\cdot D^B(i)}{\sqrt{\sum_{i=1}^{N}\left(D^A(i)\right)^2\cdot\sum_{i=1}^{N}\left(D^B(i)\right)^2}}$	Jensen Distance	$d=\sum_{i=1}^{N}\left[\frac{D^A(i)\boldsymbol{ln}D^A(i)+D^B(i)\boldsymbol{ln}D^B(i)}{2}-\left(\frac{D^A(i)+D^B(i)}{2}\right)\boldsymbol{ln}(\frac{D^A(i)+D^B(i)}{2})\right]$
Tanimoto Coefficient	$d=\frac{\sum_{i=1}^{N}D^A(i)\cdot D^B(i)}{\sum_{i=1}^{N}\left(D^A(i)\right)^2+\sum_{i=1}^{N}\left(D^B(i)\right)^2-\sum_{i=1}^{N}D^A(i)\cdot D^B(i)}$	Bray-Curtis Coefficient	$d=\frac{\sum_{i=1}^{N}\left\|D^A(i)-D^B(i)\right\|}{2+\sum_{i=1}^{N}\left\|D^A(i)+D^B(i)\right\|}$
Soergel Distance	$d=\frac{\sum_{i=1}^{N}\left\|D^A(i)-D^B(i)\right\|}{\sum_{i=1}^{N}\boldsymbol{max}(D^A(i),D^B(i))}$	Dice Coefficient	$d=\frac{2\cdot\sum_{i=1}^{N}D^A(i)\cdot D^B(i)}{\sum_{i=1}^{N}\left(D^A(i)\right)^2+\sum_{i=1}^{N}\left(D^B(i)\right)^2}$
X-Distance	$d=\frac{1}{2}\cdot\sum_{i=1}^{N}\frac{\left\|D^A(i)-D^B(i)\right\|}{\left(D^A(i)+D^B(i)\right)}$	X^2-Distance	$d=\frac{1}{2}\cdot\sum_{i=1}^{N}\frac{\left(D^A(i)-D^B(i)\right)^2}{\left(D^A(i)+D^B(i)\right)}$
Canberra Distance	$d=\sum_{i=1}^{N}\frac{\left\|D^A(i)-D^B(i)\right\|}{\left\|D^A(i)\right\|+\left\|D^B(i)\right\|}$	Bhattacharyya Distance	$d=\sqrt{1-\frac{1}{\sqrt{\overline{D}^A\overline{D}^B N^2}}\sum_{i=1}^{N}\sqrt{D^A(i)\cdot D^B(i)}}$

Based on the above conclusions, this study is expected to provide a useful reference for further research, as well as to contribute to shaping the future research directions in 3-D object retrieval. The main outcome of this study is that the combination of multiple 3-D object descriptors can achieve better retrieval accuracy than a single descriptor vector alone, no matter how efficient this descriptor is. Thus, research should focus not only on the investigation of the optimal descriptor but also on the appropriate combination of low-level descriptors as well as on the selection of the best features and matching metrics. This means that as soon as a new promising method appears, the next step is to combine it with other state-of-the-art approaches and incorporate it into a unified 3-D shape matching framework. Provided that the new method belongs to transform-based or view-based methods and describes the global features of a 3-D object, it can be easily integrated to the framework proposed in this paper, as follows: find the optimal dissimilarity metric for this descriptor and normalize it to a common range of values (e.g., between 0 and 1); the overall dissimilarity is the weighted sum of dissimilarities of multiple descriptors, where the weights should be optimized with a method similar to the one presented in Section V.

Apart from transform-based and view-based methods, graph-based methods can be also used for global-shape 3-D object retrieval tasks. However, graph-based methods may usually require complex matching schemes, which makes their combination with transform and view-based methods problematic. Another limitation is that most of the graph-based methods, although they work well with articulated objects, they have very limited accuracy in categories of 3-D models that have no articulation (e.g., buildings, furniture, cars, cups, etc.). On the other hand, in the case of non-rigid and partial 3-D shape retrieval tasks, graph-based (topology-based) methods should be preferred instead of transform-based and view-based ones. It is worth to mention that in SHREC'11 Track: Retrieval on Non-rigid 3-D Watertight Meshes [49], the best retrieval performance is achieved by a framework that combines two different methods, the Spectral Decomposition of the Geodesic Distance Matrix [52] and the Scale Invariant Feature Transform for meshes (meshSIFT) [53].

Finally, for future work, appropriate schemes for Relevance Feedback will be investigated, which is expected to further improve the retrieval performance of the proposed framework.

REFERENCES

[1] H. Dutagaci, A. Godil, P. Daras, A. Axenopoulos, G. Litos, S. Manolopoulou, K. Goto, T. Yanagimachi, Y. Kurita, S. Kawamura, T. Furuya, R. Ohbuchi, B. Gong, J. Liu, and X. Tang, "SHREC'11 track: Generic shape retrieval," in *Proc. 4th Eurographics Workshop 3-D Object Retrieval (3-DOR 2011)*, Llandudno, U.K., Apr. 10, 2011.

[2] T. P. Vanamali, A. Godil, H. Dutagaci, T. Furuya, Z. Lian, R. I. Ohbuchi, M. Daoudi, and T. Schreck, M. Spagnuolo, I. Pratikakis, and R. Veltkamp, Eds., "SHREC'10 track: Generic 3D warehouse," in *Proc. Eurographics/ACM SIGGRAPH Symp. 3D Object Retrieval*, 2010.

[3] E. Paquet and M. Rioux, "Nefertiti: A query by content system for three-dimensional model and image databases management," *Image Vision Comput.*, vol. 17, no. 2, pp. 157–166, 1999.

[4] D. V. Vranic, D. Saupe, and J. Richter, "Tools for 3D-object retrieval: Karhunen-loeve transform and spherical harmonics," in *Proc. 2001 IEEE 4th Workshop Multimedia Signal Processing*, 2001, pp. 293–298.

[5] P. Daras and A. Axenopoulos, "A 3D shape retrieval framework supporting multimodal queries," *Springer Int. J. Comput. Vis.*, Jul. 2009.

[6] J. T. Pu and K. Ramani, "An automatic drawing-like view generation method from 3D models," in *Proc. ASME IDETC/CIE 2005, 25th Computers and Information in Engineering (CIE) Conf.*, Long Beach, CA, 2005, pp. 301–320.

[7] Z. Lian, P. L. Rosin, and X. Sun, "Rectilinearity of 3D meshes," *Int. J. Comput. Visi.*, vol. 89, no. 2-3, pp. 130–151.

[8] J. Podolak, P. Shilane, A. Golovinskiy, S. Rusinkiewicz, and T. Funkhouser, "A planar-reflective symmetry transform for 3D shapes," *ACM Trans. Graph. (Proc. SIGGRAPH)*, vol. 25, no. 3, Jul. 2006.

[9] M. Chaouch and A. Verroust-Blondet, "Alignment of 3D models," *Graph. Models*, vol. 71, no. 2, pp. 63–76, Mar. 2009.

[10] H.-P. Kriegel, P. Kroeger, Z. Mashael, M. Pfeifle, M. Poetke, and T. Seidl, "Effective similarity search on voxelized cad objects," in *Proc. IEEE 8th Int. Conf. Database Systems for Advanced Applications*, Washington, DC, 2003.

[11] A. Mademlis, P. Daras, A. Axenopoulos, D. Tzovaras, and M. G. Strintzis, "Combining topological and geometrical features for global and partial 3D shape retrieval," *IEEE Trans. Multimedia*, vol. 10, no. 5, pp. 819–831, 2008.

[12] T. Tung and F. Schmitt, "The augmented multiresolution Reeb graph approach for content-based retrieval of 3D shapes," *Int. J. Shape Model.*, vol. 11, no. 1, pp. 91–120, 2005.

[13] P. Papadakis, I. Pratikakis, S. Perantonis, and T. Theoharis, "Efficient 3D shape matching and retrieval using a concrete radialized spherical projection representation," *Pattern Recognit.*, vol. 40, no. 9, pp. 2437–2452, 2007.

[14] D. V. Vranic, "An improvement of rotation invariant 3D shape descriptor based on functions on concentric spheres," in *Proc. IEEE Int. Conf. Image Processing*, Sep. 2003, pp. 757–760.

[15] R. Ohbuchi, K. Osada, T. Furuya, and T. Banno, "Salient local visual features for shape-based 3D model retrieval," in *Proc. IEEE Int. Conf. Shape Modeling and Applications (SMI 2008)*, 2008, pp. 93–102.

[16] M. Chaouch and A. Verroust-Blondet, "3D model retrieval based on depth line descriptor," in *Proc. IEEE Int. Conf. Multimedia & Expo (ICME'07)*, Beijing, China, Jul. 2007.

[17] C. Akgul, A. Axenopoulos, B. Bustos, M. Chaouch, P. Daras, H. Dutagaci, T. Furuya, A. Godil, S. Kreft, Z. Lian, T. Napoleon, A. Mademlis, R. Ohbuchi, P. L. Rosin, B. Sankur, T. Schreck, X. Sun, M. Tezuka, Y. Yemez, A. Verroust-Blondet, and M. Walter, "SHREC 2009—generic shape retrieval contest," in *Proc. 30th Int. Conf. EUROGRAPHICS 2009, Workshop 3D object retrieval*, Munich, Germany, Mar. 2009.

[18] M. Dash and H. Liu, "Feature selection for classification," *Intell. Data Anal.*, vol. 1, pp. 131–156, 1997.

[19] M. A. Hall, "Correlation-based feature selection for discrete and numeric class machine learning," in *Proc. 17th Int. Conf. Machine Learning*, 2000, pp. 359–366.

[20] H. Liu and R. Setiono, "A probabilistic approach to feature selection—a filter solution," in *Proc. Int. Conf. Machine Learning*, 1996, pp. 319–327.

[21] D. Vranic, "DESIRE: A composite 3D-shape descriptor," in *Proc. 2005 IEEE Int. Conf. Multimedia and Expo (ICME 2005)*, Amsterdam, The Netherlands, Jul. 6–9, 2005, pp. 962–965.

[22] B. Bustos, D. Keim, D. Saupe, T. Schreck, and D. Vranic, "Automatic selection and combination of descriptors for effective 3D similarity search," in *Proc. IEEE Int. Workshop Multimedia Content-Based Analysis and Retrieval*, 2004.

[23] R. Ohbuchi and J. Kobayashi, "Unsupervised learning from a corpus for shape-based 3D model retrieval," in *Proc. ACM MIR*, Santa Barbara, CA, 2006.

[24] D. Zarpalas, P. Daras, A. Axenopoulos, D. Tzovaras, and M. G. Strintzis, "3D model search and retrieval using the spherical trace transform," *EURASIP J. Adv. Signal Process.*, vol. 2007, no. 23912, p. 14, 2007.

[25] P. Minovic, S. Ishikawa, and K. Kato, "Symmetry identification of a 3-D object represented by octree," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 5, pp. 507–514, 1993.

[26] C. Sun and J. Sherrah, "3D symmetry detection using the extended Gaussian image," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 2, pp. 164–168, 1997.

[27] A. Martinet, C. Soler, N. Holzschuch, and F. X. Sillion, "Accurate detection of symmetries in 3D shapes," *ACM Trans. Graph.*, vol. 25, no. 2, pp. 439–464, 2006.

[28] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz, "Symmetry descriptors and 3D shape matching," in *Proc. Symp. Geometry Processing*, 2004.

[29] N. L. Jonhson, S. Kotz, and N. Balakrishnan*, Continuous Univariate Distributions*, 2nd ed. New York: Wiley, 1994, vol. 1, ch. 18.

[30] Y. Rubner, C. Tomasi, and L. J. Guibas, "The earth mover's distance as a metric for image retrieval," *Int. J. Comput. Vis.*, vol. 40, no. 2, pp. 9–121, 2000.

[31] H. Ling and K. Okada, "Diffusion distance for histogram comparison," in *Proc. CVPR*, 2006, vol. 1, no. 2, pp. 246–253.

[32] [Online]. Available: http://www.swarmintelligence.org/tutorials.php.

[33] A. Axenopoulos, G. Litos, and P. Daras, "3D model retrieval using accurate pose estimation and view-based similarity," in *Proc. 1st ACM Int. Conf. Multimedia Retrieval (ICMR2011)*, Trento, Italy, Apr. 17–20, 2011.

[34] I. T. Jolliffe*, Principal Component Analysis*. New York: Springer-Verlag, 1986.

[35] R.-S. Marko and K. Igor, "An adaptation of relief for attribute estimation in regression," in *Proc. 14th Int. Conf. Machine Learning*, 1997, pp. 296–304.

[36] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," *Mach. Learn.*, vol. 46, pp. 389–422, 2002.

[37] G. Lupatini, C. Saraceno, and R. Leonardi, "Scene break detection: A comparison," in *Proc. 8th Int. Workshop Research Issues in Data Engineering*, Feb. 1998, pp. 34–41.

[38] L. K. Saul and S. T. Roweis, "Think globally, fit locally: Unsupervised learning of low dimensional manifolds," *J. Mach. Learn. Res.*, 2003.

[39] Y. Yang, D. Xu, F. Nie, J. Luo, and Y. Zhuang, "Ranking with local regression and global alignment for cross media retrieval," in *Proc. ACM MM*, Beijing, China, 2009.

[40] J. R. He, M. J. Li, H. J. Zhang, H. H. Tong, and C. S. Zhang, "Manifold-ranking based image retrieval," in *Proc. ACM MM*, New York, 2004.

[41] F. Yang and B. Leng, "OFS: A feature selection method for shape-based 3D model retrieval," in *Proc. IEEE Int. Conf. Computer-Aided Design and Computer Graphics*, 2007.

[42] B. Bonev, F. Escolano, D. Giorgi, and S. Biasotti, "High-dimensional spectral feature selection for 3D object recognition based on Reeb graphs," *Lecture Notes in Computer Science*, 2010.

[43] S. Marini, G. Patane, M. Spagnuolo, and B. Falcidieno, "Feature selection for enhanced spectral shape comparison," in *Proc. Eurographics Workshop 3D Object Retrieval*, 2010.

[44] N. Leonenko, L. Pronzato, and V. Savani, "A class of renyi information estimators for multidimensional densities," *Ann. Statist.*, vol. 36, no. 5, pp. 2153–2182, 2008.

[45] S. Goodall, P. Lewis, and K. Martinez, "3-D shape descriptors and distance metrics for content-based artefact retrieval," *Storage Retriev. Meth. Appl. Multimedia*, 2005.

[46] R. A. Johnson and D. W. Wichern*, Applied Multivariate Statistical Analysis*. Englewood Cliffs, NJ: Prentice-Hall, 1998, pp. 226–235.

[47] M. Hilaga, Y. Shinagawa, T. Kohmura, and T. L. Kunii, "Topology matching for fully automatic similarity estimation of 3D shapes," in *Proc. ACM SIGGRAPH*, 2001, pp. 203–212.

[48] A. Tal and E. Zuckerberger, "Mesh retrieval by components," in *Proc. Int. Conf. Computer Graphics Theory and Applications*, 2006, pp. 142–149.

[49] Z. Lian, A. Godil, B. Bustos, M. Daoudi, J. Hermans, S. Kawamura, Y. Kurita, G. Lavoué, H. V. Nguyen, R. Ohbuchi, Y. Ohkita, Y. Ohishi, F. Porikli, M. Reuter, I. Sipiran, D. Smeets, P. Suetens, H. Tabia, and D. Vandermeulen, "SHREC'11 track: Shape retrieval on non-rigid 3D watertight meshes," in *Proc. Eurographics Workshop 3D Object Retrieval 2011*, 2011, pp. 79–88.

[50] M. Novotni, P. Degener, and R. Klein, Correspondence Generation and Matching of 3D Shape Subparts, Friedrich-Wilhelms-Universität Bonn, 2005, Tech. Rep. CG-2005–2,ISSN 1610–8892.

[51] T. Funkhouser and P. Shilane, "Partial matching of 3D shapes with priority-driven search," in *Proc. Eurographics Symp. Geometry Processing*, 2006, pp. 131–142.

[52] D. Smeets, T. Fabry, J. Hermans, D. Vandermeulen, and P. Suetens, "Isometric deformation modelling for object recognition," in *Proc. CAIP'09*, 2009, pp. 757–765.

[53] C. Maes, T. Fabry, J. Keustermans, D. Smeets, P. Suetens, and D. Vandermeulen, "Feature detection on 3D face surfaces for pose normalisation and recognition," in *Proc. BTAS'10*, 2010.

[54] M.-C. Chang and B. B. Kimia, "Measuring 3D shape similarity by graph-based matching of the medial scaffolds," *Comput. Vis. Image Understand.*, vol. 115, no. 5, May 2011.

Petros Daras (M'07) was born in Athens, Greece, in 1974. He received the Diploma degree in electrical and computer engineering, the M.Sc. degree in medical informatics, and the Ph.D. degree in electrical and computer engineering, all from the Aristotle University of Thessaloniki, Thessaloniki, Greece, in 1999, 2002, and 2005, respectively.

He is a Researcher Grade C, at the Informatics and Telematics Institute (ITI) of the Centre for Research and Technology Hellas (CERTH). His main research interests include search, retrieval and recognition of 3-D objects, 3-D object processing, medical informatics applications, medical image processing, 3-D object watermarking, and bioinformatics. He serves as a reviewer/evaluator of European projects and he is a key member of the IEEE MMTC 3-DRPC IG.

Apostolos Axenopoulos was born in Thessaloniki, Greece, in 1980. He received the Diploma degree in electrical and computer engineering and the M.S. degree in advanced computing systems from Aristotle University of Thessaloniki, Thessaloniki, Greece, in 2003 and 2006, respectively. Currently, he is pursuing the Ph.D. degree in the Computer and Communication Engineering Department, University of Thessaly.

He has been an Associate Researcher at the Informatics and Telematics Institute (ITI) of Centre for Research and Technology—Hellas (CERTH) since November 2003. His main research interests include 3-D object indexing, content-based search and retrieval, and bioinformatics.

Georgios Litos is pursuing the degree in computer science at the Computer Science Department of the Hellenic Open University.

He was born in Naoussa, Greece, in 1973. He has ten years of professional experience in software development related to computer imaging, compression, communications, and robotics with applications to video, audio, compression, and health. He has been a Research Assistant at the Informatics and Telematics Institute of the Centre for Research and Technology—Hellas (CERTH) since January 2000. His research interests include internet applications, content management systems, multimedia and image processing, databases, e-commerce, knowledge, and multimedia systems technologies. He has participated in several research projects funded by the European Union and the Greek Secretariat of Research and Technology.