# 3D Object Retrieval using an Efficient and Compact Hybrid Shape Descriptor

Panagiotis Papadakis, Ioannis Pratikakis, Theoharis Theoharis, Georgios Passalis, Stavros Perantonis

# 3D Object Retrieval using an Efficient and Compact Hybrid Shape Descriptor

P. Papadakis[1,2], I. Pratikakis[1], T. Theoharis[2], G. Passalis[2] and S. Perantonis[1]

[1]Computational Intelligence Laboratory, Institute of Informatics and Telecommunications,
National Center for Scientific Research Demokritos, Agia Paraskevi, Athens, Greece
[2]Computer Graphics Group, Department of Informatics and Telecommunications,
National and Capodistrian University of Athens, Greece

**Abstract**

*We present a novel 3D object retrieval method that relies upon a hybrid descriptor which is composed of 2D features based on depth buffers and 3D features based on spherical harmonics. To compensate for rotation, two alignment methods, namely CPCA and NPCA, are used while compactness is supported via scalar feature quantization to a set of values that is further compressed using Huffman coding. The superior performance of the proposed retrieval methodology is demonstrated through an extensive comparison against state-of-the-art methods on standard datasets.*

Categories and Subject Descriptors (according to ACM CCS): I.5.4 [Pattern Recognition]: Computer Vision, H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval

## 1. Introduction

In order to efficiently search and retrieve 3D models, content-based retrieval methods are necessary that use information extracted from the shape of a 3D model. In particular, the search process is initiated by a query 3D model provided by a user and the system retrieves 3D models that are considered similar to the query. The information that is used to represent each 3D model is encoded in a signature known as *shape descriptor*. The shape descriptor should capture the shape of the model in a discriminative way in order to support effective comparison against models of the database. To this effect, it should be insensitive to variations in scale, translation and rotation (similarity transformations) of a 3D model which are not considered to affect the measure of similarity. Moreover, it should have a compact structure to reduce storage requirements and support fast computation of the similarity measure between models.

In this paper, we focus on the development of a 3D object retrieval methodology that is based upon a hybrid shape descriptor that provides top discriminative power and has minimal storage requirements. Based on the previous work of Passalis et al. [PTK07], we extract 2D features from a depth-buffer based representation of a model and 3D features (spherical harmonic coefficients) from a spherical function based representation as in the work of Papadakis et al. [PPPT07]. The novelty of our approach consists of: (i) The two kinds of features combined to produce an integrated hybrid 3D shape descriptor; (ii) Invariance to similarity transformations achieved using CPCA and NPCA [PPPT07], where the two methods are employed in parallel to alleviate the rotation invariance problem, for both 2D and 3D features; (iii) To reduce the size of the descriptor, scalar quantization is applied to the set of coefficients and the resulting set of symbols is Huffman coded, producing a highly compact descriptor. We demonstrate the increased performance of the proposed method through a comparison against state-of-the-art approaches on standard benchmarks that clearly indicates its superior overall performance.

## 2. Related Work

In this section, we give a brief description of the related work in the area of content-based 3D object retrieval methods. Extended state-of-the-art surveys can be found in [TV04], [BKS*05], [IJL*05] and [SMKF04].

Content-based 3D model retrieval methods may be classified into three main categories according to the spatial di-

mensionality of the information used, namely 2D, 3D and their combination (2D/3D).

In the first category, shape descriptors are generated from images-projections which may be contours, silhouettes, depth buffers or other kinds of 2D information. Thus similarity is measured using 2D matching techniques used for 2D shape search. Chen et al. [CTSO03] proposed the Light Field descriptor, which is comprised of Zernike moments and Fourier coefficients computed on a set of projections taken from the vertices of a dodecahedron. Vranic [Vra04] proposed a shape descriptor where features are extracted from depth buffers produced by six projections of the model, one for each side of a cube which encloses the model. In the same work, the Silhouette-based descriptor is proposed which uses the silhouettes produced by the three projections taken from the Cartesian planes. Zarpalas et al. [ZDA*07] introduce the Spherical Trace Transform descriptor, which is an extension to 3D shape matching of the 2D Trace Transform. Among a variety of 2D features which are computed over a set of planes intersecting the model, one of the components of the shape descriptor are 2D Krawtchouk moments which significantly improve the overall performance. In [PTK07], Passalis et al. proposed PTK, a depth buffer based descriptor which uses parallel projections to capture the model's thickness and an alignment scheme that is based on symmetry.

In the second category, shape descriptors are extracted from the 3D shape-geometry of the 3D model and the similarity is measured using appropriate representations in the spatial domain or in the spectral domain. Ankerst et al. [AKKS99] proposed the Shape Histograms descriptor where 3D space can be divided into concentric shells, sectors, or both and for each part the model's shape distribution is computed, giving a sum of histograms bins. Vranic [Vra04] proposed the Ray-based descriptor which characterizes a 3D model by a spherical extent function capturing the furthest intersection points of the model's surface with rays emanating from the origin. Spherical harmonics or moments can be used to represent the spherical extent function. A generalization of the previous approach is also described in [Vra04], that uses several spherical extent functions of different radii. The GEDT descriptor proposed by Kazhdan et al. [KFR03] is a volumetric representation of the Gaussian Euclidean Distance Transform of the 3D model's volume, expressed by norms of spherical harmonic frequencies. Shape distributions proposed by Osada et al. [OFCD01], is a class of methods where several features can be computed for a random set of points belonging to the model, according to the selection of an appropriate function. In [PPPT07], the CRSP descriptor was proposed which uses a combination of CPCA and NPCA to alleviate the rotation invariance problem and describes a 3D model using an alternative spherical function based representation which is expressed by spherical harmonics.

Besides the previous categories, combinations of different methods have been considered in order to enhance the overall performance, which comprise a third category. Vranic developed a hybrid descriptor called DSR472 [Vra04, Vra05], that consists of the Silhouette, Ray and Depth buffer based descriptors, which are combined linearly by fixed weights. The approach of Bustos et al. [BKS*04] assumes that the classification of a particular dataset is given, in order to estimate the expected performance of the individual shape descriptors for the submitted query and automatically weigh the contribution of each method. However, in the general case, the classification of a 3D model dataset is not fixed since the content of a 3D model dataset is not static. In the context of partial shape matching, Funkhouser et al. [FS06] use the predicted distinction performance of a set of descriptors based on a preceding training stage and perform a priority driven search in the space of feature correspondences to determine the best match of features between a pair of models. The disadvantages of this approach is its time complexity which is prohibitive for online interaction as well as the storage requirements for the descriptors of all the models in the database.

## 3. Hybrid 3D Shape descriptor extraction

In this Section, we describe the distinct stages for the extraction of the proposed hybrid 3D shape descriptor, namely: (i) pose normalization; (ii) 2D feature extraction; (iii) 3D feature extraction; (iv) feature compression.

### 3.1. Pose Normalization

Before the extraction of the hybrid features, it is necessary to normalize the pose of a 3D model as the position, orientation and size parameters of a 3D model are arbitrarily set. Thereupon, the final 3D shape descriptor should be invariant under translation, rotation and scaling.

First, we normalize each 3D model for translation by computing its centroid using CPCA [Vra04] and translating the model so that the centroid coincides with the origin.

We address the rotation invariance problem, by following the same approach as in [PPPT07] where two alignment methods are used, namely CPCA and NPCA. This is a meaningful choice, since by using CPCA and NPCA we consider both the surface area distribution and the surface orientation distribution that both characterize the rotation of a model's surface and as mentioned in [PPPT07], there are cases where one method performs better than the other and vice versa. In particular, NPCA works better for objects with well defined flat surfaces while CPCA is better at aligning objects where their surface area distribution better discriminates the principal axes of the object.

Thus, given a translation normalized 3D model we compute the CPCA and NPCA aligned version of the model
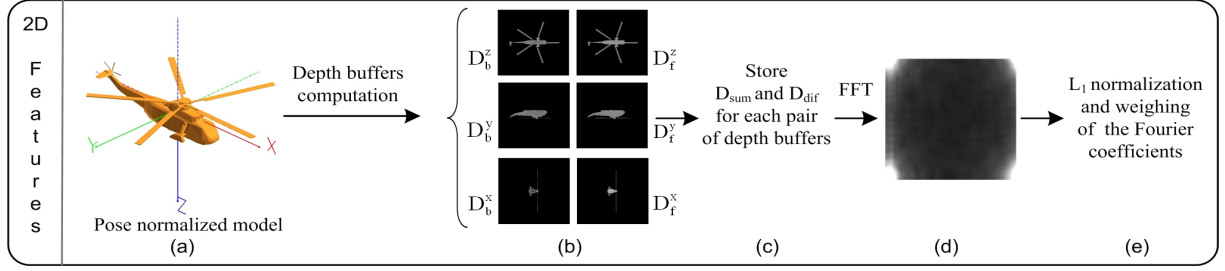
**Figure 1:** *Stages of the 2D feature extraction procedure.*

which produces two sets of hybrid features that comprise the hybrid shape descriptor. In Section 3.4 we explain how these sets are used to measure the similarity between two models.

Scaling invariance is achieved by normalizing the 3D and 2D features respectively to their unit $L_1$ norm since the computed features are proportional to the model's scale. Therefore, by normalizing the scale of the features we are in fact normalizing the scale of the corresponding model. The advantage is that this approach does not depend on the polygonal resolution of the 3D model but on the number of features which is set to a fixed value for all models.

### 3.2. 2D Feature Extraction

Based on the work of [PTK07], we use a depth-buffer based representation of a 3D model to compute a set of 2D features. In Figure 1, the main steps for the computation of the 2D features are depicted.

A depth buffer is a regular grid of scalar values that holds the distance between the model's surface and a given clipping plane. Using as clipping planes the faces of a cube having constant size and center at the origin, we acquire six depth buffers by rendering the model using an orthographic projection. In Figure 1 (b), the six depth buffers of a 3D model are shown.

Using a constant sized cube means that the 3D model may not necessarily lie completely inside the cube. However, this approach is better than using the bounding cube that may contain the model in small resolution in the presence of outlying parts of the model. Thereupon, the edge of the cube $E$, is set empirically to a fixed value such that the majority of 3D models lies completely inside the cube, while having large enough scale to produce descriptive depth buffers. The value of $E$ is set to $E = 6 * d_{mean}$ where $d_{mean}$ is the average distance of the model's surface from its centroid and it is easily computed using the diagonal elements of the CPCA covariance matrix as in [PPPT07].

Instead of storing the back $D_b$ and front $D_f$ depth buffers along each direction, we store their difference $D_{dif}=D_f-D_b$ and their sum $D_{sum}=D_f+D_b$ (Figure 1 (c)). The two new depth buffers ($D_{dif}$, $D_{sum}$) require the same storage space

as the original two ($D_f$, $D_b$) and there is no information loss since we can produce the one pair from the other. The difference lies in the fact that $D_{dif}$ holds the thickness of the model at each pixel, which is a characteristic of the model. This value is insensitive to errors in the translation of the model along the direction of projection, thus alleviating errors introduced in the translation normalization step. $D_{sum}$ has no physical meaning and it is used only as a complement to $D_{dif}$ and therefore is considered less important. This is reflected on the weight selection as shown later.

To produce the 2D features, a spectral transformation is applied to each pair of depth buffers along each direction. The discrete 2D Fourier transform (DFT) of a depth buffer D of size S denoted as ($F_d$) is given by:

$$F_d(m,n) = \sum_{x=0}^{S-1}\sum_{y=0}^{S-1} D(x,y) \cdot exp\left(-2j\pi\frac{mx+ny}{S}\right) \quad (1)$$

To reduce the computational cost, we implement the DFT, by using the Fast Fourier Transform algorithm. As shown in Figure 1 (d) most information is concentrated on the four corners of the image. For an SxS depth buffer, we keep KxK coefficients as follows:

$$F(m,n) = |F_d(m,n)| + |F_d(m,S-n-1)| +$$
$$|F_d(S-m-1,n)| + |F_d(S-m-1,S-n-1)| \quad (2)$$

for $m,n = 0,1,...,K-1$ where we sum the norm of the coefficients from all four corners to have a reflection invariant representation. In our implementation we use depth buffers of resolution S=128 and set K=48. In the sequel, we normalize the coefficients to the unit $L_1$ norm and weigh the F(m,n) coefficients inversely proportionally to their degree, since lower frequency coefficients are considered to have more information compared to higher frequency coefficients, which are more sensitive to noise.

A final weighting is applied to the coefficients, based on the following observation. Each pair of depth buffers is perpendicular to a specific coordinate axis and encodes information from two of the principal directions of the 3D model which are computed from the rotation normalization step. Since the principal directions are sorted according to their degree of significance, we can intuitively assert that a pair
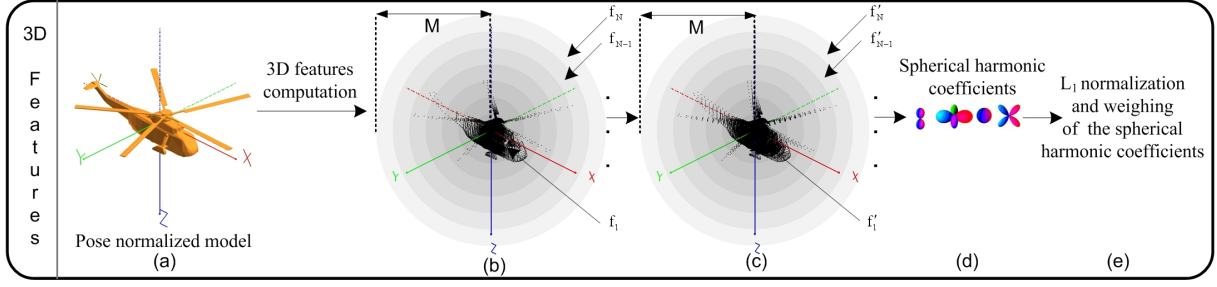
**Figure 2:** *Stages of the 3D feature extraction procedure.*

of buffers encodes an amount of information proportional to the principal directions that it encodes, thus the coefficients of the corresponding depth buffers are weighted according to this idea. The respective weights are set to fixed values that were determined empirically. In particular, if $w_{ij}$ is the weight applied to the coefficients encoding the $i^{th}$ and $j^{th}$ principal direction then $w_{12}=3$, $w_{13}=2$ and $w_{23}=1$ for the $D_{dif}$ coefficients and $w_{12}=1$, $w_{13}=0.7$ and $w_{23}=0.3$ for the $D_{sum}$ coefficients. The final 2D feature set is the concatenation of the coefficients of the six depth buffers. The 2D features set of model i that is aligned either by CPCA or NPCA, is denoted as $2Df_i^j$ where $j \in \{CPCA, NPCA\}$.

### 3.3. 3D Feature Extraction

Based on the work of [PPPT07] we extract the 3D features using a spherical function-based representation of the 3D model and compute the spherical harmonics transform for each spherical function. In Figure 2, the overall procedure for the extraction of the 3D features of the hybrid 3D shape descriptor is depicted.

A spherical function describes a bounded area of the model, defined by a lower and an upper radius which delimit a spherical shell. This shell is the area in which the underlying surface of the model is represented by the equivalent spherical function points. The region corresponding to the $r^{th}$ spherical function is the spherical shell defined in the interval $[l_r, l_{r+1})$ where the $l_r$ boundary is given by:

$$l_r = \begin{cases} 0 & , r = 1 \\ \frac{r-1/2}{N} \cdot M & , 1 < r \leq N+1 \end{cases} \quad (3)$$

where $M$ is a weighting factor that determines the radius of the largest sphere and $N$ is the number of spherical functions. Multiplying by $M$ in Eq. (3) ensures that the majority of 3D models lies completely inside the sphere with the largest radius while having large enough scale to give descriptive 3D features. Therefore, the value of $M$ is set to half the length of the edge of the cube which was used in the 2D features extraction step, thus $M = E/2$.

The representation of a 3D model as a set of spherical

functions is similar to projecting approximately equidistant parts of the 3D model to concentric spheres of increasing radius. This is done by computing the intersections of the surface of the model with rays emanating from the origin in the $(\theta_j, \varphi_k)$ directions, denoted as ray$(\theta_j, \varphi_k)$, where $\theta_j$ denotes the latitude and $\varphi_k$ denotes the longitude coordinate, j, k=0,1,...2B-1 and B is the sampling bandwidth. In the direction of ray$(\theta_j, \varphi_k)$ inside the spherical shell $r(r \leq N)$, we compute all the intersections of the model's surface with this ray. The value of the corresponding spherical function point is set to the distance of the furthest intersection point within the boundaries of the $r^{th}$ spherical function, or to zero if there is no intersection. Thus, if $I(r, \theta_j \varphi_k)$ denotes the distance of the furthest intersection point along ray$(\theta_j, \varphi_k)$ within the boundaries of the $r^{th}$ spherical function, then the N spherical functions representing the 3D model are given by:

$$f_r(\theta_j, \varphi_k) = I(r, \theta_j, \varphi_k) \quad (4)$$

where $f_r : S^2 \to [l_r, l_{r+1}) \bigcup \{0\}, r = 1, 2, ..., N$ and $S^2$ denotes the points $(\theta_j, \varphi_k)$ on the surface of a sphere with unit radius. These functions are shown in Figure 2 (b) for the "helicopter" example model, wherein the corresponding intersections are depicted as black dots.

Practically, if the model is viewed from the $(\theta_j, \varphi_k)$ direction then the part of the model along ray$(\theta_j, \varphi_k)$ which is closer to the origin than $I(D, \theta_j, \varphi_k)$ is occluded by the intersection point corresponding to $I(D, \theta_j, \varphi_k)$. As shown in [PPPT07], including the occluded points into the spherical functions increases discriminative power significantly. Thus, we take into account a modified version of the spherical functions $f_r$. In particular, let $I(D, \theta_j, \varphi_k)$, corresponding to the furthest intersection point on ray$(\theta_j, \varphi_k)$, be assigned to the spherical function point $f_D(\theta_j, \varphi_k)$. Then all $f_r(\theta_j, \varphi_k)$, where $r < D$ are assigned values by setting:

$$f_r(\theta_j, \varphi_k) = r \cdot \frac{M}{N}, \quad \forall r < D \quad (5)$$

which stands for the intersections of the ray$(\theta_j, \varphi_k)$ with all the spheres of radius less than D. Thus, we get a set of modified spherical functions $f_r'$, shown in Figure 2 (c), that are

formally denoted as:

$$f_r^{'}(\theta_j, \varphi_k) = \begin{cases} I(r, \theta_j, \varphi_k) & , I(D, \theta_j, \varphi_k) \neq 0 \text{ and } r = D \\ r \cdot \frac{M}{N} & , I(D, \theta_j, \varphi_k) \neq 0 \text{ and } r < D \\ 0 & , \text{ otherwise} \end{cases} \quad (6)$$

At the final step we compute the 3D features as the spherical harmonic coefficients of each spherical function. The spherical harmonic coefficients can be computed using the spherical harmonics transform (SHT) [JRKM03]. We compute the SHT at constant bandwidth $B = 40$ for all models and choose to store only the coefficients up to the $L^{th}$ degree (we set L=16), since spherical harmonics of higher degree are more sensitive to noise. Using properties of spherical harmonics we can ensure axial flipping invariance. Flipping a model with respect to an axis may only change the sign of the real or the imaginary part of a coefficient thus we use their absolute values. After normalizing the coefficients to the unit $L_1$ norm, each coefficient is weighted inversely proportionally to its degree. The final 3D feature set is the concatenation of the spherical harmonic coefficients of the N spherical functions. The 3D features set of model i that is aligned either by CPCA or NPCA, is denoted as $3Df_i^j$ where $j \in \{CPCA, NPCA\}$.

### 3.4. Similarity in the Hybrid Scheme

As explained at Section 3.1, the 2D and 3D features are computed for two alternative rotation normalized versions of a model. Thus, the final hybrid 3D shape descriptor $\mathbf{s}_i$ of a model i is the concatenation of the 2D and 3D features for each aligned version of the 3D model, giving:

$$\mathbf{s}_i = \left( 2Df_i^{CPCA}, 2Df_i^{NPCA}, 3Df_i^{CPCA}, 3Df_i^{NPCA} \right) \quad (7)$$

To compare the descriptors $\mathbf{s}_1$ and $\mathbf{s}_2$ of two models we adopt the following schema to compute their distance:

$$Dist(\mathbf{s}_1, \mathbf{s}_2) = dist_{2Df} + dist_{3Df} \quad (8)$$

where $dist_{2Df}$ and $dist_{3Df}$ is the distance between the 2D and 3D features, respectively, denoted as:

$$dist_{2Df} = min_j \left( L_1 \left( 2Df_1^j, 2Df_2^j \right) \right)$$
$$dist_{3Df} = min_j \left( L_1 \left( 3Df_1^j, 3Df_2^j \right) \right) \quad (9)$$

where $j \in \{CPCA, NPCA\}$ and $L_1$ is the Manhattan distance between the corresponding features.

### 4. Feature Compactness

Using two kinds of features (2D and 3D), as well as two different aligned versions of a 3D model for each kind, the issue of the descriptor's storage requirements is raised.

In particular, the 2D features is a set of $KxK = 48 * 48 = 2304$ coefficients for each of the six depth buffers, for each

aligned version of the model. Thus the 2D features component consists of $2 * 6 * 2304 = 27648$ floating point numbers. The 3D features is a set of $L * \frac{(L+1)}{2} * 2 = 272$ floating point numbers for each of the N=24 spherical functions, for each aligned version of the model. Thus the 3D features component consists of $272 * 24 * 2 = 13056$ floating point numbers. In total, the hybrid descriptor consists of 40704 floating point numbers that require 159 Kbytes of storage (4 bytes each).

Since both 2D and 3D features are normalized to their respective unit $L_1$ norm, the values of the hybrid features lie within the [0,1] space. We apply uniform scalar quantization and divide the [0,1] space to $2^{16} = 65536$ intervals of size $\Delta = 1/65536$. The $i^{th}$ interval, $i = 1, 2, ..., 65536$ corresponds to the $[(i-1)\Delta, i\Delta)$ space and is represented by the mean value of the lower and upper bound of the interval. Thus, each floating point number is represented by the closest quantized value (symbol), according to the nearest neighbor rule. Figure 3 shows the probabilities $P_i$ of appearance in the hybrid 3D shape descriptor, of the first 1024 symbols.

The probability of each symbol was computed by counting its frequency from the set of descriptors of the datasets used in the experiments shown in Section 5. The overall probability of a coefficient to be quantized to one of the first 1024 symbols is more than 99,9%, while the probability of each remaining symbol is less than $P_0 = 10^{-6}$. Thereupon, we use only 10 bits to encode $2^{10}$ quantized symbols and the last symbol (i=1024) is used to cover the $[1023\Delta, 1]$ space. After this, the hybrid descriptor requires $\frac{40704 * 10}{8 * 1024} \simeq 50$ Kbytes.

While scalar quantization is a lossy compression method, the employed quantization scheme does not affect the discrimination ability of the hybrid descriptor. In particular, by
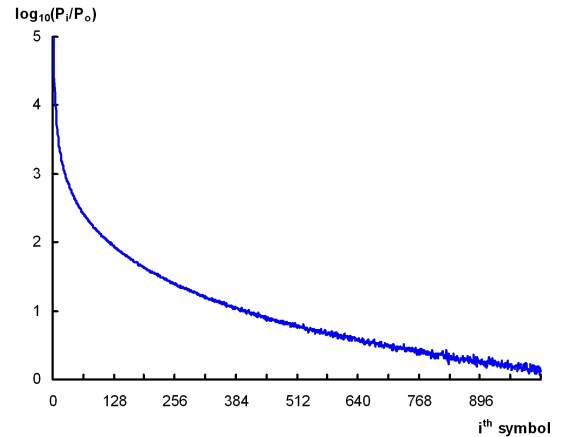


**Figure 3:** *Probabilities $P_i$ of quantized values of the hybrid features ($P_o = 10^{-6}$).*

evaluating the performance of the hybrid descriptor with and without scalar quantization the difference in average precision was less than 0.5%, which is negligible taking into account the achieved reduction in storage requirements.

To further reduce the size of the hybrid descriptor, we code the selected symbols using Huffman coding [PS02] which considers the probability of each symbol in order to assign a bit sequence whose length is inversely proportional to the symbol's probability. In Table 1, the Huffman codes for the first and last 3 symbols are shown. Using the computed Huffman codes, we take $2,373$ bits per coefficient on average, or equivalently, $11,79$ KBytes per descriptor. Thus, we achieve an average compression ratio $c = (159 - 11,79)/159 \simeq 92,6\%$.

Huffman coding is a lossless compression method, nevertheless, coding and decoding the symbols adds computational cost. In our case, the overall cost of employing Huffman coding was dominated by the hard disk access time, therefore, the overall time complexity of object retrieval is not considerably affected.

**Table 1**. *Huffman codes of the first and last 3 symbols of the hybrid descriptor.*

| Symbol i | Computed Huffman code |
|----------|----------------------|
| 1 | 1 |
| 2 | 00 |
| 3 | 0110 |
| ⋮ | ⋮ |
| 1022 | 0100100000010111110 |
| 1023 | 0101010101111000010 |
| 1024 | 0111110011 |

## 5. Experimental Evaluation

In this section, we give the results of an extensive evaluation of the proposed Hybrid descriptor and compare it against other state-of-the-art methods. We tested the performance using the following 3D model datasets:

(i.) The test set of the Princeton Shape Benchmark (PSB) (907 models) [SMKF04].

(ii.) The classified set of the National Taiwan University Benchmark (NTU) (549 models) (http://3D.csie.ntu.edu.tw/).

(iii.) The classified set of the CCCC set (472 models) (http://merkur01.inf.uni-konstanz.de/CCCC/).

(iv.) The MPEG-7 set (1300 models) (http://merkur01.inf.uni-konstanz.de/CCCC/).

(v.) The Engineering Shape Benchmark (ESB) set (866 models) [JKIR06].

The datasets (i.)-(iv) are generic while the last dataset is a domain specific case consisting only of engineering models.

To depict the performance we use precision-recall diagrams. Recall is the ratio of relevant to the query retrieved
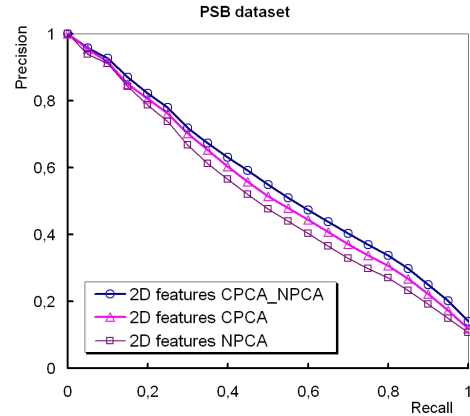


**Figure 4:** *Precision-recall plot showing the perfomance of the 2D features when using either CPCA or NPCA for rotation normalization compared to the usage of both methods.*

models to the total number of relevant models while precision is the ratio of relevant to the query retrieved models to the number of retrieved models.

In Figure 4, we show the increased performance of the 2D features when using both CPCA and NPCA to alleviate the rotation invariance problem compared to using either one of the two alignment methods.

In Figure 5 we show the complementarity in the discrimination power between the selected 2D and 3D features. In Figure 5 (a), the vertical axis shows the difference between the Discounted Cumulative Gain (DCG) score of the 3D and 2D features for the NTU dataset. The DCG score is a commonly used quantitative performance measure [SMKF04] which is computed by weighing correct results according to their position in the retrieval list (the maximum score is 100%). The horizontal axis is an id-number corresponding to a specific class of 3D models as shown in Figure 5 (b). From Figure 5, we can conclude that the chosen 2D and 3D features have a complementary behavior, that is, there are classes where the 3D features are more discriminative than the 2D features and vice versa. This is also true for the rest of the datasets. Therefore, using both the selected 2D and 3D features is a meaningful choice that produces an effective hybrid descriptor.

We compare the proposed hybrid descriptor against the following state-of-the-art descriptors:

1. The Light Field (LF) descriptor [CTSO03].
2. The spherical harmonic representation of the Gaussian Euclidean Distance Transform (SH-GEDT) descriptor [KFR03].
3. The DSR472 descriptor [Vra04].

Each descriptor was computed using the executables that the authors provide at their web-pages.
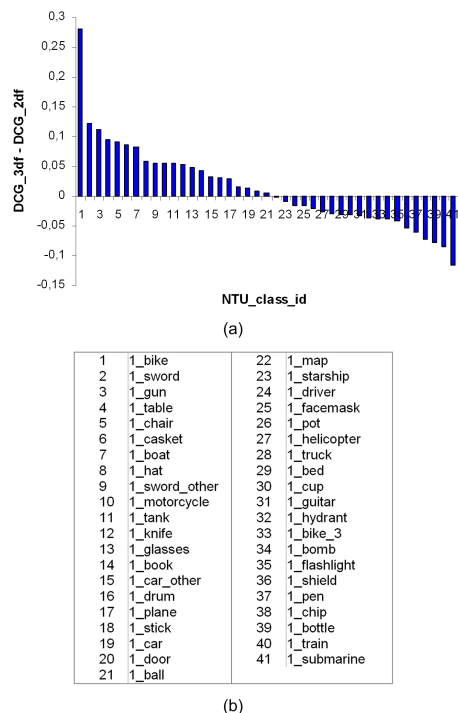
(a)

| | | | |
|---|---|---|---|
| 1 | 1_bike | 22 | 1_map |
| 2 | 1_sword | 23 | 1_starship |
| 3 | 1_gun | 24 | 1_driver |
| 4 | 1_table | 25 | 1_facemask |
| 5 | 1_chair | 26 | 1_pot |
| 6 | 1_casket | 27 | 1_helicopter |
| 7 | 1_boat | 28 | 1_truck |
| 8 | 1_hat | 29 | 1_bed |
| 9 | 1_sword_other | 30 | 1_cup |
| 10 | 1_motorcycle | 31 | 1_guitar |
| 11 | 1_tank | 32 | 1_hydrant |
| 12 | 1_knife | 33 | 1_bike_3 |
| 13 | 1_glasses | 34 | 1_bomb |
| 14 | 1_book | 35 | 1_flashlight |
| 15 | 1_car_other | 36 | 1_shield |
| 16 | 1_drum | 37 | 1_pen |
| 17 | 1_plane | 38 | 1_chip |
| 18 | 1_stick | 39 | 1_bottle |
| 19 | 1_car | 40 | 1_train |
| 20 | 1_door | 41 | 1_submarine |
| 21 | 1_ball | | |

(b)

**Figure 5:** *(a) Difference between the DCG of the 3D (3df) and 2D (2df) features, among the classes of the NTU dataset; (b) The correspondence between class ids and class names.*

In Figure 6 (a), we give the precision-recall diagrams comparing the proposed Hybrid descriptor against the other descriptors in the generic datasets and in Figure 6 (b) we show the performance for the case of engineering models. It is evident that the average precision of the proposed Hybrid descriptor is significantly higher compared to the other methods in all generic datasets as well as in the ESB dataset.

To further quantify the performance of the compared methods, we use the following quantitative measures:

- Nearest Neighbor (NN): The percentage of queries where the closest match belongs to the query's class.
- First Tier (FT) : The recall for the $(k-1)$ closest matches were $k$ is the cardinality of the query's class.
- Second Tier (ST) : The recall for the $2(k-1)$ closest matches were $k$ is the cardinality of the query's class.

The above measures range from 0%-100% and higher values indicate better performance.

In table 2, we give the respective scores of each method for the datasets of Figure 6. Apparently, the proposed Hybrid descriptor outperforms all compared methods with respect to the quantitative measure scores as well.

The proposed Hybrid descriptor has the additional advantage of short extraction and comparison times. Using a 2.0
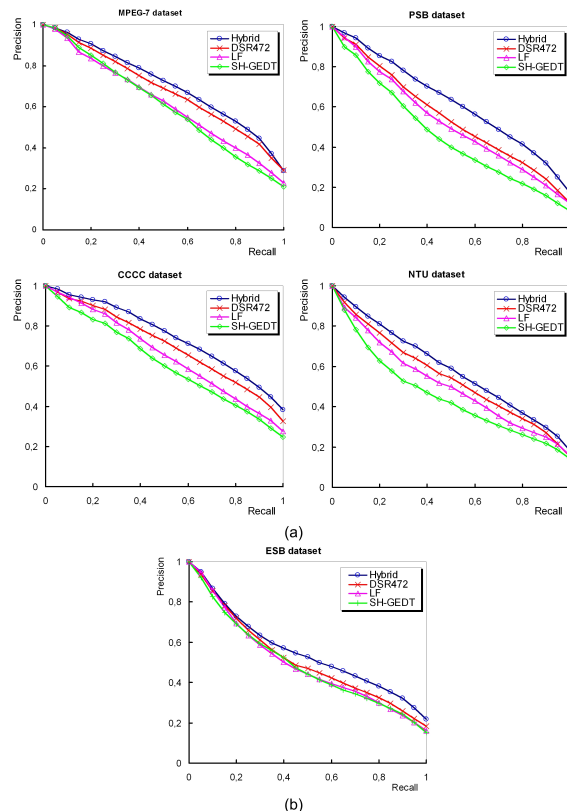


(a)



(b)

**Figure 6:** *Precision-recall plots of the proposed Hybrid, DSR472, LF and SH-GEDT descriptors for: (a) The MPEG-7, PSB, CCCC and NTU datasets and (b) the ESB dataset.*

Ghz AMD CPU, the average extraction and one-to-one comparison time is 1.05 s and 0.17 ms, respectively.

## 6. Conclusions

In this paper, a 3D object retrieval method is proposed that uses 2D features extracted from a depth buffer-based representation and 3D features from the spherical harmonics transformation of a spherical functions-based representation. The 3D model is normalized for rotation using two alternative alignment methods, namely CPCA and NPCA. The hybrid features are compressed using scalar quantization and Huffman coding. The result is a highly compact descriptor that achieves top performance as demonstrated through an extensive evaluation against state-of-the-art descriptors on standard datasets.

## 7. Acknowledgement

**Table 2**. *Quantitative measures scores of the proposed Hybrid, DSR472, LF and SH-GEDT method for the MPEG-7, PSB, CCCC, NTU and ESB dataset.*

| Dataset | Method | NN (%) | FT (%) | ST (%) |
|---------|--------|--------|--------|--------|
| MPEG-7 | Hybrid | 86.1 | **59.6** | **70.7** |
|         | DSR472 | **86.4** | 57.7 | 67.7 |
|         | LF | 86.0 | 53.8 | 63.8 |
|         | SH-GEDT | 83.7 | 50.3 | 59.4 |
| PSB | Hybrid | **74.2** | **47.3** | **60.6** |
|     | DSR472 | 65.8 | 40.4 | 51.3 |
|     | LF | 64.2 | 37.5 | 48.4 |
|     | SH-GEDT | 55.3 | 31.0 | 41.4 |
| CCCC | Hybrid | **87.4** | **60.2** | **75.8** |
|      | DSR472 | 82.8 | 55.6 | 70.0 |
|      | LF | 79.8 | 50.2 | 63.1 |
|      | SH-GEDT | 75.7 | 45.9 | 59.9 |
| NTU | Hybrid | **76.2** | **46.6** | **59.1** |
|     | DSR472 | 71.9 | 42.7 | 55.4 |
|     | LF | 70.0 | 39.0 | 50.1 |
|     | SH-GEDT | 58.8 | 33.9 | 46.3 |
| ESB | Hybrid | **82.9** | **46.5** | **60.5** |
|     | DSR472 | 82.3 | 41.7 | 55.0 |
|     | LF | 82.0 | 40.4 | 53.9 |
|     | SH-GEDT | 80.3 | 40.1 | 53.6 |

## References

[AKKS99] ANKERST M., KASTENMULLER G., KRIEGEL H. P., SEIDL T.: Nearest neighbor classification in 3D protein databases. In *ISMB* (1999), pp. 34–43.

[BKS*04] BUSTOS B., KEIM D., SAUPE D., SCHRECK T., VRANIC D.: Automatic selection and combination of descriptors for effective 3D similarity search. In *IEEE Sixth Int. Symp. on Multimedia Software Engineering* (2004), pp. 514–521.

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

[CTSO03] CHEN D. Y., TIAN X. P., SHEN Y. T., OUHYOUNG M.: On visual similarity based 3D model retrieval. *Eurographics, Computer Graphics Forum* (2003), 223–232.

[FS06] FUNKHOUSER T., SHILANE P.: Partial matching of 3D shapes with priority-driven search. In *Fourth Eurographics symposium on Geometry processing* (2006), pp. 131–142.

[IJL*05] IYER N., JAYANTI S., LOU K., KALYANARAMAN Y., RAMANI K.: Feature-based similarity search in 3d object databases. *Computer-Aided Design 37*, 5 (2005), 509–530.

[JKIR06] JAYANTI S., KALYANARAMAN Y., IYER N., RAMANI K.: Developing an engineering shape benchmark for cad models. *Computer-Aided Design 38* (2006), 939–953.

[JRKM03] JR. D. H., ROCKMORE D., KOSTELEC P., MOORE S.: Ffts for the 2-sphere – improvements and variations. *Journal of Fourier Analysis and Applications 9*, 4 (2003), 341–385.

[KFR03] KAZHDAN M., FUNKHOUSER T., RUSINKIEWICZ S.: Rotation invariant spherical harmonic representation of 3D shape descriptors. In *Eurographics/ACM SIGGRAPH symposium on Geometry processing* (2003), pp. 156–164.

[OFCD01] OSADA R., FUNKHOUSER T., CHAZELLE B., DOBKIN D.: Matching 3D models with shape distributions. In *Int. Conf. on Shape Modeling and Applications* (2001), pp. 154–166.

[PPPT07] PAPADAKIS P., PRATIKAKIS I., PERANTONIS S., THEOHARIS T.: Efficient 3D shape matching and retrieval using a concrete radialized spherical projection representation. *Pattern Recognition 40*, 9 (2007), 2437–2452.

[PS02] PROAKIS J. G., SALEHI M.: *Communication Systems Engineering*. Prentice Hall, 2002.

[PTK07] PASSALIS G., THEOHARIS T., KAKADIARIS I. A.: Ptk: A novel depth buffer-based shape descriptor for three-dimensional object retrieval. *Visual Computer 23*, 1 (2007), 5–14.

[SMKF04] SHILANE P., MIN P., KAZHDAN M., FUNKHOUSER T.: The princeton shape benchmark. In *Shape Modeling International* (2004), pp. 167–178.

[TV04] TANGELDER J., VELTKAMP R.: A survey of content based 3d shape retrieval methods. In *Shape Modeling International* (2004), pp. 145–156.

[Vra04] VRANIC D. V.: *3D Model Retrieval*. PhD thesis, University of Leipzig, 2004.

[Vra05] VRANIC D.: Desire: a composite 3d-shape descriptor. 145–156.

[ZDA*07] ZARPALAS D., DARAS P., AXENOPOULOS A., TZOVARAS D., STRINTZIS M. G.: 3D model search and retrieval using the spherical trace transform. *EURASIP Journal on Advances in Signal Processing* (2007), Article ID 23912, 14 pages.