Multivariate Density-Based 3D Shape Descriptors

Ceyhun Burak Akgül[1,2], Bülent Sankur[1], Francis Schmitt[2], Yücel Yemez[3]

[1] *Boğaziçi University, Electrical and Electronics Engineering Dept., Istanbul, Turkey*
[2] *GET - Télécom Paris - CNRS UMR 5141, Paris, France*
[3] *Koç University, Computer Engineering Dept., Istanbul, Turkey*
{akgul,schmitt}@enst.fr, bulent.sankur@boun.edu.tr, yyemez@ku.edu.tr

Abstract

We address the 3D object retrieval problem using multivariate density-based shape descriptors. Considering the fusion of first and second order local surface information, we construct multivariate features up to five dimensions and process them by the kernel density estimation methodology to obtain descriptor vectors. We can compute these descriptors very efficiently using the fast Gauss transform algorithm. We also make use of descriptor level information fusion by concatenating descriptor vectors to increase their discrimination power further. To render the resulting descriptors storage-wise efficient, we develop two analytical tools, marginalization and probability density suppression, for descriptor dimensionality reduction. The experiments on two different databases, Princeton Shape Benchmark and Sculpteur, show that, boosted with both feature level and descriptor level information fusion, and powered with fast computational schemes, the density-based shape description framework enables effective and efficient 3D object retrieval.

1 Introduction

Density-based shape description is a framework to extract 3D shape descriptors from local surface features characterizing the object geometry [1, 2]. The feature information is processed with the kernel methodology for density estimation (KDE) [10, 18]. The probability density function (pdf) of these local features estimated at chosen target points generates the shape descriptor vector. In other words, this density-based approach provides a mechanism to convert local shape evidences, using KDE, into a global shape description. Our recent work on density-based shape descriptors [1, 2] for 3D object retrieval has proven that this scheme is both computationally rapid and effective compared to other state-of-the-art descriptors.

The impact of local surface features within a density-

based framework can be analyzed for their individual contributions to retrieval as in [16]. However, they present a greater discrimination power if we consider them in ensembles, via fusion or concatenation. This analysis can be carried out at two levels. First at the feature level, one considers various scalar local features at the surface points and joins them to form multivariate features or feature vectors. The corresponding probability densities can then be estimated by the multivariate KDE approach. Second at the descriptor level, descriptors, i.e., univariate or multivariate sampled pdfs are concatenated to obtain larger descriptors.

In this work, we consider first and second order local surface features and present a comparative analysis of their multivariate combinations both at the feature-level and the descriptor-level for the 3D shape retrieval problem. The analysis is not brute-force since we reduce the combinatorial multitude of feature m-tuples by informed choices. We also make use of two analytical tools, called *marginalization* and *probability density suppression* for reducing the descriptor size.

The paper is organized as follows. In Section 2, we introduce the pdf-based descriptor design problem and discuss the related previous work. Section 3.1 defines a set of local features. We briefly describe kernel density estimation (KDE) in Section 3.2 and introduce tools to manipulate the pdf-based descriptors in Sections 3.3 and 3.4. The experiments in Section 4 demonstrate the impact of feature-level and descriptor-level combinations, show the limits of descriptor simplification and provide a comprehensive comparison with other descriptors.

2 Previous Work

The last decade has witnessed a growing interest in 3D shape description resulting in a plethora of methods for content-based retrieval. Here, we briefly review the descriptors that have had major impact in shape-based retrieval. Survey papers [4, 19, 20] present more extensive overviews.

Several descriptors, akin to the density-based framework

in this paper, fall under the category of histogram-based methods. Historically, the first such descriptors are Paquet and Rioux's univariate cord and angle histograms (CAH) [17] that are basically accumulators of radial distances and angles (with respect to a reference frame). In the shape distributions (SD) approach [16], Osada et al. have introduced the concept of a shape function, which is a scalar geometrical quantity computed via random sampling of the 3D surface. The collection of shape functions proposed in [16] includes surface point-to-fixed point distance (D1), surface point-to-surface point (D2) distance, area of the triangle formed by three surface points (D3), etc. Despite their simplicity and intuitive appeal, both CAH and SD fall short in providing fine grain discrimination for content-based retrieval. The poor retrieval performance of these approaches has been usually attributed to their global nature. The more recent generalized shape distributions (GSD) [15] have addressed this difficulty by a "3D" histogram where two dimensions account for local and global shape signatures and one for distances between local shape pairs. However, the improvement that GSD has provided was not sufficient to lift the histogram methodology to the discrimination level of its competitors. The shape histograms (SEC-SHELL) of Ankerst et al. [3], on the other hand, are constructed by accumulating the surface points in the bins (in the form of shells, sectors, or both) based on a nearest neighbor rule, providing better discrimination than the former ones. There also exists other count-and-accumulate type of schemes where the histogram aspect is less emphasized. Extended Gaussian images (EGI) [11] and 3D Hough Transform (3DHT) [25] are two such descriptors. While the former relies on mapping the surface normals onto a unit sphere, the latter is an accumulator of the local tangent plane parameters and can be viewed as a generalized version of EGI. Actually, the 3DHT descriptor is simple to implement and effective for retrieval [25].

The density-based framework provides a statistical and analytical formalism to the histogram-based methods for 3D shape description [1, 2]. The density-based setting, in principle, can accommodate local features of any dimension. The concomitant increase in computational complexity can be managed with the use of fast approximation algorithms such as the fast Gauss transform [24]. The ability of the density-based framework to process multivariate local surface information not only promotes it to the top rank in the league of histogram-based schemes [1], but also makes it a strong competitor to other state-of-the art methods.

In the present work, we provide experimental evidence for the above claim. It has been reported that global spherical harmonics descriptor (GSHD) [12] and radialized extent function (REXT) [22] yield the best retrieval performances among purely 3D shape description schemes [19]. As exemplified in Section 4, multivariate density-based de-

scriptors exhibit better discrimination than both of these two methods.

Apart from purely 3D and vector-based shape description paradigms [1, 2, 3, 11, 12, 15, 16, 17, 22, 25], there also exist others that rely on effective representations of 2D projections of 3D objects or on matching by feature correspondences. Interestingly, the 2D methods, namely, the light field descriptor (LFD) [5] and depth buffer images (DBI) [4] achieve better discrimination than any known 3D method at the cost of increased descriptor extraction time and storage size. Even more impressive retrieval performances have been reported using Funkhouser and Shilane's priority-driven search (PDS) method [8], which belongs to the paradigm of matching by feature correspondences. We cite this highly discriminating methodology for completeness, even if it relies on a philosophy that is completely different from the 3D descriptor approach. In Section 4.4, we list the retrieval performances of a selected group of the above-mentioned shape matching schemes to give the reader an idea about the performance landscape in content-based 3D model retrieval research.

3   Description Framework

A density-based descriptor of a 3D shape is defined as the sampled pdf of some surface feature, such as radial distance or direction. The feature is local to the surface patch and treated as a random variable. At each surface point, e.g., at each mesh vertex and/or triangle, one has a realization of this random variable. To set the notation, let S be a random variable taking values within a subspace $\mathcal{R}_S$ of $\mathbb{R}^m$ and let $f(s|O_t)$ be the pdf of S evaluated on the surface of the object O_t. In the sequel, random variables appear as uppercase letters while their specific instances as lowercase. Suppose furthermore that we have specified a finite set of points within $\mathcal{R}_S$, denoted as $\overline{\mathcal{R}}_S = \{s_n \in \mathcal{R}_S : n = 1, \ldots, N\}$, called *the target set*. The density-based descriptor $\mathbf{f}_{S|O_t}$ for the object O_t (with respect to the feature S) is then simply an N-dimensional vector whose entries consist of the pdf samples at the target set, that is, $\mathbf{f}_{S|O_t} = [f(s_1|O_t), \ldots, f(s_N|O_t)]$.

Density-based shape description consists of three stages. First, in *the design problem*, one chooses good local features that accumulate to global shape descriptors. Good features are computationally feasible and discriminating. Second, one should address *the computational problem*, in search of an efficient computational scheme to estimate $f(s|O_t)$ at designated targets $s \in \overline{\mathcal{R}}_S$. In [1, 2], we have used the kernel approach coupled with a fast algorithm, the fast Gauss Transform (FGT) [24]. Finally, the target selection problem is focused on determining the locations of the points sampled in $\mathcal{R}_S$, i.e., determining the target set $\overline{\mathcal{R}}_S$. The final output is the shape descriptor vector $\mathbf{f}_{S|O_t}$, whose compo-
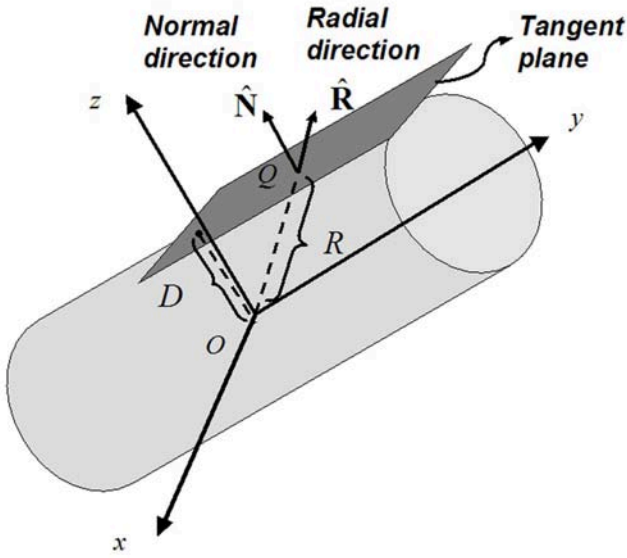
Figure 1. Local features at a surface point Q.

nents are the pdf values evaluated at the target set $\overline{\mathcal{R}}_S$.

3.1 Local Features

In this section, we describe the local geometric features that we use in this work to characterize 3D surfaces (see Figure 1).

Radial distance R measures the distance of a surface point Q to the origin (centroid) and has taken place in many different shape descriptors [16, 17]. Although it is not an effective shape feature all by itself, when coupled to other local surface features, it helps to manifest their distribution at varying radii.

Radial direction $\hat{\mathbf{R}}$ is a unit length vector $(\hat{R}_x, \hat{R}_y, \hat{R}_z)$ collinear with the ray traced from the origin to a surface point. This unit direction vector is obviously scale invariant. When we augment the $\hat{\mathbf{R}}$-vector with the radial distance R, the resulting 4-tuple $(R, \hat{R}_x, \hat{R}_y, \hat{R}_z)$ can be viewed as an alternative to the Cartesian coordinate representation of the surface point. However in this parameterization, the distance and direction information are decoupled. Hence, the range of individual features can be determined independently. In fact, the vector $\hat{\mathbf{R}}$ lies on the unit 2-sphere, and the scalar R lies on the interval $(0, r_{max})$, where r_{max} depends on the size of the surface.

Normal direction $\hat{\mathbf{N}}$ is simply the unit normal vector at a surface point and represented as a 3-tuple $(\hat{N}_x, \hat{N}_y, \hat{N}_z)$. Similar to the radial direction $\hat{\mathbf{R}}$, the normal $\hat{\mathbf{N}}$ is scale invariant.

Radial-normal alignment A is the absolute cosine of the angle between the radial and normal directions and is com-

puted as $A = \left| \langle \hat{\mathbf{R}}, \hat{\mathbf{N}} \rangle \right|$. This feature measures crudely how the surface deviates locally from sphericity. For example, if the local surface approximates a spherical cap, then the radial and normal directions align, and the alignment A approaches unity.

Tangent-plane distance D stands for the absolute value of the distance between the tangent plane at a surface point and the origin. This scalar feature D is related to the radial distance R by $D = R.A$. The joining of D with the normal direction $\hat{\mathbf{N}}$ provides a four-dimensional vector $(D, \hat{N}_x, \hat{N}_y, \hat{N}_z)$ that corresponds to the representation of the local tangent plane as suggested in [25]. As in the radial case, this representation also separates the distance and direction information associated with the tangent plane.

In additition to the radial-normal alignment A, the interaction between the surface normal vector and the radial direction can be quantified by taking the cross product between $\hat{\mathbf{R}}$ and $\hat{\mathbf{N}}$. The *torque* feature $\mathbf{C} = \hat{\mathbf{R}} \times \hat{\mathbf{N}}$ can be considered as a local rotational force when $\hat{\mathbf{R}}$ is viewed as the position of a particle, which is under the influence of an external force $\hat{\mathbf{N}}$.

Shape index SI, first proposed by Koenderink and van Doorn [14], provides a local categorization of the shape into primitive forms such as spherical cap and cup, rut, ridge, trough, or saddle. In the present work, we consider the parameterization proposed in [6] given by

$$SI = \frac{1}{2} - \left(\frac{2}{\pi}\right) arctan\left(\frac{\kappa_1 + \kappa_2}{\kappa_1 - \kappa_2}\right),$$

where κ_1 and κ_2 are the principal curvatures at the surface point. SI is confined within the range $[0, 1]$ and not defined when $\kappa_1 = \kappa_1 = 0$ (planar patch). Since the shape index SI is a function of the principal curvatures, it is considered as a second-order feature. In the present work, we have used Taubin's algorithm [21] to estimate the principal curvatures.

3.2 Kernel Density Estimation

The density-based paradigm relies on the premise that the features of similar shapes induce similar probability distributions. Given a set of observations $\{s^k\}_{k=1}^{K}$ within $\mathcal{R}_S$ for a random variable (scalar or vector) S, the kernel approach for its probability density estimation is formulated as

$$f(s|O_t) = \sum_{k=1}^{K} w_k \, |H|^{-1} \, \mathcal{K}(H^{-1}(s - s^k)),$$

where $\mathcal{K} : \mathbb{R}^m \rightarrow \mathbb{R}$ is a kernel function, H is a $m \times m$ matrix composed of a set of design parameters called bandwidth parameters, and w_k is the importance weight associated with the kth observation s^k [10, 18].

The contribution of each data point (observation) s^k to the density function $f(s|O_t)$ at some target $s \in \overline{\mathcal{R}}_S$ is computed through the kernel function $\mathcal{K}$ scaled by the matrix H

and the weight w_k. Thus KDE involves a data set $\left\{s^k\right\}_{k=1}^{K}$ with their associated importance weights $\{w_k\}_{k=1}^{K}$, the choice of a kernel function $\mathcal{K}$ and the setting of the bandwidth matrix H.

It is known that the particular functional form of the kernel does not significantly affect the accuracy of the estimator [10, 18]. In our scheme, we choose the Gaussian kernel since there exists a fast algorithm, the fast Gauss transform (FGT) [24], to rapidly evaluate large KDE sums in $O(K + N)$ instead of the $O(KN)$-complexity of the direct evaluation.

The setting of the bandwidth matrix H has been shown to be critical for accurate density estimation [10, 18], which in turn affects shape discrimination and retrieval performance [1, 2]. The optimal bandwidth for KDE depends on the unknown density itself [1, 2], making the appropriate choice of the bandwidth parameter a challenging problem. Intuitively speaking, in applications such as object recognition where class information can be acquired during the training phase, the bandwidth parameter should be different for each class. In the absence of class information, the bandwidth parameter can be set at the object-level or at the database-level. In [1, 2], we have experimentally demonstrated that the database level setting yields better discrimination performance by as much as 11%. Accordingly, we set the bandwidth parameter by averaging all object-level bandwidths given by Scott's rule [10, 18]. This averaging process has some intuitive plausibility as it eliminates object-level details and provides us with bandwidth parameters that accord class-level information.

3.3 Marginalization

Marginalization is a tool to selectively remove a subset of features from a multi-dimensional descriptor by integrating out those features. To remove the information brought by a certain component S_j from the pdf of some m-dimensional feature $S = (S_1, \ldots, S_m) \in \mathcal{R}_S$, we use the marginalization equation:

$$f\left(s_1, \ldots, s_{j-1}, s_{j+1}, \ldots, s_m | O_t\right) = \int_{S_j} f\left(s_1, \ldots, s_j, \ldots, s_m\right) ds_j.$$

After marginalization, we obtain the pdf of a reduced feature $S_{j'} = (S_1, \ldots, S_{j-1}, S_{j+1}, \ldots, S_m) \in \mathcal{R}_{S_{j'}}$, hence a new descriptor deprived of any information regarding the component S_j. The removal of a feature component reduces the dimension of the space $\mathcal{R}_S$ by one (observe that $\mathcal{R}_{S_{j'}} \subset \mathcal{R}_S$), and hence a reduction in the descriptor size as explained next. The target set $\overline{\mathcal{R}}_S$ for a multivariate density is chosen as the Cartesian product of individual domains of local feature components. Each marginalization of a feature

component reduces then the descriptor size by a factor equal to the number of discrete values used in the target selection for that component. For example, to obtain the target set for the feature $(R, \hat{R}_x, \hat{R}_y, \hat{R}_z)$, we first sample the scalar radial distance R component in N_R discrete values. For the radial direction $\hat{\mathbf{R}}$, we sample the unit sphere in $N_{\hat{\mathbf{R}}}$ discrete directions. The target set for the whole multivariate feature contains $N_R \times N_{\hat{\mathbf{R}}}$ points, which is also the size of the descriptor. Marginalizing R leads then to a reduction of the descriptor size by a factor N_R. Marginalization aims at simplifying the descriptor without compromising its discrimination capability and is a tool to explore feature components that are more effective for retrieval. Less discriminating features can be identified by marginalization as follows. We first obtain a full density-based descriptor including all the available features in the limit of the computational resources and memory, and then we remove features one at a time. The feature corresponding to the the least performance loss after removal can be considered as the least discriminating one.

3.4 Probability Density Suppression

The second approach we used to reduce the descriptor size is to select a subset of significant target points from the set $\overline{\mathcal{R}}_S$ and eliminate the rest. Let the unconditional pdf $f(s)$ of the feature S be

$$f(s) = \sum_{t=1}^{T} f(s|O_t) Pr\left\{O_t\right\} = \frac{1}{T} \sum_{t=1}^{T} f(s|O_t),$$

where $Pr\left\{O_t\right\}$ is the probability of occurence of the object O_t and T is the total number of objects in the database. Assuming that all objects are equiprobable, i.e., $Pr\left\{O_t\right\} = 1/T$, $f(s)$ is obtained by averaging $f(s|O_t)$'s over all objects $O_t, t = 1, \ldots, T$. The idea of suppression relies on the premise that negligible pdf values would not be effective in discriminating the shapes and can be removed from the descriptor. Thus, for a selected threshold λ, we define a new target set $\overline{\mathcal{R}}_S^{\lambda} = \left\{s_n \in \overline{\mathcal{R}}_S : f(s_n) \geq \lambda\right\}$ and retain only the pdf values at the targets in this new set for the descriptor. The reduced descriptor for some object O_t becomes then $\mathbf{f}_{S|O_t}^{\lambda} = [f(s_n|O_t)]_{s_n \in \overline{\mathcal{R}}_S^{\lambda}}$. Notice that greedy component selection methods are not practical when the descriptor sizes are in the order of thousands. Suppressing small pdf values, albeit not tantamount to component selection, still serves the goal of reducing descriptor sizes. In other words, probability density suppression, although not to the full extent, guides us in choosing more informative feature locations, i.e., targets.

4 Experiments

We have conducted retrieval experiments on two major databases: the Princeton Shape Benchmark (PSB) [19] and the Sculpteur database (SCU) [9]. PSB was split into a training set (907 objects, 90 classes) and a test set (907 objects, 92 classes), while SCU consists of a single set (513 objects, 53 classes). The goals of the experiments were two-fold: (i) to understand the interplay of feature-level and descriptor-level combination alternatives to improve retrieval effectiveness, (ii) to evaluate the efficiency of our descriptor simplification tools, marginalization and suppression.

Prior to descriptor computation, all models have been normalized so that descriptors are translation, rotation, flipping and scale invariant. For translation invariance, the object's center of mass is considered as the origin of the coordinate frame. For rotation and flipping invariance, we applied Vranic's continuous PCA [23]. For isotropic scale invariance, we calculate a scale factor so that the average point-to-origin distance is unity. We have used discounted cumulative gain (DCG) and nearest neighbor (NN) scores to assess the retrieval performance [19]. In all experiments, we have used the L^1-metric to measure the similarity (or the distance) between two descriptors, since it is computationally the cheapest measure with adequate discrimination for density-based descriptors [2].

4.1 Combination at the feature-level

We first explore the impact of feature-level combinations on the retrieval performance. Obviously, the KDE scheme can be applied to any joining of the 13 features described in Section 3.1. The whole space of combinations contains $2^{13} - 1 = 8191$ possibilities. However, estimation accuracy and computational/memory constraints limit the maximum number of features that can be joined. In fact, for dimensions exceeding five, even with a sparse sampling of multi-dimensional feature domains, we obtain descriptor sizes of the order of 10^4, which are impractical. Hence, we limit ourselves to three-, four-, or five-tuple feature combinations in which case the cardinality of the combination space reduces to $C_{13}^3 + C_{13}^4 + C_{13}^5 = 2288$. Even without an exhaustive search, a small subset of judiciously chosen combinations can indicate the potential of feature joining in improving discrimination performance.

In Table 1, we give the results of a small subset of 13 combinations, where the shaded cells in any row show the features taking role in that specific combination. Consider, for example, row **1**: the shaded cells pick the radial distance R and the radial direction $\hat{\mathbf{R}}$ so that this row corresponds to the $(R, \hat{R}_x, \hat{R}_y, \hat{R}_z)$-combination. On the rightmost columns, we provide the size of the multivariate density descriptor, the NN and DCG scores respectively for the three data sets (PSB training, PSB test and SCU). The size of a descriptor equals the number of density evaluation points, i.e., the targets (see Section 3.3). In the experiments, as a general rule, for each scalar feature we have chosen 8 equally spaced points within its domain (9 for the shape index SI, in order to cover the shape primitives such as spherical cap and cup, ridge, trough, saddle, etc., see [6, 13] for details). For unit-norm directional features such as radial $\hat{\mathbf{R}}$ or normal $\hat{\mathbf{C}}$, we have sampled 128 points on the unit sphere (by subdividing two-times the triangles of an octahedron in four triangles). When a directional feature occurs without its third coordinate, as for example $(\hat{R}_x, \hat{R}_y)$, we choose 64 points. An exception to this case is $(R, \hat{R}_x, \hat{R}_y, \hat{N}_x, \hat{N}_y)$ where we have chosen 16 points for both $(\hat{R}_x, \hat{R}_y)$ and $(\hat{N}_x, \hat{N}_y)$ pairs to control the descriptor size. Finally, for the torque feature $\mathbf{C}$, we have sampled 320 points within the unit sphere.

For the three datasets, the $(R, \hat{R}_x, \hat{R}_y, \hat{N}_x, \hat{N}_y)$ (row **4** in Table 1) and $(\hat{R}_z, D, \hat{N}_x, \hat{N}_y)$ (row **10**) configurations, of sizes 2048 and 4096 respectively, are the two descriptors with top performance. The better performance of $(\hat{R}_z, D, \hat{N}_x, \hat{N}_y)$ against other 4-tuple combinations suggests that mixing pieces of radial and normal information results in a more discriminating descriptor than any other using solely the radial information $(R, \hat{R}_x, \hat{R}_y, \hat{R}_z)$ or the tangent plane information $(D, \hat{N}_x, \hat{N}_y, \hat{N}_z)$. We also observe that adding the alignment information A into a configuration significantly increases the discrimination performance, at the cost of increased descriptor size. For instance on PSB training set, the $(R, \hat{R}_x, \hat{R}_y, \hat{R}_z)$-configuration has a DCG performance of 57% (row **1**), while the augmented $(R, \hat{R}_x, \hat{R}_y, \hat{R}_z, A)$ has a DCG of 61.7% (row **2**).

4.2 Combination at the descriptor-level

As pointed out in the previous section, there is a limit for feature-level combinations with a dimension greater than five: (i) the quality of density estimation degrades due the curse of dimensionality [7], and (ii) descriptors become prohibitively large and cannot be computed fast enough for online applications. Descriptor-level combination provides an alternative to extract most of the information brought by different features. However, in contrast to the feature-level combination, it cannot make use of feature correlations in discriminating shapes. Accordingly, it is a "suboptimal" way to fuse shape information. In descriptor-level combination, we concatenate the descriptors computed using various feature combinations as in Section 4.1. We consider pair-wise and triple concatenations of the 13 descriptors (each indicated by its boldface number) presented in Section 4.1. The performances of various descriptor combinations are visualized in Figure 2 (pair-wise) and Figure 3
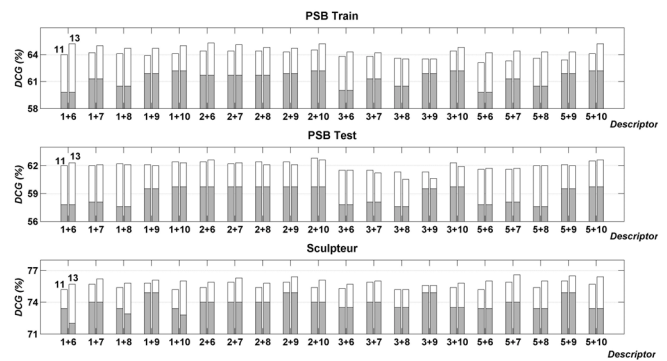
Table 1. Retrieval performance for feature-level combinations: gray cells denote the selected features for the 13 multivariate density-based descriptors.

| | | Features | | | | | | | | | | | | | | Databases | | | | | |
| | | | | | | | | | | | | | | | | PSB Train. | | PSB Test | | Sculpteur | |
Multivariate Density Descriptor		R	$\hat{R}_x$	$\hat{R}_y$	$\hat{R}_z$	D	$\hat{N}_x$	$\hat{N}_y$	$\hat{N}_z$	C_x	C_y	C_z	A	SI	Size	NN (%)	DCG (%)	NN (%)	DCG (%)	NN (%)	DCG (%)
	1	■	■	■	■										1024	57.8	57.0	54.6	54.9	75	71.3
	2	■	■	■	■								■		8192	63.9	61.7	62.1	59.7	76.6	74.0
	3	■	■	■	■								■		4096	60.9	60.0	58.7	57.1	76.2	73.5
	4	■	■	■	■		■	■							2048	62.4	61.0	61.6	60.1	78.8	75.4
	5	■	■	■	■				■	■					4096	58.4	57.6	57.9	57.2	77.6	73.4
	6					■	■	■	■						1024	59.9	59.8	59.1	57.8	74.5	72.0
	7					■	■	■	■				■		8192	63.5	61.3	59.0	58.1	77.8	74.0
	8					■	■	■	■				■		4096	61.0	60.5	55.0	57.6	74.3	72.9
	9	■					■	■	■				■		4096	60.8	61.9	58.7	59.5	75.8	74.9
	10				■		■	■	■						4096	64.1	62.2	60.1	59.7	76.4	72.8
	11	■								■	■	■			2560	51.0	55.6	51.0	54.8	73.5	73.4
	12	■								■	■	■			4096	61.4	60.5	56.8	57.8	78.2	74.7
	13	■											■	■	576	42.9	52.6	40.5	49.1	69.8	71.0
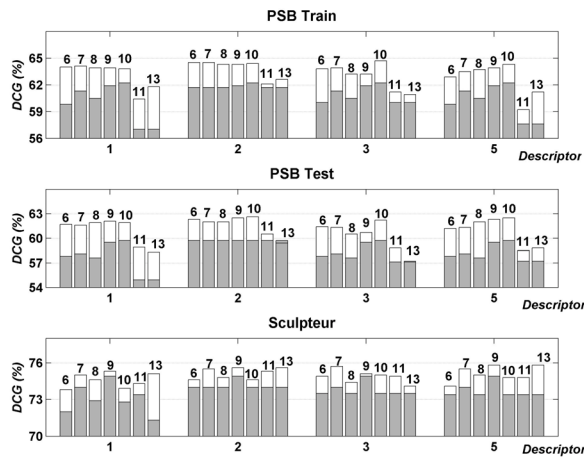
Figure 2. DCG improvements with pair-wise descriptor combinations. Gray portion of the bar: maximum DCG before combination; white portion of the bar: DCG increase after combination.

Figure 3. DCG improvements with triple descriptor combinations. Gray portion of the bar: maximum DCG before combination; white portion of the bar: DCG increase after combination.

(triple). In Figure 2, the horizontal axis stands for descriptors **1**, **2**, **3**, and **5** of Table 1. Each bar series illustrates DCG performance improvements after pair-wise combination with descriptors **6**, **7**, **8**, **9**, **10**, **11**, and **13** (displayed on the top of each bar) one at a time. For instance, at descriptor **1**, the bar with the index **6** on the top corresponds to the pair-wise combination **1+6**, i.e., the concatenation of the pdf of $(R, \hat{R}_x, \hat{R}_y, \hat{R}_z)$ with that of $(D, \hat{N}_x, \hat{N}_y, \hat{N}_z)$.

The gray bars indicate the maximum of DCG performances of the individual descriptors involved in that specific combination, e.g., $max(DCG_1, DCG_6)$. The white portion of the bar on top of the gray one shows the DCG improvement due to the combination. Figure 3 illustrates these gains for triple combinations, that is, descriptor vectors resulting from three concatenated descriptors. Accordingly, in Figure 3, the horizontal axis has labels of pair-wise combinations, such as **1+6**, **1+7**,..., **5+9**, **5+10**. The darker portion of the bar again indicates the maximum individual performances of descriptors before combination, and the light portion indicates the corresponding DCG improvement after

triple combination. In these experiments, the third descriptor is always either the (R, C_x, C_y, C_z)-descriptor (**11**) or (R, A, SI)-descriptor (**13**), as they bring in different shape information than the pair-wise combinations displayed at the horizontal axis of Figure 3.

The combinations have been intuitively selected so that their corresponding feature sets are as disjoint as possible. For instance, we did not consider concatenating the pdf of $(R, \hat{R}_x, \hat{R}_y, \hat{R}_z)$ (descriptor **1**) with that of $(R, \hat{R}_x, \hat{R}_y, \hat{R}_z, A)$ (descriptor **2**) as the latter already contains the information brought by the former. As expected, descriptor combinations become effective only when the involved features bring in a different kind of shape information. A case in point involves $(R, \hat{R}_x, \hat{R}_y, \hat{R}_z)$ (descriptor **1** in Table 1) and $(D, \hat{N}_x, \hat{N}_y, \hat{N}_z)$ (descriptor **6** in Table 1). For PSB training set, individual DCGs of these descriptors are 57% and 59.8% respectively (see Table 1). After the combination, the DCG performance boosts to 64% achieving a DCG improvement of 7% for PSB training set (6.7% for PSB test set and 2.5% for Sculpteur database), as shown in Figure 2. Furthermore, in this case, the descriptor size is 2048 (2×1024) and quite reasonable in comparison to other options. When we consider the triple combinations, in the case where (R, A, SI)-descriptor (descriptor **13** in Table 1) is concatenated to the previous descriptor combination **1+6**, DCG score has a further but smaller jump to 65.2% for PSB training set, to 62.3% for PSB test set, and to 75.7% for Sculpteur (Figure 3). Presently, this combination constitutes the most interesting option as it has the smallest size among other triple combinations tested ($1024 + 1024 + 576 = 2624$ in total).

4.3 Reducing the size of descriptors

In this section, we explore descriptor reduction tools that remove insignificant features and/or target points with controlled sacrifice of retrieval effectiveness. These tools concentrate on features and/or their targets with substantial mass concentration. The previous two sections have indicated that the descriptor-level combination (**1+6+13**) of $(R, \hat{R}_x, \hat{R}_y, \hat{R}_z)$, $(D, \hat{N}_x, \hat{N}_y, \hat{N}_z)$ and (R, A, SI)-densities yield the best performance and is the most economical among triple combinations tested (DCG = 62.3% on PSB test set and DCG = 75.7% on Sculpteur; see Figure 3). Accordingly, it makes sense to explore this particular descriptor for dimensionality reduction.

Reduction by marginalization: We marginalize with respect to one of the directional components $\hat{R}_x, \hat{R}_y, \hat{R}_z$ (or $\hat{N}_x, \hat{N}_y, \hat{N}_z$), since the radial (normal) direction is a unit-norm vector as defined in Section 3.1, hence potentially redundant. We admit that it is also possible to reach the same goal by directly computing the descriptors of reduced size via KDE, say the pdf of $(\hat{R}, \hat{R}_x, \hat{R}_y)$. This exercise, how-ever, must be repeated for every conceivable pruned configuration of $(R, \hat{R}_x, \hat{R}_y, \hat{R}_z)$. Instead, once the density of $(R, \hat{R}_x, \hat{R}_y, \hat{R}_z)$ is at our disposal, the marginalization operator allows us to obtain alternate reduced descriptors very rapidly. We find that the marginalization tool brings to light the flexibility of the density-based framework.

Table 2 displays individual NN and DCG performances of $(R, \hat{R}_x, \hat{R}_y, \hat{R}_z)$ (descriptor **1**) and $(D, \hat{N}_x, \hat{N}_y, \hat{N}_z)$ (descriptor **6**) and the performances obtained after removing one of the $\hat{R}_x, \hat{R}_y, \hat{R}_z$ (or $\hat{N}_x, \hat{N}_y, \hat{N}_z$ respectively). In Table 2, $1_{\not x}$, $1_{\not y}$, $1_{\not z}$, $6_{\not x}$, $6_{\not y}$ and $6_{\not z}$ denote descriptors with the removed feature being subscripted and crossed out. Notice that no performance loss is incurred in solo mode or in combined mode except for a maximum of 3% DCG fluctuation. Interestingly enough, for some cases there is even an improvement of similar magnitude. However such variations are often due to permutations of retrieved models at the end of the sorted list and hence should not be considered as significant. The net effect of marginalization has been to halve the descriptor size without loss of DCG performance. In Table 2, we also see that the DCG of the triple combination obtained by adding (R, A, SI) (descriptor **13**) persists at its original level for all the data sets while the descriptor size falls from 2624 to 1600.

Reduction by pdf suppression: We can achieve further dimensionality reduction without compromising performance by invoking the probability density suppression tool. In Figure 4, we display the DCG profile for $(R, \hat{R}_x, \hat{R}_y)$, $(D, \hat{N}_x, \hat{N}_z)$, and (R, A, SI) as a function of the suppression threshold λ on PSB training set. From this figure, it can be seen than $(D, \hat{N}_x, \hat{N}_z)$ is more susceptible to suppression than $(R, \hat{R}_x, \hat{R}_y)$ and (R, A, SI) in terms of DCG loss. Nevertheless, a fixed threshold level of 0.03 for suppressing targets for $(R, \hat{R}_x, \hat{R}_y)$, $(D, \hat{N}_x, \hat{N}_z)$, and (R, A, SI) on PSB training set incurs no DCG loss and reduces the combined descriptor size to 1150 from 1600 (DCG of the combination being 65.5% for PSB training set, 62.5% for PSB test set and 75.8% for Sculpteur, see Table 3 for other threshold values from 0.01 to 0.05).

Reduction by subspace methods: A more traditional approach to reduce dimensionality is to project the descriptor vector onto a linear subspace using standard methods such as principal components analysis (PCA) or independent component analysis (ICA) [7]. The PCA of our $(R, \hat{R}_x, \hat{R}_y, \hat{R}_z)$-descriptor on PSB training set reveals that the total variance in the first 270 components reaches 99%, suggesting that significant dimensionality reduction can be achieved. We have experimented by changing the projection dimension from 200 down to 10 (with increments of 10) and performed retrieval experiments using L^1 and L^2-distances. We define the DCG efficiency as the ratio of the DCG after PCA to the DCG of the full descriptor. In Figure 5, DCG efficiency profiles for both L^1 and L^2-distances
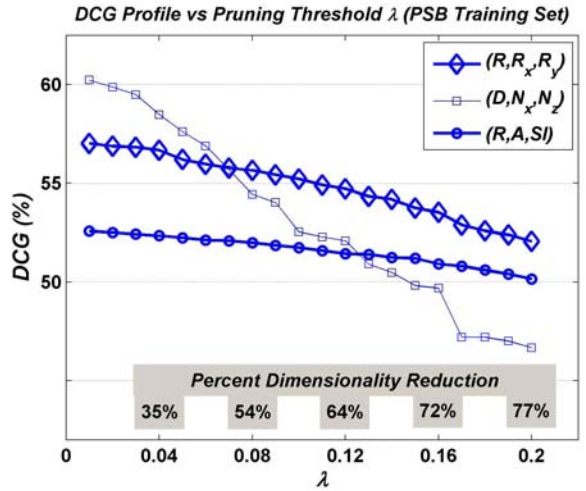
Table 2. Retrieval Performance After Marginalization (crossed subscript of the descriptor denotes the feature marginalized out).

Desc.	Size	PSB Test		SCU	
		NN	DCG	NN	DCG
1	1024	54.6	54.9	75.0	71.3
$1_{\cancel{x}}$	512	53.5	54.5	74.7	70.5
$1_{\cancel{y}}$	512	52.7	54.3	76.0	71.4
$1_{\cancel{z}}$	512	53.7	54.2	73.9	70.8
6	1024	59.1	57.8	74.5	72.0
$6_{\cancel{x}}$	512	55.7	57.3	71.7	69.6
$6_{\cancel{y}}$	512	57.0	58.0	74.3	72.7
$6_{\cancel{z}}$	512	56.7	57.7	73.1	72.0
$1+6$	2048	64.2	61.7	77.8	73.8
$1_{\cancel{x}}+6_{\cancel{y}}$	1024	62.7	62.4	78.0	74.3
$1+6+13$	2624	65.3	62.3	78.2	75.7
$1_{\cancel{x}}+6_{\cancel{y}}+13_{\cancel{x}}$	1600	64.8	62.6	76.8	75.8

Figure 4. DCG profile as a function of the suppression threshold parameter λ on PSB training set.

Table 3. Retrieval Performance After Marginalization and pdf Suppression for Combined Descriptor 1+6+13

λ	Size	PSB Test		SCU	
		NN	DCG	NN	DCG
0.00	1600	64.8	62.6	76.8	75.8
0.01	1343	64.6	62.6	77.2	75.9
0.02	1236	64.7	62.6	77.2	75.9
0.03	1150	64.5	62.5	77.2	75.8
0.04	1041	65.2	62.3	76.8	75.7
0.05	925	65.0	62.0	76.4	75.5

are shown. Note that L^1 and L^2 performances of the full descriptor (size = 1024) are 57% and 54.8% respectively in terms of DCG. In Figure 5, we see that, using the L^1-distance, we can never achieve 100% DCG efficiency although the dimensionality reduction at the maximum possible DCG efficiency ($\approx 96\%$) is significant (reduced descriptor size is 50). On the other hand in the same figure, we observe that, when the L^2-metric is used, we can improve the DCG with respect to the full descriptor even after dimensionality reduction (efficiency above 100%). This should not be a big surprise as PCA is basically an L^2 method. Unfortunately, the L^2-metric is not our preferred one as it is always inferior to L^1 in terms of DCG performance [2]. Nevertheless, reduction rates achieved by PCA are quite impressive. In applications where a DCG loss at the order of

2-5% is affordable, PCA may become interesting. We have also experimented with the ICA method and observed that no significant gain can be obtained in comparison to PCA.

4.4 Comparison with other descriptors

In Figure 6, we show the DCG performance comparison of 3D shape descriptors, on PSB test set. Priority-driven search (PDS) constitutes an exception in this comparison, as it does not exactly rely on descriptor paradigm but on matching by feature correspondences. Quoted DCG values are either taken from their original sources or from the works in [8, 19], whereas CAH and 3DHT methods are implemented and tested by us. The performance of a representative density-based descriptor, namely the triple combination of $(R, \hat{R}_x, \hat{R}_y)$, $(D, \hat{N}_x, \hat{N}_z)$, and (R, A, SI)-densities after probability suppression are also included in Figure 6.

We see that the proposed density-based descriptor gives the best performance (DCG = 62.5%) among all the purely 3D methods [3, 11, 12, 15, 16, 17, 22, 25]. The 3D method closest in performance to our descriptor is REXT with DCG = 60.1%. The two methods with higher performance are light field descriptor (LFD, DCG = 63.6% [5]) and depth buffer images (DBI, DCG = 66.3% [4]), both of which rely on 2D features extracted from 2D projections of 3D objects. The top discriminating (DCG = 75.9%) but computationally heavy PDS is outside the vector-based paradigm (4-5 minutes per object to process the database, 100 KB memory per object and 2.4 secs to match per query as the authors report [8]). On the other hand, the computational
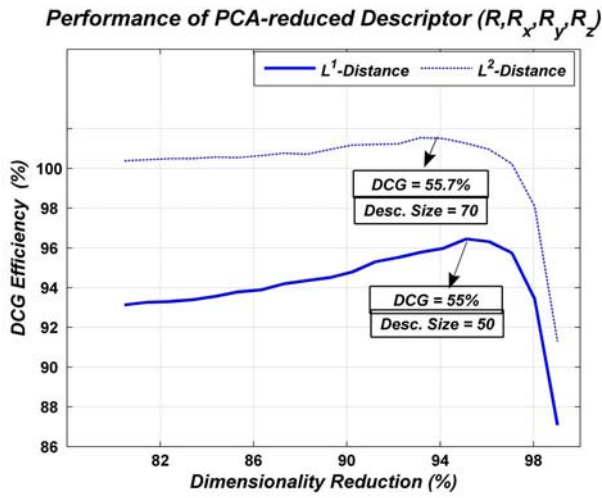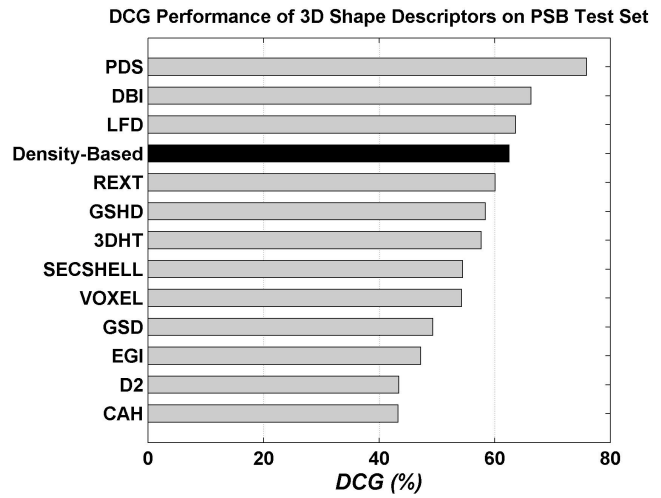
Figure 5. DCG efficiency profile after PCA-based dimensionality reduction. Sample point for the L^1 and L^2 metric shown on the curves illustrate the economies achieved with PCA with little change in DCG performance

Figure 6. DCGs of a selected group of state-of-the-art descriptors on PSB test set.

and memory advantages of the triple descriptor combination of the $(R, \hat{R}_x, \hat{R}_y)$, $(D, \hat{N}_x, \hat{N}_z)$, and (R, A, SI)-densities against its competitors is compelling. More importantly, on a standard laptop PC (1.86 GHz CPU and 1 GB RAM), a density-based descriptor of this size can be calculated within half a second using the FGT algorithm [24]. For instance, the extraction time for LFD, which is DCG-wise 1% superior to our descriptor, is 3.25 secs per object on a comparable hardware configuration [19]. Apart from all these performance figures, we think that the density-based framework provides a sound formalism to process 3D local surface features and paves the way towards the probabilistic treatment of 3D surface information.

5 Conclusion

In this study, we have experimentally demonstrated the benefits of feature-level and descriptor-level fusion for the density-based 3D object retrieval. The descriptors presented are also attractive because they are computationally much less costly as compared to alternate descriptors in Figure 6 thanks to the fast Gauss transform [24]. In addition, these descriptors based on multivariate density functions can be simplified significantly without compromising performance using marginalization and probability density suppression.

Feature fusion followed by KDE results in the multivariate density descriptors, which proves to be more effective than histogram-based methods [3, 11, 15, 16, 17, 25]. Compared to the best case of scalar features (DCG≈ 40%), we can improve the DCG performance by as much as 50%. Furthermore, descriptor fusion augments the DCG performance from the best case by as much as 4.5% DCG points on PSB test set, yielding the most effective description scheme among 3D methods [3, 11, 12, 15, 16, 17, 22, 25]. Fusion is effective especially when features and/or descriptors involved reflect different aspects of shape information. Less effective feature components can be selectively removed from the descriptors by marginalization. In the best case, single component marginalization reduce the descriptor size as much as 50% without any performance loss. After marginalization, descriptors can be further simplified by pdf suppression by as much as 30% without losing any discrimination ability; for example, the best performing descriptor in Section 4.3 after marginalization has been reduced from the original size of 1600 to 1150. The pdf suppression method, when used on its own, can reduce the descriptor size by as much as 70% with a 5% loss in DCG performance while the PCA method can provide 95% reduction in descriptor size.

At this stage, this work constitutes an extensive investigation on the discrimination capabilities of local surface features for the 3D retrieval problem. Our future research will be concentrated on exploring the feature and descriptor level combination space using DCG-driven sequential forward/backward floating search methods and on the statistical learning of dissimilarity measures between descriptors for retrieval.

References

[1] C. B. Akgül, B. Sankur, Y. Yemez, and F. Schmitt. Density-based 3D shape descriptors. *EURASIP Journal on Advances in Signal Processing*, 2007:Article ID 32503, 16 pages, 2007. doi:10.1155/2007/32503.

[2] C. B. Akgül, B. Sankur, Y. Yemez, and F. Schmitt. Improving efficiency of density-based shape descriptors for 3D object retrieval. In *Computer Vision/Computer Graphics Collaboration Techniques and Applications (MIRAGE '07)*, Paris, France, March 2007.

[3] M. Ankerst, G. Kastenmüller, H.-P. Kriegel, and T. Seidl. 3D shape histograms for similarity search and classification in spatial databases. In *Proc. of the 6th International Symposium on Advances in Spatial Databases (SSD '99)*, pages 207–226. Springer-Verlag, 1999.

[4] B. Bustos, D. A. Keim, D. Saupe, T. Schreck, and D. V. Vranic. Feature-based similarity search in 3D object databases. *ACM Comput. Surv.*, 37(4):345–387, 2005.

[5] D.-Y. Chen, X.-P. Tian, Y.-T. Shen, and M. Ouhyoung. On visual similarity based 3D model retrieval. In *Computer Graphics Forum*, volume 22, pages 223–232, September 2003.

[6] C. Dorai and A. K. Jain. COSMOS - A representation scheme for 3D free-form objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(10):1115–1130, 1997.

[7] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley-Interscience, 2000.

[8] T. Funkhouser and P. Shilane. Partial matching of 3D shapes with priority-driven search. In *Symposium on Geometry Processing*, June 2006.

[9] S. Goodall, P. H. Lewis, K. Martinez, P. A. S. Sinclair, F. Giorgini, M. Addis, M. J. Boniface, C. Lahanier, and J. Stevenson. SCULPTEUR: Multimedia retrieval for museums. In *Proc. of the Image and Video Retrieval: Third International Conference (CIVR '04)*, pages 638–646, 2004.

[10] W. Härdle, M. Müller, S. Sperlich, and A. Werwatz. *Nonparametric and Semiparametric Models*. Springer Series in Statistics. Springer, 2004.

[11] B. K. P. Horn. Extended Gaussian images. *Proc. of the IEEE*, 72:1671–1686, 1984.

[12] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz. Rotation invariant spherical harmonic representation of 3D shape descriptors. In *Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry Processing (SGP '03)*, pages 156–164, Aire-la-Ville, Switzerland, 2003. Eurographics Association.

[13] J. J. Koenderink. *Solid Shape*. The MIT Press, Cambridge, Massachusetts, 1990.

[14] J. J. Koenderink and A. J. van Doorn. Surface shape and curvature scales. *Image Vision Comput.*, 10(8):557–565, 1992.

[15] Y. Liu, H. Zha, and H. Qin. The generalized shape distributions for shape matching and analysis. In *Proceedings of the IEEE International Conference on Shape Modeling and Applications (SMI 2006)*, June 2006.

[16] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin. Shape distributions. *ACM Trans. Graph.*, 21(4):807–832, 2002.

[17] E. Paquet and M. Rioux. Nefertiti: a query by content software for three-dimensional models databases management. In *Proc. of the International Conference on Recent Advances in 3-D Digital Imaging and Modeling (NRC '97)*, page 345, Washington, DC, USA, 1997. IEEE Computer Society.

[18] D. W. Scott. *Multivariate Density Estimation. Theory, Practice and Visualization*. Wiley, New York, 1992.

[19] P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser. The Princeton shape benchmark. In *Proc. of the Shape Modeling International 2004 (SMI '04)*, pages 167–178, Genoa, Italy, 2004.

[20] J. W. H. Tangelder and R. C. Veltkamp. A survey of content based 3D shape retrieval methods. In *Proc. of the Shape Modeling International 2004 (SMI '04)*, pages 145–156, Genoa, Italy, 2004.

[21] G. Taubin. Estimating the tensor of curvature of a surface from a polyhedral approximation. In *ICCV '95: Proceedings of the Fifth International Conference on Computer Vision*, page 902, Washington, DC, USA, 1995. IEEE Computer Society.

[22] 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 2003)*, pages 757–760, Barcelona, Spain, Spetember 2003.

[23] D. V. Vranić. *3D Model Retrieval*. PhD thesis, University of Leipzig, 2004.

[24] C. Yang, R. Duraiswami, N. A. Gumerov, and L. Davis. Improved fast Gauss transform and efficient kernel density estimation. *ICCV*, 1:464, 2003.

[25] T. Zaharia and F. Prêteux. Shape-based retrieval of 3D mesh models. In *Proc. of the IEEE International Conference on Multimedia and Expo (ICME'2002)*, Lausanne, Switzerland, August 2002.