

## Research Article

# 3D Model Search and Retrieval Using the Spherical Trace Transform

Dimitrios Zarpalas,<sup>1,2</sup> Petros Daras,<sup>1,2</sup> Apostolos Axenopoulos,<sup>1,2</sup>  
Dimitrios Tzouvaras,<sup>1,2</sup> and Michael G. Strintzis<sup>1,2</sup>

<sup>1</sup>Information Processing Laboratory, Electrical and Computer Engineering Department, Aristotle University of Thessaloniki, Thessaloniki 54006, Greece

<sup>2</sup>Informatics and Telematics Institute, 1st km Thermi-Panorama Road, P.O.Box 361, Thermi-Thessaloniki 57001, Greece

Received 31 January 2006; Accepted 22 June 2006

Recommended by Ming Ouhyoung

This paper presents a novel methodology for content-based search and retrieval of 3D objects. After proper positioning of the 3D objects using translation and scaling, a set of functionals is applied to the 3D model producing a new domain of concentric spheres. In this new domain, a new set of functionals is applied, resulting in a descriptor vector which is completely rotation invariant and thus suitable for 3D model matching. Further, weights are assigned to each descriptor, so as to significantly improve the retrieval results. Experiments on two different databases of 3D objects are performed so as to evaluate the proposed method in comparison with those most commonly cited in the literature. The experimental results show that the proposed method is superior in terms of precision versus recall and can be used for 3D model search and retrieval in a highly efficient manner.

Copyright © 2007 Dimitrios Zarpalas et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

## 1. INTRODUCTION

With the general availability of 3D digitizers, scanners and the technology innovation in 3D graphics and computational equipment, large collections of 3D graphical models can be readily built up for different applications [1], that is, in CAD/CAM, games design, computer animations, manufacturing, and molecular biology. For example, a high number of new 3D structures of molecules have been stored in the worldwide repository Protein Data Bank (PDB) [2], where the number of the 3D molecular structure data increases rapidly, currently exceeding 24 000. For such large databases, the method whereby 3D models are sought merits careful consideration. The simple and efficient query-by-content approach has, up to now, been almost universally adopted in the literature. Any such method, however, must first deal with the proper positioning of the 3D models. The two prevalent in the literature methods for the solution to this problem seek either:

- (i) *pose normalization*: models are first placed into a canonical coordinate frame (normalizing for translation, scaling, and rotation), then, the best measure

of similarity is found comparing the extracted feature vectors; or

- (ii) *descriptor invariance*: models are described in a transformation invariant manner, so that any transformation of a model will be described in the same way, and the best measure of similarity is obtained at any transformation.

### 1.1. Background and related work

#### 1.1.1. Pose normalization

Most of the existing methods for 3D content-based search and retrieval of 3D models are applied following their placement into a canonical coordinate frame.

In [3] a fast querying-by-3D-model approach is presented, where the descriptors are chosen so as to mimic the basic criteria that humans use for the same purpose. More specifically, the specific descriptors that are extracted from the input model are the geometrical characteristics of the 3D objects included in the VRML such as the angles and edges that describe the outline of the model. Ohbuchi et al [4] employ shape histograms that are discretely parameterized

along the principal axes of inertia of the model. The three shape histograms used are the moment of inertia about the axis, the average distance from the surface to the axis, and the variance of the distance from the surface to the axis. Osada et al. [5, 6] introduce and compare shape distributions, which measure properties based on distance, angle, area, and volume measurements between random surface points. They evaluate the similarity between the objects using a metric that measures distances between distributions.

In [7] an approach that measures the similarity among 3D models by visual similarity is proposed. The main idea is that if two 3D models are similar, they also look similar from all viewing angles. Thus, one hundred projections of an object are encoded both by Zernike moments and Fourier descriptors as characteristic features to be used for retrieval purposes.

In [8, 9] the authors present a method where the descriptor vector is obtained by forming a complex function on the sphere. Then, the fast Fourier transform (FFT) is applied on the sphere and Fourier coefficients for spherical harmonics are obtained. The absolute values of the coefficients form the descriptor vector.

In [10] a 3D search and retrieval method based on the generalized radon transform (GRT) is proposed. Two forms of the GRT are implemented: (a) the radial integration transform (RIT), which integrates the 3D model's information on lines passing through its center of mass and contains all the radial information of the model, and (b) the spherical integration transform (SIT), which integrates the 3D model's information on the surfaces of concentric spheres and contains all the spherical information of the model. Additionally, an approach for reducing the dimension of the descriptor vectors is proposed, providing a more compact representation (EnRIT), which makes the procedure for the comparison of two models very efficient.

The aforementioned methods are applied following model normalization. In general, models are normalized by using the center of mass for translation, the root of the average square radius for scaling, and the principal axes for rotation. While the methods for translation and scale normalization are robust for object matching [11], rotation normalization via PCA-alignment is not considered robust for many matching applications. This is due to the fact that PCA-alignment is performed by solving for the eigenvalues of the covariance matrix. This matrix captures only second-order model information, and the assumption when using PCA is that the alignment of higher frequency information is strongly correlated with the alignment of the second order components [12]. Further, PCA lacks any information about the direction (orientation) of each axis and finally, if the eigenvalues are equal, no unique set of principal axes can be extracted.

### 1.1.2. Descriptor invariance

Relatively few approaches for 3D-model retrieval have been reported in which pose estimation is unnecessary. Topology matching [13] is an interesting and intricate such technique, based on matching graph representations of 3D-objects.

However, the method is suitable only for certain types of models.

The MPEG-7 shape spectrum descriptor [14] is defined as the histogram of the shape index, calculated over the entire surface of a 3D object. The shape index gives the angular coordinate of a polar representation of the principal curvature vector, and it is implicitly invariant with respect to rotation, translation and scaling.

In [15] a web-based 3D search system is developed that indexes a large repository of computer graphics models collected from the web supports queries based on 3D sketches, 2D sketches, 3D models, and/or text keywords. For the shape-based queries, a new matching algorithm was developed that uses spherical harmonics to compute discriminating similarity measures without requiring model alignment. In [12] a tool for transforming rotation-dependent spherical and voxel shape descriptors into rotation invariant ones is presented. The key idea of this approach is to describe a spherical function in terms of the amount of energy it contains at different frequencies. The results indicate that the application of the spherical harmonic representation improves the performance of most of the descriptors.

Novotni and Klein presented the 3D "Zernike" moments in [16]. These are computed as a projection of the function defining the object onto a set of orthonormal functions within the unit ball; their work was an extension of the 3D Zernike polynomials, which were introduced by Canterakis [17]. From these, Canterakis has derived affine invariant features of 3D objects represented by a volumetric function.

In [18], a 3D shape descriptor was proposed, which is invariant to rotations of 90 degrees around the coordinate axes. This restricted rotation invariance is attained by a very coarse shape representation computed by clustering point clouds. Since the normalization step is omitted, if an object is rotated around an axis by a different angle (e.g., by 45 degrees), the feature vector alters significantly.

In this paper a novel framework of rotation invariant descriptors is constructed without the use of rotation normalization. An efficient 3D model search and retrieval method is then proposed. This is an extension of the 2D image search technique where the "trace transform" is computed by tracing an image (2D function) with straight lines along which certain functionals of the image are calculated [19].

The "spherical trace transform," proposed in this paper, consists of tracing the volume of a 3D model with

- (i) radius segments,
- (ii) 2D planes, tangential to concentric spheres.

Then using three sets of functionals with specific properties, completely rotation invariant descriptor vectors are produced.

The paper is organized as follows. In Section 2 the proposed framework with the mathematical background is given. Section 3 presents in detail the proposed descriptor extraction method. In Section 4 the matching algorithms used are described. Experimental results evaluating the proposed method and comparing it with other methods are presented in Section 5. Finally, conclusions are drawn in Section 6.

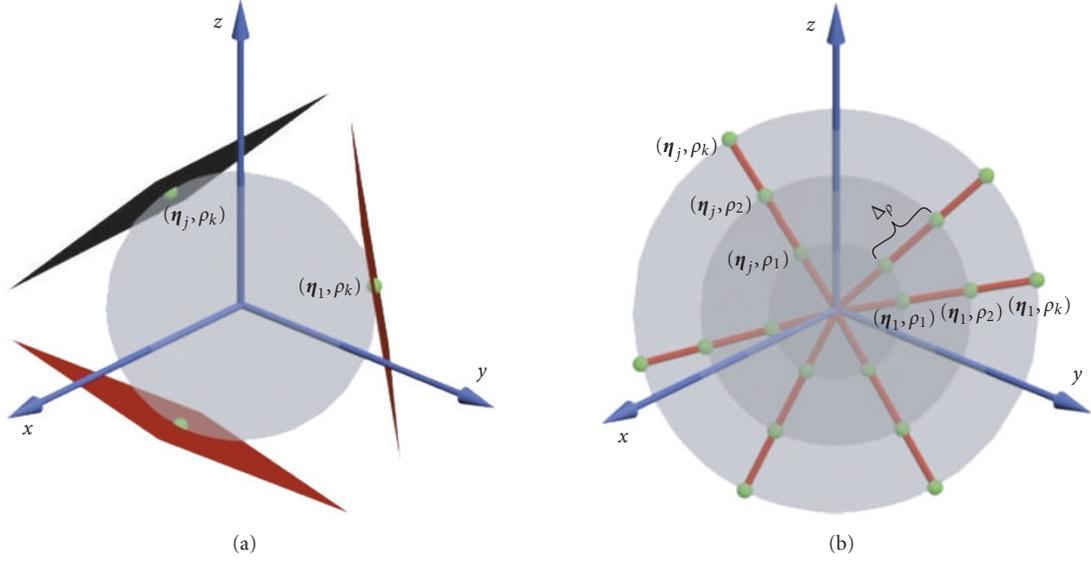


FIGURE 1: The spherical trace transform.

## 2. THE SPHERICAL TRACE TRANSFORM

Let  $M$  be a 3D model and  $f(\mathbf{x})$  the binary volumetric function of  $M$ , where  $\mathbf{x} = [x, y, z]^T$ , and

$$f(\mathbf{x}) = \begin{cases} 1 & \text{when } \mathbf{x} \text{ lies within the 3D model's volume,} \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

Let us define plane  $\Pi(\boldsymbol{\eta}, \rho) = \{\mathbf{x} \mid \mathbf{x}^T \cdot \boldsymbol{\eta} = \rho\}$  to be tangential to the sphere  $S_\rho$  with radius  $\rho$  and center at the origin, at the point  $(\boldsymbol{\eta}, \rho)$ , where  $\boldsymbol{\eta} = [\cos \phi \sin \theta, \sin \phi \sin \theta, \cos \theta]$  is the unit vector in  $\mathcal{R}^3$ , and  $\rho$  a real positive number (Figure 1(a)). Additionally, let us define radius segment  $\Lambda(\boldsymbol{\eta}, \rho) = \{\mathbf{x} \mid \mathbf{x}/|\mathbf{x}| = \boldsymbol{\eta}, \rho \leq |\mathbf{x}| < \rho + \Delta_\rho\}$ , where  $\Delta_\rho$  is the length of the radius segment (Figure 1(b)).

The intersection of  $\Pi(\boldsymbol{\eta}, \rho)$  with  $f(\mathbf{x})$  produces a 2D function  $\hat{f}(a, b)$ , ( $a, b \in \Pi(\boldsymbol{\eta}, \rho) \cap f(\mathbf{x})$ ), which is then sampled and its discrete form  $\hat{f}(i, j)$ , ( $i, j \in \mathcal{N}$ ) is produced. Similarly, the intersection of  $\Lambda(\boldsymbol{\eta}, \rho)$  with  $f(\mathbf{x})$  produces a 1D function  $\check{f}(c)$  ( $c \in \Lambda(\boldsymbol{\eta}, \rho) \cap f(\mathbf{x})$ ) which is also sampled and its discrete form  $\check{f}(i)$ , ( $i \in \mathcal{N}$ ) is produced. These two forms of data,  $\hat{f}(i, j)$  and  $\check{f}(i)$ , will serve as input in the sequel.

The ‘‘spherical trace transform,’’ proposed in this paper can be expressed using the general formulas

$$\begin{aligned} g_s(T; F; h) &= T(F(h(\cdot))), \\ g_a(T; A; F; h) &= T(A(F(h(\cdot)))) \end{aligned} \quad (2)$$

where

$$h(\cdot) = \begin{cases} \hat{f}(i, j), & \text{assuming representation using 2D planes} \\ \check{f}(i), & \text{assuming representation using} \\ & \text{radius segments} \end{cases} \quad (3)$$

and  $F(\boldsymbol{\eta}, \rho)$  denotes an ‘‘initial functional,’’ which can be applied to each  $\hat{f}(i, j)$  or  $\check{f}(i)$ , that is,  $F(\boldsymbol{\eta}, \rho) = F(\hat{f}(i, j))$  or  $F(\boldsymbol{\eta}, \rho) = F(\check{f}(i))$ . The set of  $F(\boldsymbol{\eta}, \rho)$  is treated either as a collection of spherical functions  $\{F^\rho(\boldsymbol{\eta})\}_\rho$  parameterized by  $\rho$ , or as a collection of radial functions  $\{F^\eta(\rho)\}_\eta$  parameterized by  $\boldsymbol{\eta}$ .

In the first case, a set of ‘‘spherical functionals’’  $T(\rho)$  is applied to each  $F^\rho(\boldsymbol{\eta})$ , producing a descriptor vector  $g_s(T) = T(F^\rho(\boldsymbol{\eta}))$ .

In the second case, a set of ‘‘actinic functionals’’  $A(\boldsymbol{\eta})$  is applied to each  $F^\eta(\rho)$ , producing the  $A(\boldsymbol{\eta}) = A(F^\eta(\rho))$ . Then, the  $T$  functionals are applied to  $A(\boldsymbol{\eta})$ , generating another descriptor vector  $g_a(T) = T(A(\boldsymbol{\eta}))$ .

Let us now examine the conditions that must be satisfied by the functionals in order to produce rotation invariant descriptor vectors. Under a 3D object rotation governed by a 3D rotation matrix  $\mathbf{R}$ , the points  $\boldsymbol{\eta}$  will be rotated:

$$\boldsymbol{\eta}' = \mathbf{R} \cdot \boldsymbol{\eta}, \quad (4)$$

therefore

$$F(\boldsymbol{\eta}', \rho) = F(\mathbf{R} \cdot \boldsymbol{\eta}, \rho) \quad (5)$$

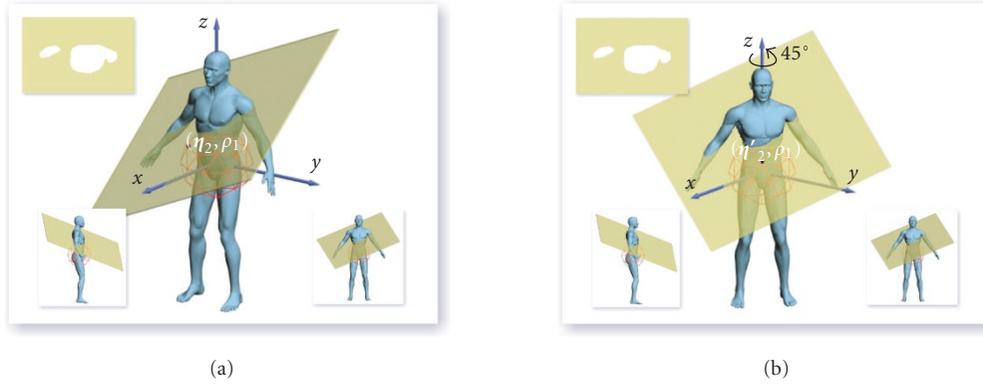


FIGURE 2: Rotation of  $f(\mathbf{x})$  rotates  $F(\boldsymbol{\eta}, \rho)$ , without rotating the corresponding  $f(i, j)$  (upper left image). Thus,  $F(\boldsymbol{\eta}_2, \rho_1) = F(\boldsymbol{\eta}'_2, \rho_1)$ .

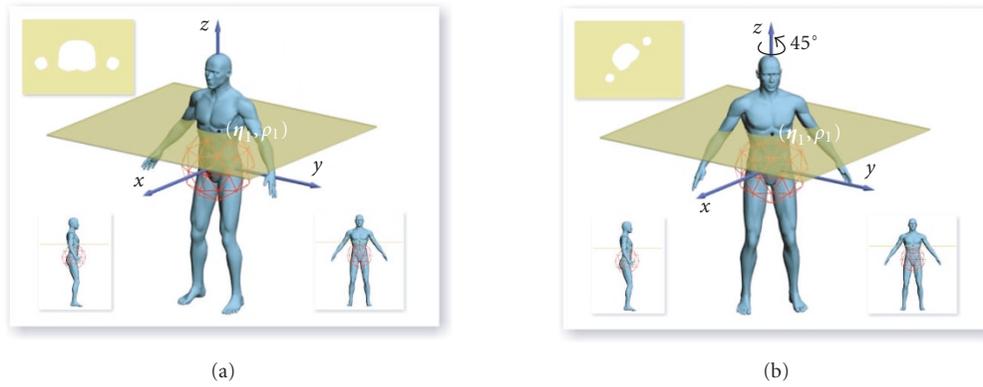


FIGURE 3: Rotation of  $f(\mathbf{x})$  rotates  $\hat{f}(i, j)$  (upper left image) without causing a rotation of the point  $(\boldsymbol{\eta}_1, \rho_1)$ .

and thus, rotation invariant  $T$  functionals must be applied, so that  $T(F(\boldsymbol{\eta}', \rho)) = T(F(\boldsymbol{\eta}, \rho))$  (Figure 2).

In the specific case where the points  $\boldsymbol{\eta}$  lie on the axis of rotation the corresponding  $\hat{f}(i, j)$  will be rotated (Figure 3), that is,

$$\hat{f}'(i, j) = \hat{f}(i', j') \quad (6)$$

and thus, 2D rotation invariant functionals must be applied, so that  $F(\hat{f}'(i, j)) = F(\hat{f}(i', j'))$ . Therefore, a general solution is given using 2D rotation invariant functionals  $F$  and rotation invariant spherical functionals  $T$ , producing completely rotation invariant descriptor vectors.

The functionals which satisfy the above-stated conditions, as initial, actinic, and spherical, will be briefly discussed in the following section.

The advantage of this approach is threefold: firstly, the rotation normalization which hampers the performance of

the descriptors in most 3D search approaches, is avoided. Secondly, the possibility of constructing a large number of descriptor vectors is presented. Indeed, the recognition of 3D objects is facilitated when a large number of features are present and in fact, the more classes must be distinguished, the more features may be necessary. The proposed method permits the construction of a large number of invariant features by defining a sufficient number of  $F$ ,  $A$ , and  $T$  functionals. Thirdly, the use of the  $T$  functionals leads to the definition of descriptor vectors with low dimensionality since each  $T$  functional produces a single number per concentric sphere. Thus, a compact representation of the descriptor vectors is achieved, which in turn simplifies the comparison between two models.

Another advantage of the proposed method is that it overcomes the problem analyzed in [12, Section 5.2] that face all the existing algorithms that use a rotation invariant transformation applied on concentric spheres. When independent

rotations are applied on an object at specific radius, an object of totally different shape will be produced. Because of the integration over all shells of the same radius, all these methods will produce identical descriptors for these totally different objects. The proposed method will not be affected of such a transformation, since in the case of decomposing the object's volume in 2D planes, the planes will contain information of the object in different radius. Moreover, the actinic functionals will be applied on the results from the previous step, that all share the same angular position, thus information on the different spheres will be combined. These two facts will assure that objects, of totally different shape, produced from transformations of independent rotations on an object, will not produce identical descriptors.

In the following a brief description of the functionals that were selected will be given.

## 2.1. Initial functionals $F$

### 2.1.1. The "mutated" radial integration transform (RIT)

Let  $\Lambda(\boldsymbol{\eta}, \rho) = \{\mathbf{x} \mid |\mathbf{x}| = \rho, \rho \leq |\mathbf{x}| < \rho + \Delta\rho\}$  be a radius segment (Figure 1(b)). Let also  $\check{f}_t(i)$  be the discrete function, which is derived from  $\check{f}_t(c)$ .  $\check{f}_t(c)$  is produced from the intersection of  $f(\mathbf{x})$  with the  $\Lambda(\boldsymbol{\eta}_t, \rho_t)$  which begins from the point  $(\boldsymbol{\eta}_t, \rho_t)$  and ends at the point  $(\boldsymbol{\eta}_t, \rho_t + \Delta\rho)$ . Then, the "mutated" radial integration transform RIT( $\boldsymbol{\eta}, \rho$ ) [10] is defined as:

$$\text{RIT}(\boldsymbol{\eta}_t, \rho_t) = \sum_{i=0}^{N-1} \check{f}_t(i), \quad (7)$$

where  $t = 1, \dots, N_R$ ,  $N_R$  is the total number of radius segments, and  $N$  is the total number of sampled points on each line segment.

### 2.1.2. 1D Fourier transform

The 1D discrete Fourier transform of  $\check{f}_t(i)$  is calculated, producing the vectors  $\text{DF}_t(k)$ , where  $t = 1, \dots, N_R$ ,  $N_R$  is the total number of radius segments, and  $k = 0, \dots, N-1$ ,  $N$  is the total number of sampled points on each radius segment. The vectors contain only the first  $K$  harmonic amplitudes. As a result, the 1D DFT generates  $K$  different initial functionals.

### 2.1.3. The 3D Radon transform

Let  $\Pi(\boldsymbol{\eta}, \rho) = \{\mathbf{x} \mid \mathbf{x}^T \cdot \boldsymbol{\eta} = \rho\}$  be a plane (Figure 1(a)). Let also  $\hat{f}_t(i, j)$  be the discrete function, which is derived from  $\hat{f}_t(a, b)$ . The function  $\hat{f}_t(a, b)$  is produced from the intersection of  $f(\mathbf{x})$  with  $\Pi(\boldsymbol{\eta}_t, \rho_t)$ , which is tangential to the sphere with radius  $\rho_t$  at the point  $(\boldsymbol{\eta}_t, \rho_t)$ . Then, the 3D radon transform  $R(\boldsymbol{\eta}, \rho)$  is defined as

$$R(\boldsymbol{\eta}_t, \rho_t) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \hat{f}_t(i, j), \quad (8)$$

where  $t = 1, \dots, N_R$ ,  $N_R$  is the total number of planes ( $\equiv$  total number of radius segments), and  $N \times N$  are the sampled points on each plane.

### 2.1.4. The Polar-Fourier transform

The discrete Fourier transform (DFT) is computed for each  $\hat{f}_t(i, j)$ , producing the vectors  $\text{FT}_t(k, m)$ , where  $k, m = 0, \dots, N-1$  and  $t = 1, \dots, N_R$ . Considering the first  $\mathcal{K} \times \mathcal{M}$  harmonic amplitudes for each  $\hat{f}_t(i, j)$ , the polar-DFT generates  $\mathcal{K} \times \mathcal{M}$  different initial functionals.

### 2.1.5. Hu moments

Moment invariants have become a classical tool for 2D object recognition. They were firstly introduced by Hu [20], who employed the results of the theory of algebraic invariants [21] and derived the seven well-known Hu moments,  $\phi_i$ ,  $i = 1, \dots, 7$ , which are invariant to the rotation of 2D objects. They are calculated for each  $\hat{f}_t(i, j)$  with spatial dimension  $N \times N$ , producing the vectors  $\text{HU}_t^i$ , where  $i = 1, \dots, 7$  and  $t = 1, \dots, N_R$ .

### 2.1.6. Zernike moments

Zernike moments are defined over a set of complex polynomials which forms a complete orthogonal set over the unit disk and are rotation invariant. The Zernike moments  $Z_{km}$  [22], where  $k \in \mathbb{N}^+$ ,  $m \leq k$ , are calculated for each  $\hat{f}_t(i, j)$  with spatial dimension  $N \times N$ , producing the vectors  $Z_t^{km}$ .

### 2.1.7. Krawtchouk moments

Krawtchouk moments are a set of moments formed by using Krawtchouk polynomials as the basis function set. Following the analysis in [23] and some specifications mentioned in [24], they were computed for each  $\hat{f}_t(i, j)$  producing the vectors  $K_t^{km}$ .

### 2.1.8. The 2D Polar wavelet transform

The 2D wavelet transform includes the convolution of the two-dimensional function  $\hat{f}_t(i, j)$  with a pair of QMF filters, followed by downsampling by a factor of two. In order to produce rotation invariant features,  $\hat{f}_t(i, j)$  should be transformed to the polar coordinate system, resulting in the Polar wavelet transform [25]. In the first level of decomposition, four different subbands are produced. The rotation invariant functionals  $\text{WT}_t^{km}$  are derived by computing an energy signature for each subband ( $k, m = 0, 1$ ). In this paper, the Daubechies  $D_6$  wavelet [26] was chosen as an appropriate pair of filters.

Each of the aforementioned  $F$  functionals produces a value (in case of RIT and Radon), or more values (in all other cases), per plane or per radius segment. The entire set

of values for each initial functional  $F$  generates a function  $F(\boldsymbol{\eta}, \rho)$  whose domain consists of concentric spheres.

## 2.2. Actinic functionals $A$

The  $F(\boldsymbol{\eta}, \rho)$  produced as above is now treated as a collection of radial functions  $F^\eta(\rho)$  by restricting at different  $\boldsymbol{\eta}$ . Then, the following set of ‘‘actinic functionals’’  $A_i(\boldsymbol{\eta})$ ,  $i = 1, \dots, 4$ , is applied to each  $F^\eta(\rho_t)$ :

- (1)  $A_1(\boldsymbol{\eta}) = DF(F^\eta(\rho_t)) = DF_k^\eta(\rho_t)$ ,
- (2)  $A_2(\boldsymbol{\eta}) = \max\{F^\eta(\rho_t)\}$ ,
- (3)  $A_3(\boldsymbol{\eta}) = \max\{F^\eta(\rho_t)\} - \min\{F^\eta(\rho_t)\}$ ,
- (4)  $A_4(\boldsymbol{\eta}) = \sum_{t=1}^{N_r} |F'^\eta(\rho_t)|$ ,

where  $F'$  is the derivative of  $F$ ,  $t = 1, \dots, N_r$  are sample points on each  $\boldsymbol{\eta}$ , and  $N_r$  is their total number.

## 2.3. Spherical functionals $T$

The set of functionals  $T$ , which is applied to each  $F^\rho(\boldsymbol{\eta})$  and  $A_i(\boldsymbol{\eta})$ , in order to produce the descriptor vector, includes

- (1)  $T_1(\omega) = \max\{\omega(\boldsymbol{\eta}_j)\}$ ,  $j = 1, \dots, N_s$ ,
- (2)  $T_2(\omega) = \sum_{j=1}^{N_s} |\omega'(\boldsymbol{\eta}_j)|$ ,
- (3)  $T_3(\omega) = \sum_{j=1}^{N_s} \omega(\boldsymbol{\eta}_j)$ ,
- (4)  $T_4(\omega) = \max\{\omega(\boldsymbol{\eta}_j)\} - \min\{\omega(\boldsymbol{\eta}_j)\}$ ,  $j = 1, \dots, N_s$ ,
- (5) the amplitudes of the first  $L$  harmonics of the spherical Fourier transform (SFT), applied on  $\omega(\boldsymbol{\eta}_j)$ , which are also called as the ‘‘rotationally invariant shape descriptors’’  $A_l$  [27]. In the proposed method, for each  $l$ ,  $l = 1, \dots, L$ , the corresponding  $A_l$  is a spherical functional  $T$ ,

where  $\omega(\boldsymbol{\eta}_j) = F^\rho(\boldsymbol{\eta}_j)$  or  $\omega(\boldsymbol{\eta}_j) = A_i(\boldsymbol{\eta}_j)$ ,  $\omega'$  its derivative, and  $N_s = N_R/N_c$ , where  $N_c$  is the total number of concentric spheres.

In our case,

$$\omega(\boldsymbol{\eta}) = \begin{cases} \text{RIT}^\rho(\boldsymbol{\eta}), \\ \text{DF}_k^\rho(\boldsymbol{\eta}), \\ R^\rho(\boldsymbol{\eta}), \\ \text{FT}_{km}^\rho(\boldsymbol{\eta}), \\ \text{HU}_k^\rho(\boldsymbol{\eta}), \\ Z_{km}^\rho(\boldsymbol{\eta}), \\ K_{km}^\rho(\boldsymbol{\eta}), \\ \text{WT}_{km}^\rho(\boldsymbol{\eta}), \\ A(\boldsymbol{\eta}). \end{cases} \quad (9)$$

Concluding this section, it should be noted that the total number of spherical functionals  $T$  used is  $L + 4$  for each concentric sphere.

## 3. DESCRIPTOR EXTRACTION PROCEDURE

### 3.1. Preprocessing

A 3D model  $M$  is generally described by a 3D mesh. Let  $R \times R \times R$  be the size of the smallest cube bounding the mesh. The bounding cube is partitioned in  $(2 \cdot N)^3$  equal cube shaped voxels  $u_i$  with centers  $\mathbf{v}_i = [x_i, y_i, z_i]$ , where  $i = 1, \dots, (2 \cdot N)^3$ . The size of each voxel is  $(R/(2 \cdot N))^3$ . Let  $U$  be the set of all voxels inside the bounding cube and  $U_1 \subseteq U$ , be the set of all voxels belonging to the bounding cube and lying inside  $M$ . Then, the discrete binary volume function  $\tilde{f}(\mathbf{v}_i)$  of  $M$ , is defined as

$$\tilde{f}(\mathbf{v}_i) = \begin{cases} 1 & \text{when } u_i \in U_1, \\ 0 & \text{otherwise.} \end{cases} \quad (10)$$

In order to achieve translation invariance, the center of mass of the model is first calculated. Then, the model is translated so that the center of mass coincides with the center of the bounding cube. Translation invariance follows.

To achieve scaling invariance, the maximum distance  $d_{\max}$  between the center of mass and the most distant voxel, where  $\tilde{f}(\mathbf{v}_i) = 1$ , is calculated. Then, the translated  $\tilde{f}(\mathbf{v}_i)$  is scaled so that  $d_{\max} = 1$ . At this point, scaling invariance is also accomplished.

A coarser mesh is then constructed by combining every eight neighboring voxels  $u_i$ , to form a bigger voxel  $v_k$  with centers  $v_k$ ,  $k = 1 \dots, N^3$ . The discrete integer volume function  $\tilde{f}(v_k)$  of  $M$  is defined as

$$\tilde{f}(v_k) = \sum_{n=1}^8 \tilde{f}(\mathbf{v}_n) : u_n \in v_k. \quad (11)$$

Thus, the domain of  $\tilde{f}(v_k)$  is  $[0, \dots, 8]$ . The procedure described in Section 2 is then applied to the function  $\tilde{f}(v_k)$  instead of the function  $f(\mathbf{x})$ . Specifically,  $\tilde{f}(v_k)$  is assumed to intersect with planes. Each plane is tangential to the sphere with radius  $\rho$  at the point  $B$ . Further,  $\tilde{f}(v_k)$  is assumed to intersect with radius segments.

In order to avoid possible sampling errors caused using the lines of latitude and longitude (since they are too much concentrated towards the poles), each concentric sphere is simulated by an icosahedron where each of the 20 main triangles is iteratively subdivided into  $q$  equal parts to form sub-triangles. The vertices of the subtriangles are the sampled points  $B_t$ . Their total number  $N_s$ , for each concentric sphere (icosahedron)  $C_s$ , with radius  $\rho_s$ ,  $s = 1, \dots, N_c$ , where  $N_c$  is the total number of concentric spheres, is easily seen to be

$$N_s = 10 \cdot q^2 + 2. \quad (12)$$

### 3.2. Descriptor extraction

Each function  $\hat{f}_t(a, b)$ ,  $t = 1, \dots, N_s$ , is quantized into  $N \times N$  samples and its discrete form  $\hat{f}_t(i, j)$  is produced. The

domain of  $\hat{f}_i(i, j)$  is  $[0, \dots, 8]$ . Similarly, each function  $\hat{f}_i(c)$  is quantized into  $N$  samples and its discrete form  $\check{f}_i(i)$  is produced. The domain of  $\check{f}_i(i)$  is  $[0, \dots, 8]$ .

Then, the procedure described in Section 2 is followed for each functional  $F$ , producing the descriptor vectors  $g_s(T) = T(F^{\rho_i}(\boldsymbol{\eta}_t)) = \mathbf{D1}_F(l_1)$ , and  $g_a(T) = T(A(\boldsymbol{\eta}_t)) = \mathbf{D2}_F(l_2)$ , where  $l_1 = 1, \dots, (L+4) \cdot N_c$ ,  $l_2 = 1, \dots, (L+4) \cdot 4$  and  $L$  is the total number of spherical harmonics. The integrated descriptor vector is  $\mathbf{D}_F(l) = [\mathbf{D1}_F(l_1), \mathbf{D2}_F(l_2)]^T$ , where  $l = 1, \dots, \{(L+4) \cdot N_c + (L+4) \cdot 4\}$ .

The same procedure is followed for all  $F$  functionals, producing the descriptor vectors  $\mathbf{D}_{\text{RIT}}(l)$ ,  $\mathbf{D}_{\text{DF}_k}(l)$ ,  $\mathbf{D}_R(l)$ ,  $\mathbf{D}_{\text{HU}_k}(l)$ ,  $\mathbf{D}_{\text{FT}_{km}}(l)$ ,  $\mathbf{D}_{\text{Z}_{km}}(l)$ ,  $\mathbf{D}_{\text{K}_{km}}(l)$ , and  $\mathbf{D}_{\text{WT}_{km}}(l)$ .

Our experiments presented in the sequel were performed using the values  $N_R = 2562$ ,  $N_c = 20$ ,  $L = 26$ ,  $K = 8$ , and  $N = 64$ .

#### 4. MATCHING ALGORITHM

Let  $A, B$  be two 3D models. Let also  $\mathbf{D}^A(k) = [\mathbf{D}^{A1}(k_1), \mathbf{D}^{A2}(k_2)]^T$ ,  $\mathbf{D}^B(k) = [\mathbf{D}^{B1}(k_1), \mathbf{D}^{B2}(k_2)]^T$  be two descriptor vectors of the same kind  $\mathbf{D}(k)$ . The model descriptors are compared in pairs using their  $L1$ -distance:

$$D1_{\text{similarity}} = \sqrt{\sum_{k1=1}^{(L+4) \cdot N_c} |\mathbf{D}^{A1}(k1) - \mathbf{D}^{B1}(k1)|}, \quad (13)$$

$$D2_{\text{similarity}} = \sqrt{\sum_{k2=1}^{(L+4) \cdot 4} |\mathbf{D}^{A2}(k2) - \mathbf{D}^{B2}(k2)|}.$$

The overall similarity measure is determined by

$$D_{\text{similarity}} = a_1 \cdot D1_{\text{similarity}} + a_2 \cdot D2_{\text{similarity}}, \quad (14)$$

where  $a_1, a_2$  are descriptor vector percentage factors, which are calculated as follows. Let us assume that  $A$  belongs to a class  $C$ , which contains  $N_C$  models. Let also  $N_{\text{total}}$  be the total number of models contained in the database. Then the factor  $a_1$  is calculated as

$$a_1 = \frac{\sum_{i=1}^{N_C} d_i}{\sum_{j=1}^{N_{\text{total}} - N_C} d_j}, \quad (15)$$

where  $d_i$  is the  $L1$ -distance of the descriptor vector  $\mathbf{D}^{A1}$  of the model  $A$  from the descriptor vector  $\mathbf{D}^{A1'}$  of the model  $A'$  which also belongs to  $C$ , and  $d_j$  is the  $L1$ -distance of the descriptor vector  $\mathbf{D}^{A1}$  of the model  $A$  from the descriptor vector  $\mathbf{D}^{A1''}$  of the model  $A''$  which does not belong to  $C$ . The combination, small  $d_i$  and big  $d_j$ , implies that the descriptor vector  $\mathbf{D}^{A1}$  is good for the class  $C$ , in terms of successful retrieved results. The percentage factor  $a_2$  is calculated similarly taking into account the descriptor vector  $\mathbf{D}^{A2}$ . Then  $a_1$  and  $a_2$  are normalized so that  $1/a_1 + 1/a_2 = 100$ .

Following the above approach, a large number of descriptor vectors can be efficiently used, taking advantage of the discriminative power of each descriptor vector per different class.

Experiments have shown that a single descriptor vector does not outperform all the others, in terms of precision recall, in all different classes, thus using the percentage factors we take advantage of the real discriminative power of each descriptor vector per each different class. Such an approach has not been reported so far in this research field.

#### 4.1. Assigning weights to each class

In this section, a procedure for the calculation of weights characterizing the discriminative power of each descriptor vector per different class is described.

Let  $\mathbf{D}^i(j) = [D^i(1), \dots, D^i(S)]$  be a descriptor vector, where  $i = 1, \dots, N_{\text{total}}$ .  $N_{\text{total}}$  is the total number of 3D models and  $S$  is the total number of descriptors per descriptor vector. Let also  $C$  be a class with descriptor vectors:

$$M_C = \begin{bmatrix} D^1(1) & \dots & D^1(k) & \dots & D^1(S) \\ \dots & & & & \\ D^i(1) & \dots & D^i(k) & \dots & D^i(S) \\ \dots & & & & \\ D^{N_C}(1) & \dots & D^{N_C}(k) & \dots & D^{N_C}(S) \end{bmatrix}, \quad (16)$$

where  $N_C$  is the number of 3D models which belongs to class  $C$ .

Then, the *feature vectors*  $\mathbf{f}_{C1}, \dots, \mathbf{f}_{Ck}, \dots, \mathbf{f}_{CS}$  are formed, where  $C = 1, \dots, N_{\text{class}}$ ,  $\mathbf{f}_{Ck} = [D^1(k) \cdot \dots \cdot D^i(k) \cdot \dots \cdot D^{N_C}(k)]^T$ , and  $N_{\text{class}}$  is the total number of classes.

For each  $\mathbf{f}_{Ck}$ , the mean

$$\mu_{\mathbf{f}_{Ck}} = \frac{1}{N_C} \sum_{i=1}^{N_C} D^i(k) \quad (17)$$

and the variance

$$\sigma_{\mathbf{f}_{Ck}}^2 = \frac{1}{N_C} \sum_{i=1}^{N_C} (D^i(k))^2 - (\mu_{\mathbf{f}_{Ck}})^2 \quad (18)$$

are calculated. The magnitude of each weight  $W_{Ck}$  depends on two factors.

- (i) *The compactness factor*  $W^{(1)}$ : the  $W^{(1)}$  factor provides a measure of the compactness of the  $\mathbf{f}_{Ck}$  feature vector for the class  $C$ . It is calculated by

$$W_{Ck}^{(1)} = \frac{\sigma_{\mathbf{f}_{Ck}}}{\mu_{\mathbf{f}_{Ck}}}. \quad (19)$$

The lower the value of  $W_{Ck}^{(1)}$  the higher the weight of the  $k$ th feature vector of  $C$ th class.

(ii) *The dissimilarity factor  $W^{(2)}$* : the  $W^{(2)}$  factor provides a measure of dissimilarity between the feature vector  $\mathbf{f}_{Ck}$  of the class  $C$  and the corresponding feature vector  $\mathbf{f}_{C1k}$  of the class  $C1$ . The higher the  $W_{Ck}^{(2)}$  factor the more dissimilar is the  $k$ th feature vector of  $C$  class ( $\mathbf{f}_{Ck}$ ) when compared to the  $k$ th feature vectors of the other classes. Specifically, for the  $k$ th feature vector of  $C$ th class, the number  $M_{Ck}$  of the descriptors  $D^n(k)$ , where  $n \in ([1, \dots, N_{\text{class}}] - C)$ , which do not belong to  $[\mu_{\mathbf{f}_{Ck}} - \sigma_{Ck}, \mu_{\mathbf{f}_{Ck}} + \sigma_{Ck}]$  is calculated, and the  $W^{(2)}$  factor is evaluated using

$$W_{Ck}^{(2)} = \frac{M_{Ck}}{N_{\text{total}} - N_C}, \quad (20)$$

where  $N_{\text{total}}$  is the total number of 3D models and  $N_C$  is the number of models of the  $C$ th class. The final weights are calculated by

$$W_{Ck} = C_1(1 - W_{Ck}^{(1)}) + C_2 W_{Ck}^{(2)}, \quad (21)$$

where  $C_1, C_2 \in [0, 1]$  are coefficients and

$$C_1 + C_2 = 1. \quad (22)$$

It is obvious that

$$0 \leq W_{Ck} \leq 1. \quad (23)$$

It was experimentally found that best results were obtained for  $C_1 \in [0.2, 0.4]$  and  $C_2 \in [0.6, 0.8]$ .

A 2D array of weights is then created, for all models in database,

$$\mathbf{W} = \begin{bmatrix} W_{11} & \dots & W_{1k} & \dots & W_{1S} \\ & \dots & & & \\ W_{C1} & \dots & W_{Ck} & \dots & W_{CS} \\ & \dots & & & \\ W_{N_{\text{class}}1} & \dots & W_{N_{\text{class}}k} & \dots & W_{N_{\text{class}}S} \end{bmatrix}, \quad (24)$$

where  $W_{Ck}$  is the weight of the  $k$ th descriptor of the  $C$ th class. The weight matrix will be used to improve the performance of matching methods. In the following sections, two matching methods are described, where the contribution of weights to the final results is noticeable.

#### 4.2. First weight-based matching algorithm: "weight method 1" (WM1)

Let  $Q$  be a query model and  $A$  a model from the database to be compared with  $Q$ . The model descriptors are compared in

pairs using the following formula ( $L1$ -distance):

$$L1 = \sqrt{\sum_{k=1}^S W_{Ck} |D^Q(k) - D^A(k)|}, \quad (25)$$

where  $D^Q(k)$  is the  $k$ th descriptor of the query model  $Q$  and  $D^A(k)$  is the  $k$ th descriptor of the model  $A$  that belongs to class  $C$ . In this method, both  $D^Q(k)$  and  $D^A(k)$  descriptors are assigned the weight  $W_{Ck}$  of class  $C$ .

#### 4.3. Second weight-based matching algorithm: "weight method 2" (WM2)

Let now  $A^i$  ( $i = 1, \dots, N_{\text{total}}$ ) be a model of the database, where  $N_{\text{total}}$  is the total number of models in the database. In this method, the  $L1$ -distance between  $Q$  and  $A^i$  models is calculated. However, in this case,  $D^Q(k)$  and  $D^{A^i}(k)$  descriptors are not assigned the same weights.

Specifically, for a query  $Q$ ,  $N_{\text{class}}$  different cases are considered. For the  $n$ th case ( $n = 1, \dots, N_{\text{class}}$ ) it is assumed that the query  $Q$  belongs to class  $n$ , so that its  $D^Q(k)$  descriptor vector is assigned the corresponding  $\mathbf{W}_n(k)$  weight vector ( $n$ th row of the weight matrix). For each case  $n$ , for each pair of  $Q$  and  $A^i$  models, the  $L1$ -distance is calculated according to the following formula:

$$L1_n^i = \sqrt{\sum_{k=1}^S |W_{nk} D^Q(k) - W_{Ck} D^{A^i}(k)|}, \quad (26)$$

where  $n = 1, \dots, N_{\text{class}}$  and  $i = 1, \dots, N_{\text{total}}$ . In all  $N_{\text{class}}$  cases, the model  $A^i$  is assigned the same  $\mathbf{W}_C(k)$  weight vector ( $C$ th row of the weight matrix).

The final matching between  $Q$  and  $A^i$  is achieved by choosing only one case  $n$  (out of  $N_{\text{class}}$ ). The query  $Q$  is assigned the same weights  $\mathbf{W}_n(k)$  for all  $L1^i$  distances. The selection of the optimal case  $n$  is based on the following procedure.

For each case  $n$ , all  $L1_n^i$  distances between the query  $Q$  and the models  $A^i$  of the database ( $i = 1, \dots, N_{\text{total}}$ ) are sorted in ascending order. In order to evaluate the homogeneity of the retrieved results at the first positions of the ranking list, the popular "Gini" index  $I(n)$  [28] is used, as a measure of impurity. The smaller the Gini index, the lower the heterogeneity of the retrieved results:

$$I(n) = 1 - \sum_{C=1}^{N_{\text{class}}} p_C^2, \quad (27)$$

where  $p_C$  is the fraction of models retrieved at the first  $k$  positions of the ranking list that belong to class  $C$ , divided with  $k$ . Notice that  $I(n) = 0$  if all the retrieved models belong to the same class. The case  $n$  (out of  $N_{\text{class}}$ ) with the lowest Gini impurity index is used for the final matching between  $Q$  and  $A^i$  models (26).

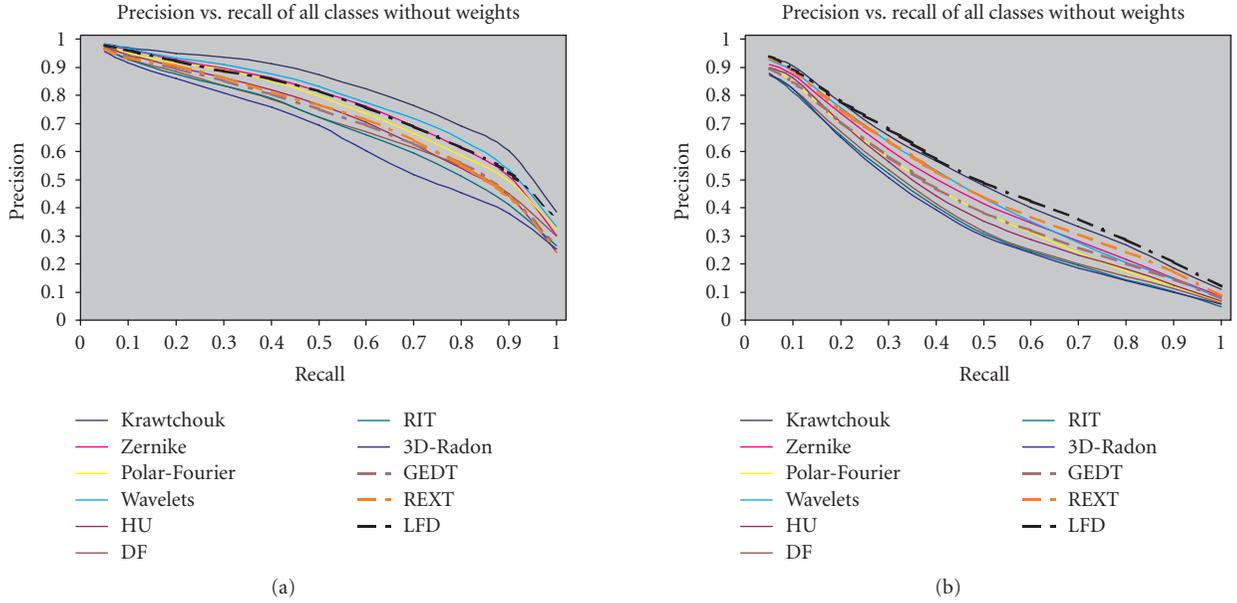


FIGURE 4: Precision-recall curves diagram using the new database (a) and the Princeton database (b).

If  $T > 1$  lowest impurity indices are encountered, a second measure is taken into account.

Let  $n_i = \arg \min I(n)$ ,  $i = 1, \dots, T$ . For each  $n_i$ , let the majority of the models retrieved at the first  $k$  positions of the ranking list belong to class  $C_i$ . The number  $M_{n_i}$  of the models of category  $C_i$ , from the first position to the position that a model of a category other than  $C_i$  occurs, is calculated for each  $n_i$ . The fraction  $M_{n_i}/N_{C_i}$ , where  $N_{C_i}$  is the total number of models in class  $C_i$ , is the second measure for the selection of the best value of  $n_i$ . The value leading to the largest value of the fraction above is the one selected for the final matching, that is,  $n_i = \arg \max \{M_{n_i}/N_{C_i}\}$ .

## 5. EXPERIMENTAL RESULTS

The proposed method was tested using two different databases. The first one, formed in Princeton University [29] consists of 907 3D models classified into 35 main categories. Most are further classified into subcategories, forming 92 categories in total. This classification reflects primarily the function of each object and secondarily its form [30]. The second one was compiled from the Internet by us, it consists of 544 3D models from different categories and was also used in [31]. The VRML models were collected from the World Wide Web so as to form 13 more balanced categories: 27 animals, 17 spheroid objects, 64 conventional airplanes, 55 delta airplanes, 54 helicopters, 48 cars, 12 motorcycles, 10 tubes, 14 couches, 42 chairs, 45 fish, 53 humans, and 103 other models. This choice reflects primarily the shape of each object and secondarily its function. The average numbers of vertices and triangles of the models in the new database are 5080 and 7061, respectively.

To evaluate the proposed method, each 3D model was used as a query object. Our results were compared with those of the following methods, which have been reported [29] as

the best-known shape matching methods that produce the best retrieval results.

- (i) *Gaussian Euclidean distance transform (GEDT)*: it is based on the comparison of a 3D function, whose value at each point is given by composition of a Gaussian with the Euclidean distance transform of the surface [12].
- (ii) *Light field descriptor (LFD)*: uses a representation of a model as a collection of images rendered from uniformly sampled positions on a view sphere. The distance between two descriptors is defined as the minimum  $L1$ -difference, taken over all rotations and all pairings of vertices on two dodecahedra [7].
- (iii) *Radialized spherical extent function (REXT)*: uses a collection of spherical functions giving the maximal distance from center of mass as a function of spherical angle and radius [32].

It is noted that we did not implement the above methods. All executables were taken from the home pages of the authors of [7, 12, 32].

The retrieval performance was evaluated in terms of “precision” and “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.

Experimental results have shown that the following descriptor vectors should be selected, for achieving best performance, in the case of multiple descriptor vector extraction:  $FT = \{FT_{00}, FT_{01}, FT_{10}\}$ ,  $HU = \{HU_0, HU_3\}$ ,  $Z = \{Z_{00}, Z_{11}, Z_{20}, Z_{31}\}$ ,  $K = \{K_{00}, K_{01}, K_{02}, K_{11}\}$ ,  $WT = \{WT_{00}, WT_{01}, WT_{10}, WT_{11}\}$ , and  $DF = \{DF_2, DF_4\}$ .

Figure 4(a) contains a numerical precision versus recall comparison with the aforementioned methods using the new

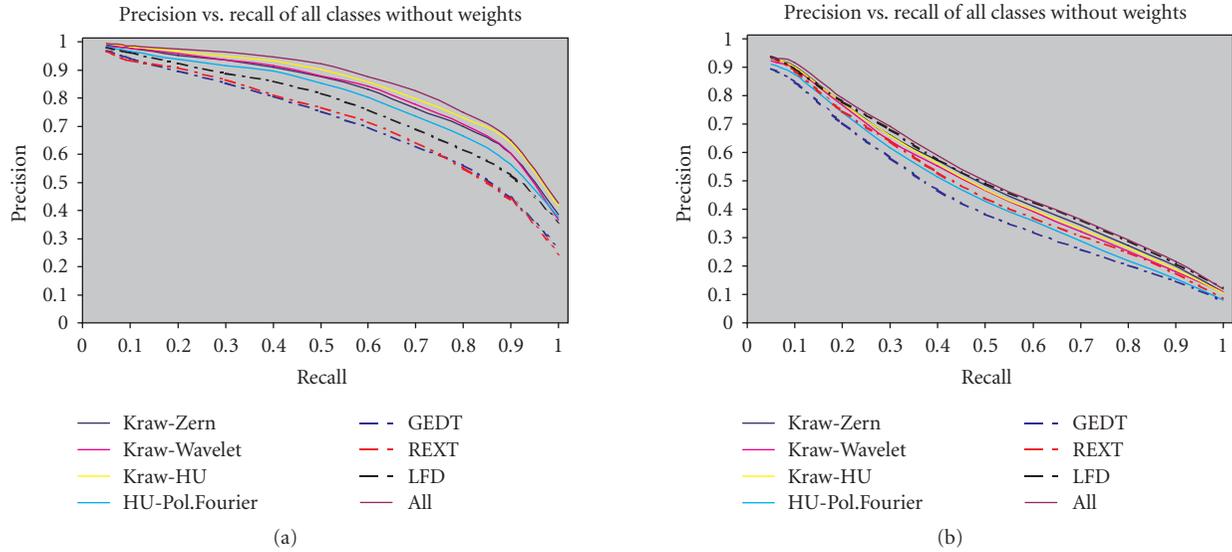


FIGURE 5: Precision-recall curves diagram: some of the best descriptor vector combinations, using the new database (a) and the Princeton database (b).

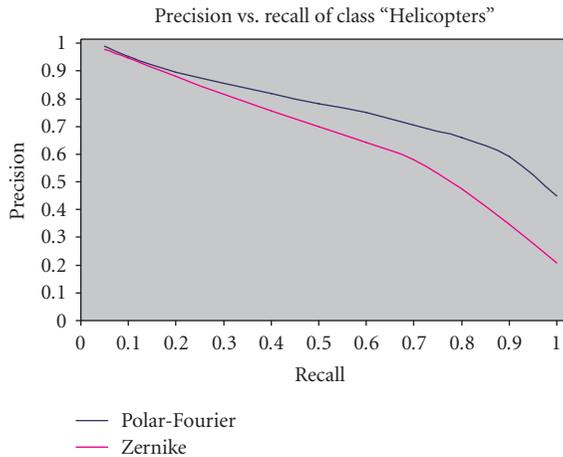


FIGURE 6: Comparison of the efficiency of the Polar-Fourier-based descriptor vector against the Zernike moments-based descriptor vector for a class of the new database.

database. It is clear that the proposed method outperforms all others using the integrated descriptor vector and calculating the percentage factors for each descriptor vector. Additionally, other descriptor vectors produced by Krawtchouk moments, Zernike moments, the Polar wavelet transform, the Polar-Fourier transform, and the HU moments outperform or are competitive with the other known state-of-the-art methods. Figure 4(b) illustrates the results using the Princeton database. In this database, the LFD method provides the best retrieval precision, and only the descriptor vec-

tors based on the Krawtchouk moments and on the Zernike moments are competitive.

In Figure 5, some of the best combinations which significantly improve the retrieval performance of the proposed method are shown. The retrieval performance is improved due to the fact that a single descriptor vector does not outperform all the others in all different classes, thus using the percentage factors (see Section 4) we can take advantage of the real discriminative power of each descriptor vector per each different class. An example is illustrated in Figure 6 where the descriptor vector based on Polar-Fourier transform is seen to outperform the descriptor vector based on Zernike moments in class "helicopters" of the new database. However, the overall retrieval performance of the descriptor vector based on Zernike moments is better (Figure 4(a)). Figure 5 illustrates the results obtained using all the descriptor vectors and their percentage factors. It is clear that the proposed method outperforms all known methods in both databases. However, this procedure is time consuming, thus, simpler alternatives such as the combination *Krawtchouk-Zernike*, or the combination *Krawtchouk-Hu*, can be used instead, with very good results.

Figure 7 depicts the precision-recall diagram using the "weight method 1" (WM1) using the new database and the Princeton database. It is obvious that the retrieval results were improved significantly. In Figure 8 some of the best combinations which significantly improve the retrieval performance of the proposed method are shown.

Figure 9 illustrates the precision-recall diagram using the "weight method 2" (WM2) using the new database and the Princeton database. The results are impressive, especially for

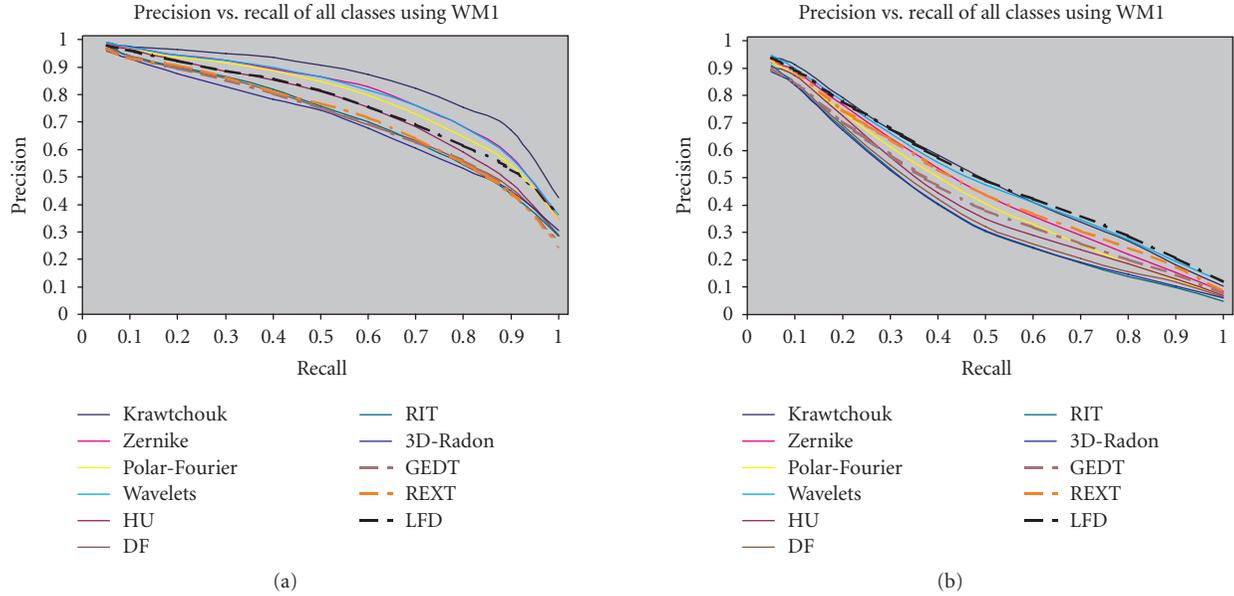


FIGURE 7: Precision-recall curves diagram using the weight method 1 for the new database (a) and for the Princeton database (b).

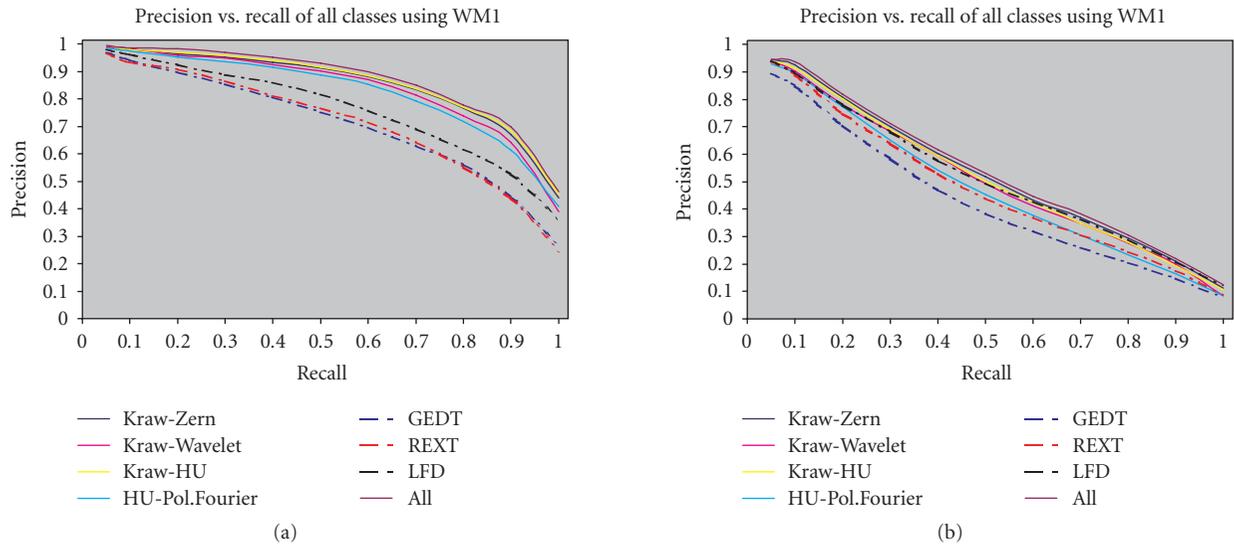


FIGURE 8: Precision-recall curves diagram some of the best descriptor vector combinations, using the weight method 1 for the new database (a) and for the Princeton database (b)

the new database where all of the proposed descriptor vectors outperform the others.

In Figure 10 some of the best combinations which significantly improve the retrieval performance of the proposed method are depicted.

Figure 11 illustrates the results of the experiments performed in the new database using different dimensionality for the RIT-based descriptor vector changing the number  $L$  of the harmonics of the spherical Fourier transform. It is obvious that an increase in precision is observed if the number

of spherical harmonics  $L$  increases from  $L = 21$  to  $L = 26$ . However, there was no commensurate modification in precision for values of  $L$  higher than 26, while the time needed for the extraction of the descriptor vectors as well as for carrying out the matching procedure increased sharply.

## 6. CONCLUSIONS

In this paper a novel framework for 3D model search and retrieval was proposed. A set of functionals is applied to

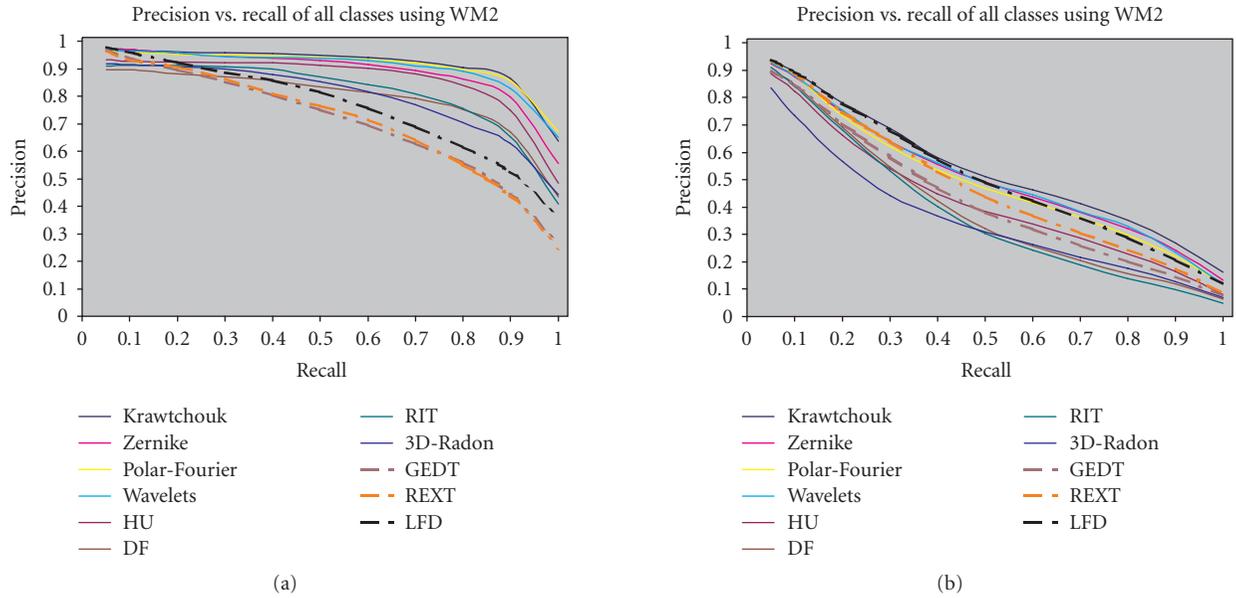


FIGURE 9: Precision-recall curves diagram using the weight method 2 for the new database (a) and for the Princeton database (b).

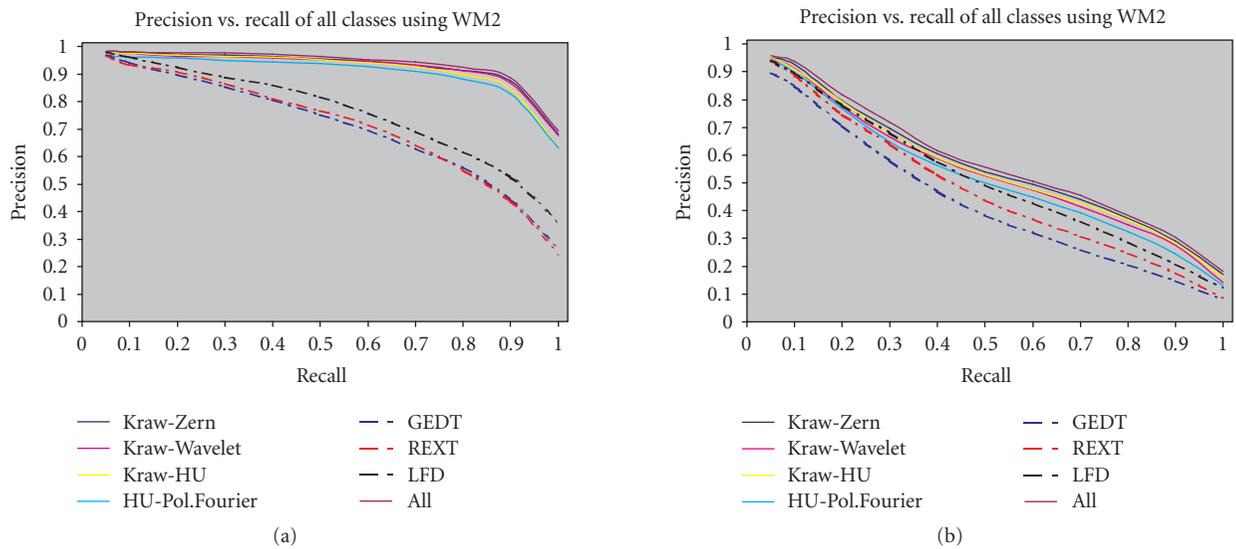


FIGURE 10: Precision-recall curves diagram some of the best descriptor vector combinations, using the weight method 2 for the new database (a) and for the Princeton database (b).

the volume of the 3D model producing a new domain of concentric spheres. In this new domain, a new set of functionals is applied, resulting in a completely rotation invariant descriptor vector, which is used for 3D model matching. Further, a novel technique, where weights are assigned to the descriptors, is introduced, which improves significantly

the retrieval results. Experiments were performed using two different databases and the results of the proposed method were compared with those of the best known retrieval methods in the literature. The results show clearly that the proposed method outperforms all others in terms of precision recall.

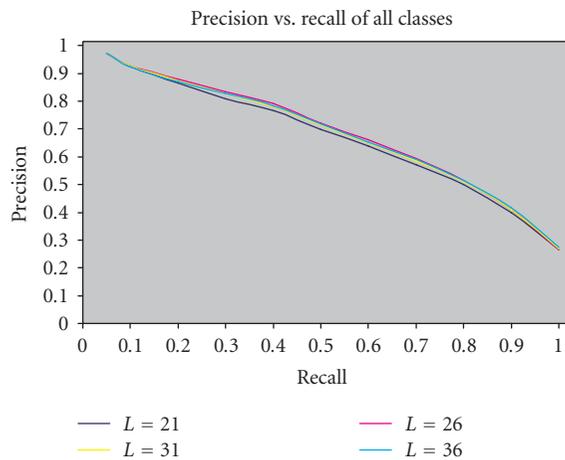


FIGURE 11: Comparison of the efficiency of RIT-based descriptor vectors using different dimensionality, in terms of precision-recall diagram using the new database.

## ACKNOWLEDGMENTS

This work was supported by the ALTAB23D project of the Greek Secretariat of Research and Technology and by the CATER EC IST project.

## REFERENCES

- [1] 3D Cafe, <http://www.3Dcafe.com>.
- [2] The Protein Data Bank, <http://www.rcsb.org>.
- [3] I. Kolonias, D. Tzovaras, S. Malassiotis, and M. G. Strintzis, "Fast content-based search of VRML models based on shape descriptors," *IEEE Transactions on Multimedia*, vol. 7, no. 1, pp. 114–126, 2005.
- [4] R. Ohbuchi, T. Otagiri, M. Ibato, and T. Takei, "Shape-similarity search of three-dimensional models using parameterized statistics," in *Proceedings of the 10th Pacific Conference on Computer Graphics and Applications*, pp. 265–274, Beijing, China, October 2002.
- [5] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin, "Matching 3D models with shape distributions," in *Proceedings of the International Conference on Shape Modelling and Applications (SMI '01)*, pp. 154–166, Genova, Italy, May 2001.
- [6] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin, "Shape distributions," *ACM Transactions on Graphics*, vol. 21, no. 4, pp. 807–832, 2002.
- [7] D.-Y. Chen, X.-P. Tian, Y.-T. Shen, and M. Ouhyoung, "On visual similarity based 3D model retrieval," *Computer Graphics Forum*, vol. 22, no. 3, pp. 223–232, 2003.
- [8] D. V. Vranić and D. Saupe, "Description of 3D-shape using a complex function on the sphere," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME '02)*, vol. 1, pp. 177–180, Lausanne, Switzerland, August 2002.
- [9] D. V. Vranić, D. Saupe, and J. Richter, "Tools for 3D-object retrieval: Karhunen-Loeve transform and spherical harmonics," in *Proceedings of the 4th IEEE Workshop on Multimedia Signal Processing (MMSP '01)*, pp. 293–298, Cannes, France, October 2001.
- [10] P. Daras, D. Zarpalas, D. Tzovaras, and M. G. Strintzis, "Generalized radon transform based 3D feature extraction for 3D object retrieval," in *Proceedings of the 5th International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS '04)*, Lisboa, Portugal, April 2004.
- [11] B. Horn, H. Hilden, and S. Negahdaripour, "Closed-form solution of absolute orientation using orthonormal matrices," *Journal of the Optical Society of America*, vol. 5, no. 7, pp. 1127–1135, 1988.
- [12] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz, "Rotation invariant spherical harmonic representation of 3D shape descriptors," in *Proceedings of the Eurographics Symposium on Geometry Processing*, pp. 156–164, Aachen, Germany, June 2003.
- [13] M. Hilaga, Y. Shinagawa, T. Kohmura, and T. L. Kunii, "Topology matching for fully automatic similarity estimation of 3D shapes," in *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '01)*, pp. 203–212, Los Angeles, Calif, USA, August 2001.
- [14] MPEG Video Group, "MPEG-7 visual part of eXperimentation model (version 9.0)," in *Proceedings of the 55th Mpeg Meeting*, Pisa, Italy, 2001, ISO/MPEG N3914.
- [15] T. Funkhouser, P. Min, M. Kazhdan, et al., "A search engine for 3D models," *ACM Transactions on Graphics*, vol. 22, no. 1, pp. 83–105, 2003.
- [16] M. Novotni and R. Klein, "3D Zernike descriptors for content based shape retrieval," in *Proceedings of the 8th ACM Symposium on Solid Modeling and Applications*, pp. 216–225, Seattle, Wash, USA, June 2003.
- [17] N. Canterakis, "3D Zernike moments and Zernike affine invariants for 3D image analysis and recognition," in *Proceedings of the 11th Scandinavian Conference on Image Analysis*, Kangerlussuaq, Greenland, June 1999.
- [18] M. T. Suzuki, "A web-based retrieval system for 3D polygonal models," in *Proceedings of the Joint 9th IFSA World Congress and 20th NAFIPS International Conference*, vol. 4, pp. 2271–2276, Vancouver, Canada, July 2001.
- [19] A. Kadyrov and M. Petrou, "The trace transform and its applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 8, pp. 811–828, 2001.
- [20] M.-K. Hu, "Visual pattern recognition by moment invariants," *IEEE Transactions on Information Theory*, vol. 8, no. 2, pp. 179–187, 1962.
- [21] G. B. Gurevich, *Foundations of the Theory of Algebraic Invariants*, Noordhoff, Groningen, The Netherlands, 1964.
- [22] A. Padilla-Vivanco, A. Martinez-Ramirez, and F. Granados-Agustin, "Digital image reconstruction by using Zernike moments," in *Optics in Atmospheric Propagation and Adaptive Systems VI*, vol. 5237 of *Proceedings of SPIE*, pp. 281–289, Barcelona, Spain, September 2004.
- [23] P.-T. Yap, R. Paramesran, and S.-H. Ong, "Image analysis by Krawtchouk moments," *IEEE Transactions on Image Processing*, vol. 12, no. 11, pp. 1367–1377, 2003.
- [24] M. R. Teague, "Image analysis via the general theory of moments," *Journal of the Optical Society of America*, vol. 70, no. 8, pp. 920–930, 1980.
- [25] C.-M. Pun and M.-C. Lee, "Log-polar wavelet energy signatures for rotation and scale invariant texture classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 590–603, 2003.
- [26] G. Strang and T. Nguyen, *Wavelets and Filter Banks*, Wellesley-Cambridge Press, Wellesley, Mass, USA, 1996.

- [27] D. W. Ritchie, *Parametric protein shape recognition*, Ph.D. thesis, University of Aberdeen, Scotland, UK, 1998.
- [28] L. E. Raileanu and K. Stoffel, "Theoretical comparison between the gini index and information gain criteria," *Annals of Mathematics and Artificial Intelligence*, vol. 41, no. 1, pp. 77–93, 2004.
- [29] P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser, "The Princeton Shape Benchmark," in *Proceedings of the Shape Modeling International (SMI '04)*, pp. 167–178, Genova, Italy, June 2004.
- [30] Princeton Shape Benchmark, <http://shape.cs.princeton.edu/search.html>.
- [31] P. Daras, D. Zarpalas, D. Tzovaras, and M. G. Strintzis, "Shape matching using the 3D radon transform," in *Proceedings of the 2nd International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT '04)*, pp. 953–960, Thessaloniki, Greece, September 2004.
- [32] D. V. Vranić, "An improvement of rotation invariant 3D-shape descriptor based on functions on concentric spheres," in *Proceedings of the IEEE International Conference on Image Processing (ICIP '03)*, vol. 3, pp. 757–760, Barcelona, Spain, September 2003.

**Dimitrios Zarpalas** was born in Thessaloniki, Greece, in 1980. He is a Ph.D. candidate at Northeastern University. He received the Diploma degree and the M.S. degree in electrical and computer engineering from the Aristotle University of Thessaloniki, Greece, in 2003, the Pennsylvania State University, USA, in 2006, respectively. His main research interests include search and retrieval of 3D objects, 3D object recognition, and medical image processing. He is a Member of the Technical Chamber of Greece.



**Petros Daras** was born in Athens, Greece, in 1974. He is a Researcher Grade D' at the Informatics and Telematics Institute. He received the Diploma degree in electrical and computer engineering, the M.S. degree in medical informatics, and the Ph.D. degree in electrical and computer engineering from the Aristotle University of Thessaloniki, Greece, in 1999, 2002, and 2005, respectively. His main research interests include computer vision, search and retrieval of 3D objects, the MPEG-4 standard, peer-to-peer technologies, and medical informatics. He has been involved in more than 10 European and national research projects. He is a Member of the Technical Chamber of Greece.



**Apostolos Axenopoulos** was born in Thessaloniki, Greece, in 1980. He is an Associate Researcher at the Informatics and Telematics Institute. He received the Diploma degree in electrical and computer engineering and the M.S. degree in advanced computing systems from the Aristotle University of Thessaloniki, Greece, in 2003 and 2006, respectively. His main research interests include 3D content-based search and retrieval. He is a Member of the Technical Chamber of Greece.



**Dimitrios Tzovaras** received the Diploma degree in electrical engineering and the Ph.D. degree in 2D and 3D image compression from Aristotle University of Thessaloniki, Thessaloniki, Greece, in 1992 and 1997, respectively. He is a Senior Researcher in the Informatics and Telematics Institute of Thessaloniki. Prior to his current position, he was a Senior Researcher on 3D imaging at the Aristotle University of Thessaloniki. His main research interests include virtual reality, assistive technologies, 3D data processing, medical image communication, 3D motion estimation, and stereo and multiview image sequence coding. His involvement with those research areas has led to the coauthoring of more than 35 papers in refereed journals and more than 80 papers in international conferences. He has served as a regular Reviewer for a number of international journals and conferences. Since 1992, he has been involved in more than 40 projects in Greece, funded by the EC, and the Greek Secretariat of Research and Technology. He is an Associate Editor of the EURASIP Journal on Advances in Signal Processing and a Member of the Technical Chamber of Greece.



**Michael G. Strintzis** received the Diploma degree in electrical engineering from the National Technical University of Athens, Athens, Greece, in 1967, and the M.A. and Ph.D. degrees in electrical engineering from Princeton University, Princeton, NJ, in 1969 and 1970, respectively. He then joined the Electrical Engineering Department at the University of Pittsburgh, Pittsburgh, Penn, where he served as Assistant Professor (1970–1976) and Associate Professor (1976–1980). Since 1980, he has been Professor of electrical and computer engineering at the University of Thessaloniki, Thessaloniki, Greece, and, since 1999, Director of the Informatics and Telematics Research Institute, Thessaloniki. His current research interests include 2D and 3D image coding, image processing, biomedical signal and image processing, and DVD, and Internet data authentication and copy protection. He has served as Associate Editor for the IEEE Transactions on Circuits and Systems for Video Technology since 1999. In 1984, he was awarded one of the Centennial Medals of the IEEE.

