3D Model Retrieval Using Probability Density-Based Shape Descriptors

Ceyhun Burak Akgül, *Student Member*, *IEEE*, Bülent Sankur, *Senior Member*, *IEEE*, Yücel Yemez, *Member*, *IEEE*, and Francis Schmitt

Abstract—We address content-based retrieval of complete 3D object models by a probabilistic generative description of local shape properties. The proposed shape description framework characterizes a 3D object with sampled multivariate probability density functions of its local surface features. This density-based descriptor can be efficiently computed via kernel density estimation (KDE) coupled with fast Gauss transform. The nonparametric KDE technique allows reliable characterization of a diverse set of shapes and yields descriptors which remain relatively insensitive to small shape perturbations and mesh resolution. Density-based characterization also induces a permutation property which can be used to guarantee invariance at the shape matching stage. As proven by extensive retrieval experiments on several 3D databases, our framework provides state-of-the-art discrimination over a broad and heterogeneous set of shape categories.

Index Terms—Shape matching, retrieval, surface representations, nonparametric statistics, geometric transformations, invariance, feature evaluation and selection, performance evaluation.

1 Introduction

Fast and accurate scanning technology equipped with shape modeling and rendering tools has enabled the means of acquiring, designing, and manipulating complete 3D models of real-world objects. Digital 3D models as a new modality of visual information find applications in several domains such as computer-aided design [1], cultural heritage archival [2], molecular modeling [3], and video games industry [4], [5]. With growing interest in 3D models, their effective retrieval from large databases is acquiring economic utility [4], [6], [7]. Text-based systems, much like in all other media applications, would remain severely limited in describing and retrieving 3D models [7]. Content-based systems, on the other hand, offer an effective and scalable complementary solution to the 3D retrieval problem.

We address content-based retrieval of complete 3D object models by a probabilistic generative description of their local shape. We call the proposed method as the density-based framework (DBF) in that it describes 3D objects with multivariate probability density functions (pdfs) of chosen shape features. Our previous study [8] has shown that such an approach has a promising retrieval potential. In this paper, we analyze DBF in greater detail and provide extensive retrieval experiments to demonstrate that it can satisfactorily handle large collections of heterogeneous shape categories. In particular, we show that DBF is relatively insensitive to small shape perturbations and mesh resolution, that it is computationally efficient, and that it enjoys a permutation property which guarantees invariance to a certain class of 3D transformations at the shape matching stage. As a consequence of these contributions, DBF qualifies as one of the best 3D shape descriptors, as established by retrieval experiments on several databases.

Our starting point is that, as similar shapes induce similar feature distributions, two shapes can be compared by the distance between their feature pdfs. Histogram-based 3D shape descriptors [9], [10], [11], [12], [13], [14], [15] (see Section 2) have relied on this intuitively appealing idea but failed to provide fine grain discrimination required by the 3D retrieval task [7]. Compared to its histogram-based ancestors, DBF is original in two aspects: 1) It employs richer sets of *multivariate* shape features and 2) it adopts the *kernel* strategy to estimate the distribution [16]. As a further contribution, we experimentally show that these two aspects overcome the performance limitation of early histogram-based 3D shape descriptors.

After transforming a given 3D model into a canonical coordinate frame and scale, our scheme first characterizes its surface locally using simple and direct features, without resorting to computationally intensive methods such as shape contexts [17] or spin images [18]. Our features are, in fact, as simple as distance-to-origin, radial, and normal directions, and principal curvatures (Sections 3.1 and 3.2). Without sacrificing computational simplicity, we construct more informative local characterizations by joining these simple features into multivariate ones. In a previous work

- *C.B. Akgül is with the Video Processing and Analysis Group, Philips Research Europe, High Tech Campus 36 (WOp122 O-1), 5656AE Eindhoven, The Netherlands. E-mail: ceyhun.akgul@philips.com.*
- *B. Sankur is with the Department of Electrical and Electronic Engineering, Boğaziçi University, Bebek 80815, Istanbul, Turkey. E-mail: bulent.sankur@boun.edu.tr.*
- *Y. Yemez is with the Department of Computer Engineering, Koç University, Rumeli Feneri Yolu, 34450 Sariyer, Istanbul, Turkey. E-mail: yyemez@ku.edu.tr.*
- *F. Schmitt was with the Image and Signal processing Department, Télécom ParisTech (Ecole Nationale Supérieure des Télécommunications before his death.*

Manuscript received 2 June 2008; revised 12 Oct. 2008; accepted 8 Jan. 2009; published online 16 Jan. 2009.
Recommended for acceptance by S. Belongie.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2008-06-0323.
Digital Object Identifier no. 10.1109/TPAMI.2009.25.

[19], we have demonstrated that, for 3D retrieval, pdf-based descriptors induced by such multivariate feature combinations are more effective than combinations of scalar feature pdfs.

Once the surface information is collected, we proceed to estimate the feature pdfs by *kernel density estimation* (KDE) [16] (Section 3.4). The samples of the pdf at given target feature points constitute our 3D shape descriptor. KDE is advantageous in our context in more than one way. First, its nonparametric nature provides us with enough flexibility to model feature distributions for a broad and diverse set of 3D objects. Second, in contrast to the histogram estimator, its smoothing parameter can be adjusted to make the descriptors relatively insensitive to small shape variations and to imperfections in object pose and scale normalization (Section 4.1). Third, descriptors can still be computed very efficiently when KDE is coupled with the fast Gauss transform (FGT) [20], [21] (Section 4.3). Note also that, in this generative distribution-based approach, the descriptor stands for the conditional density of local features for a given shape. Consequently, the averaged pdf over a set of 3D shapes belonging to the same category is semantically relevant and can serve as a category-level prior for general object recognition and classification. A further advantage of pdf-based description is that we can guarantee invariance against a class of object transformations at the shape matching stage. There are, indeed, methods in the literature, such as [22], that use invariant shape matching. However, to achieve invariance, such methods have to recompute the descriptor from scratch for every possible transformation. Obviously, they are not computationally efficient, compromising their use for practical applications. One of the major novelties of the present paper with respect to our previous work [8] is to rigorously show the permutation property of the density-based framework (Section 4.2). This enables, via a simple permutation, almost instantaneous descriptor computation for transformed versions of 3D objects.

We demonstrate the retrieval effectiveness of DBF on four different 3D model databases with varying surface mesh quality, semantic content, and classification granularity. The most notable of these is the Princeton Shape Benchmark (PSB) [23], which has become a standard test environment for 3D shape descriptors since its release in 2004. On PSB, our framework is on a par with the best performing descriptors reported so far. Furthermore, although its closest competitor DSR (a combination of the depth buffer, silhouette, and radialized extent function descriptors, see Section 2 and [24], [25]) is also highly discriminative, we have also observed that DBF and DSR methods are somewhat complementary so that it is possible to achieve even higher retrieval performance with their combination.

The paper is structured as follows: In the next section, we present an overview of previously proposed shape descriptors for 3D retrieval. In Section 3, we describe the steps of DBF in detail. In Section 4, we analyze its properties such as insensitivity to small shape variations and mesh resolution, invariance, and computational efficiency. In Section 5, we undertake an exhaustive campaign of retrieval experiments and illustrate the effectiveness of our methods on several 3D model databases. In Section 6, we conclude and discuss further research directions.

2 Previous Work

Three-dimensional model retrieval hinges on shape matching, that is, determining the extent to which two shapes resemble each other [7]. There are two main approaches to this problem: matching by feature correspondences and matching by global descriptors. The strategy in the former approach is to compute multiple local features for every object and then to compute a distance measure between pairs of objects for an optimal set of feature correspondences and an optimal relative transformation [26]. The global descriptor-based approach, on the other hand, reduces intrinsic shape characteristics to vectors or graph-like data structures, called *shape descriptors*, and then evaluates the distance between the descriptor pairs as a measure of similarity. The difficulty of finding correspondences is a well-known computational problem in computer vision and shape analysis [17]. Global descriptors try to solve the correspondence problem by "registering" the shape information on a common grid. Table 1 provides a taxonomy of 3D shape descriptors with emphasis on "registration" methods as we describe in the present section. References [4], [6], [7] provide more comprehensive reviews in this rapidly evolving field.

A number of 3D shape descriptors can be classified under the heading of *histogram-based* methods [9], [10], [11], [12], [13], [14], [15]. We use the term "histogram" as an accumulator that collects numerical values of certain attributes of the 3D object. In this sense, not all the methods in this category [13], [14], [15] are true histograms in the rigorous statistical sense of the term, but they all share the methodology of accumulating a geometric feature in bins defined over the feature space. These methods bypass the correspondence problem by discarding all the spatial information. The price paid for this solution is their lack of fine grain discrimination required for the retrieval task [7] (3D Hough transform method [15] can be considered as an exception to this, see Section 5.5).

Transform-based methods [24], [27], [28], [29], [30], [31], [32], [33], [34] implicitly register the surface points onto a 3D voxel or spherical grid by means of a scalar-valued function (e.g., a binary function testing the presence of a surface point on a grid point or the signed distance function of the surface), which is then processed by transform tools such as 3D Fourier [27], angular radial transform [28], 3D Radon [29], spherical trace transform [30], spherical harmonics [24], [31], [32], [33], or wavelets [34]. A significant advantage of using transform machinery is descriptor compaction achieved by keeping first few transform coefficients in the descriptor vector. Furthermore, pose invariance can be obtained by discarding the "phase" of the transform coefficients at the expense of some shape information, e.g., as in RISH [31] (see Table 1 for the acronym).

Two-dimensional view-based methods [22], [24] consider the 3D shape as a collection of 2D projections taken from canonical viewpoints. Each projection is then described by standard 2D image descriptors like Fourier descriptors [24] or Zernike moments [22]. These methods work surprisingly

TABLE 1
Taxonomy of 3D Shape Descriptors

Category	Comments	Examples
Histogram-based	▪ Accumulators of local or global features ▪ Easy to implement ▪ Not sufficiently discriminative (except 3DHT)	▪ Shape distributions (D2) [9] ▪ Generalized Shape Distr. [GSD] [10] ▪ Cord and Angle Hist. (CAH) [11] ▪ Shape Histograms (SHIST) [12] ▪ Extended Gaussian Image (EGI) [13, 14] ▪ 3D Hough Transform (3DHT) [15]
Transform-based	▪ Uses signal processing transforms (Fourier, Spherical harmonics,…) ▪ Descriptor compaction ▪ Pose invariance can be obtained at the expense of shape information (e.g., RISH)	▪ 3D Fourier [27] ▪ Angular Radial Transform [28] ▪ 3D Radon [29] ▪ Spherical Trace Transform [30] ▪ Radialized Extent Function (REXT) [32] ▪ Concrete Radialized Spherical Projection (CRSP) [33] ▪ Rotation-Invariant Spherical Harmonics (RISH) [31] ▪ Spherical Wavelet Descriptor (SWD) [34]
2D view-based	▪ 3D = collection of 2D views ▪ Highly discriminative	▪ Depth-Buffer Images (DBI) [24] ▪ Light Field Descriptor (LFD) [22] ▪ Silhouette Descriptor (SIL) [24]
Graph-based	▪ Can encode topology ▪ Hard to obtain ▪ Requires graph matching	▪ Reeb Graphs [35, 36] ▪ Skeletal Graphs [37]

well despite their intuitive disadvantage as they discard valuable 3D information. A possible explanation for their good performance is that, as the 3D models are completely given, projections can be produced in a controlled manner so that nuisance effects of occlusion (except self-occlusions of course), clutter, or affine deformations are avoided. These methods can also be beneficial for 2D sketch-based queries.

In [24], a hybrid descriptor, which is a combination of two 2D view-based methods, DBI and SIL, and a transform-based method REXT, is proposed (see Table 1 for the acronyms). This descriptor, denoted as DSR, is proven to be very effective on PSB [23] and on the Konstanz database [4].

Graph-based descriptors [35], [36], [37] are fundamentally different from other vector-based descriptors. They are more elaborate and complex, in general harder to obtain, but they have the potential of encoding geometrical and topological shape properties in a more faithful and intuitive manner. However, they do not generalize easily to all 3D shape representation formats and they require dedicated matching schemes. In fact, from an algorithmic point of view, graph-based methods do not completely obviate the correspondence issue. They just alleviate it by reducing the problem of matching two feature sets to that of matching graph nodes, which, however, still remains a formidable task for general-purpose retrieval applications. We note that, using tools from spectral graph theory, some part of the information contained in a graph can be encoded in the form of vector-based numerical descriptions.

We now proceed with the exposition of our density-based framework, which can be viewed as a formal generalization of histogram-based 3D shape descriptors.

3 Density-Based Framework

Density-based shape description is a generative model, aiming to represent geometrical shape properties contained within a class of 3D objects as a probability distribution. This generative model relies on the idea that, associated with each shape concept, there is an underlying random process, which induces a probability law on some local surface feature of choice. We assume that this probability law admits a pdf, which encodes intrinsic shape properties to the extent achieved by the chosen feature. The similarity between two shapes can thus be quantified by measuring the variation between their associated feature pdfs.

We define the shape descriptor of a given 3D object as the sampled pdf of some local geometric features computed over its surface. Each of these features is treated as a random variable with a realization (or observation) at every point of the surface. To set the notation, let S be a random feature vector defined on the surface of a 3D object O and taking values within a subspace $\mathcal{R}_S$ of $\mathbb{R}^m$, where m is the number of components in the vector S. Let $f_{S|O} \triangleq f_S(\cdot|O)$ be the pdf of S for the object O. This pdf can be estimated using the set of *feature observations*, called *the source set*, $\{s_k \in \mathcal{R}_S\}_{k=1}^{K}$ computed on the object's surface given in terms of a triangular mesh. Suppose furthermore that we have specified a finite set of N *pdf evaluation points* within $\mathcal{R}_S$, denoted as $\overline{\mathcal{R}}_S = \{t_n \in \mathcal{R}_S\}_{n=1}^{N}$, called *the target set*. The density-based descriptor $\mathbf{f}_{S|O}$ for the object O (w.r.t. the feature S) is then simply an N-dimensional vector whose entries consist of the pdf samples at the target set, that is, $\mathbf{f}_{S|O} = [f_S(t_1|O), \ldots, f_S(t_N|O)]$. Density-based shape description consists of three main stages (see Fig. 1):

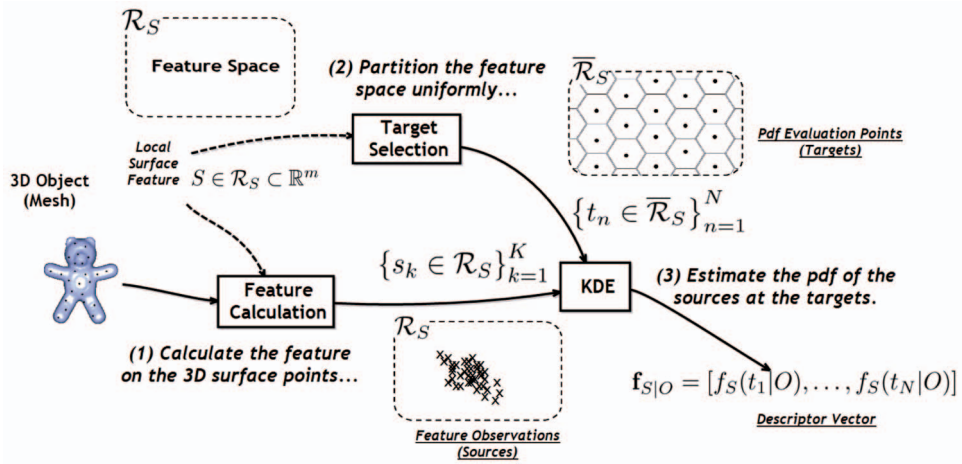

Fig. 1. Density-based shape description: Measurements of the (multivariate) feature S obtained from the 3D object surface are processed into descriptor vectors, that is, the probability density function of the feature.

1. In the *design stage*, we choose good local features that will accumulate to global shape descriptors. Good features are easy to compute and locally discriminative (Sections 3.1 and 3.2).
2. In the *target selection stage*, we determine the target set $\overline{\mathcal{R}}_S$ over which the feature pdf is evaluated (Section 3.3).
3. In the *computational stage*, we estimate $f_S(t|O)$ at the designated targets $t \in \overline{\mathcal{R}}_S$, using the KDE technique coupled with the fast Gauss transform (FGT) (Sections 3.4 and 4.3).

Once the descriptors of two different objects are computed by the above scheme, any vector distance can be used to compare them. To this end, we employ the L^1-distance or an invariant version of it (see Section 4.2) throughout the paper.

3.1 Local Surface Features

In this section, we describe the local geometric features that we employ to characterize 3D surfaces (see Fig. 2). We proceed from simple features that coarsely characterize the surface toward features that exploit differential geometry information.

3.1.1 Zero-Order Features

The most basic information about a point lying on a 3D surface can be derived from its coordinates, which we refer to as zero-order features. The *radial distance* $R \in (0, r_{max}]$ measures the distance of a surface point Q to the origin (center of the mesh) and is commonly used in shape description schemes such as [9], [11]. It may not be an effective shape feature all by itself, but it becomes useful, especially when other features need to be characterized separately at different quanta of the radial distance. The *radial direction* $\hat{\mathbf{R}} \in \mathcal{S}^2$ is the directional vector, collinear with the ray traced from the origin to the surface point Q. The $\hat{\mathbf{R}}$-vector lies on the unit 2-sphere $\mathcal{S}^2$ and is scale-invariant.

3.1.2 First-Order Features

First-order features require first-order differentiability, hence the existence of a tangent plane at each surface point, as illustrated in Fig. 2. For 3D meshes, one can compute a tangent plane at each vertex based on the triangle planes within the one-ring neighborhood [38]. In this category, the following features are considered:

First, the *normal direction* $\hat{\mathbf{N}} \in \mathcal{S}^2$ is simply the unit normal vector at a surface point and represented as a 3-tuple $(\hat{N}_x, \hat{N}_y, \hat{N}_z)$. Second, the *radial-normal alignment* A is the absolute cosine of the angle between the radial and normal directions, and computed as $A = |\langle \hat{\mathbf{R}}, \hat{\mathbf{N}} \rangle| \in [0, 1]$. This feature is a measure of the local surface deviation from sphericity. For example, if the surface locally approximates a spherical cap, then the radial and normal directions align and the A-feature approaches unity. Finally, the *tangent plane distance* $D = RA$ stands for the distance between the tangent plane at a surface point Q and the origin.

3.1.3 Second-Order Features

Second-order features around a point Q can be derived from the differential $d\mathbf{N}_Q$ of the normal field at that point [39]. By definition, $d\mathbf{N}_Q$ requires second-order differentiability. For triangular meshes, $d\mathbf{N}_Q$ can be computed by fitting a twice-differentiable surface patch around the vertex point and invoking standard formulas from differential geometry [39] or by discrete approximation using the mesh

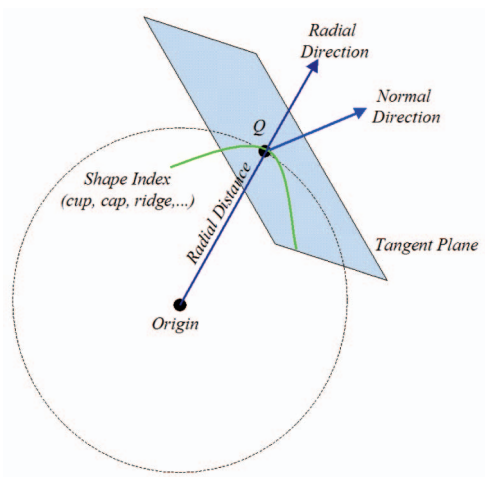

Fig. 2. Illustration of local surface features.

triangles within the one-ring of the vertex point [38]. We use the *shape index* SI as our second-order feature to provide a local categorization of the shape into primitive forms such as spherical cap and cup, dome, rut, ridge, trough, or saddle. We consider the parameterization proposed in [40] given by $SI = 1/2 - (1/\pi)arctan[(\kappa_1 + \kappa_2)/(\kappa_1 - \kappa_2)]$, where κ_1 and κ_2, the principal curvatures at the surface point Q, given, respectively, by the smallest and the largest eigenvalues of $d\mathbf{N}_Q$. The feature SI is confined within the range $[0, 1]$ and not defined when $\kappa_1 = \kappa_2 = 0$ (planar patch). It not only inherits the translation and rotation invariance of the principal curvatures, but also is a unitless quantity, hence scale-invariant.

3.1.4 Multivariate Characterization

Each of the above features reflects a certain aspect of the local shape. One can obtain a more thorough characterization of a surface point by constructing the feature vector $(R, \hat{\mathbf{R}}, \hat{\mathbf{N}}, SI)$. The multivariate pdf of this feature vector becomes a global descriptor, incorporating all local shape information up to second-order. Note that $(R, \hat{\mathbf{R}}, \hat{\mathbf{N}}, SI)$ is an 8-component feature with an intrinsic dimensionality of 6 within $(0, r_{max}] \times \mathcal{S}^2 \times \mathcal{S}^2 \times [0, 1]$. This fairly high dimensionality brings in concomitant problems of pdf estimation accuracy, high computation time, and huge storage size. For practical reasons, we design and work with the following multivariate pdf-based descriptors with manageable dimension:

- **R**-descriptor is the pdf of the coordinate representation $(R, \hat{\mathbf{R}})$ of the surface point.
- **T**-descriptor is the pdf of the $(D, \hat{\mathbf{N}})$-feature and aggregates the local tangent plane information.
- **S**-descriptor is the pdf of the (R, A, SI)-feature, which radializes the auxiliary alignment information A together with the second-order feature SI.

3.2 Feature Calculation

To obtain feature observations from a triangulated surface, we proceed on a per-triangle basis, taking also into account the shape of the triangle by a rigorous averaging scheme. To this effect, we compute the expected value of the local feature S over each triangle of the mesh. The expectation integral, when approximated in the discrete domain by applying Simpson's one-third numerical integration formula, boils down to taking a weighted average of the feature values calculated at nine adequately chosen points on the triangle (see [8] for details). For nonuniform meshes with low resolution, this averaging has the effect of smoothing the observations so that the subsequent pdf estimation can be performed more accurately. In contrast to histogram-based techniques, which sample features at isolated surface points, e.g., at barycenters of the triangles, this averaging scheme results in more reliable descriptors [8]. We should note that the Simpson averaging scheme does not apply to the shape index SI. The computation of the latter feature involves curvature estimation, which we carry out on a per-vertex basis using Taubin's algorithm [38]. To obtain per-triangle observations of SI, we take the average of the values at the three vertex points forming the triangle. In a similar way to Simpson averaging, this adds an implicit smoothing effect to the shape index calculation.

3.3 Target Selection

We define the target selection problem as sampling the range of the feature at which the pdf is evaluated. Since a density-based descriptor is a sampled version of a continuous pdf, we need to be efficient in choosing density evaluation points by exploiting any special structure of the feature range.

Once the real interval $\mathcal{I}$ of a scalar feature is determined, it is relatively simple to fix the targets. This can be done by partitioning the interval into $N_{\mathcal{I}}$ equally spaced (uniform) subintervals and by taking the midpoints. We determine the support $\mathcal{I}$ by clipping the tails of the scalar feature distribution from the lower and upper percentiles based on the empirical distribution obtained over a set of representative objects. Note that clipping the distribution makes sense only if the scalar feature has a magnitude interpretation such as the radial distance R or the tangent plane distance D for which too small and/or too large values can be considered as outliers. In fact, for the alignment A and the shape index SI, which both lie on the unit interval $[0, 1]$, the values near the boundaries are quite informative about the local shape. For these, we simply take equally spaced points within the unit interval.

For directional vector features $\hat{\mathbf{R}}$ and $\hat{\mathbf{N}}$, the corresponding target points should lie on the unit 2-sphere $\mathcal{S}^2$. Uniformly sampling the spherical coordinate intervals $\theta = [0, 2\pi)$ and $\phi = [0, \pi)$ would lead to overaccumulation of targets near the poles and to sparseness near the equator. To avoid this bias, we consider an octahedron circumscribed by the unit sphere, subdivide it into four each of its eight triangles, radially project the new triangles on the unit sphere, and iterate a factor of a-times the subdivision process recursively. The barycenters of the resulting triangles (after projecting back to the unit sphere) become the target set for directional features. This leads to an approximately uniform partitioning of the sphere. The recursion factor a determines the number of resulting points by $N_{\mathcal{S}^2} = 2^{2a+3}$.

As pointed out earlier, we can obtain informative multivariate local characterizations up to second order by joining scalar and directional features. In such cases, the target selection range occurs as the Cartesian product of the individual ranges of the features involved.

3.4 Kernel Density Estimation

The keystone of our shape description framework is a flexible and computationally efficient scheme for density estimation. We prefer the nonparametric KDE methodology with a Gaussian kernel in which case the density estimate is given by

$$f_S(t_n|O) = C \sum_{k=1}^{K} w_k e^{-\frac{1}{2}(t-s_k)^T H^{-2}(t-s_k)}, \qquad (1)$$

where $t_n = 1, \ldots, N$ and $C = ((2\pi)^{\frac{m}{2}}|H|)^{-1}$. We put this generic estimation procedure into the context of 3D shape description as follows.

Observations or *sources* $\{s_k \in \mathbb{R}^m\}_{k=1}^K$ are the feature values (or vectors) computed on the surface of an object O. They can be obtained from each of the mesh triangles, vertex points, or by the averaging scheme described in Section 3.2.

TABLE 2
Empirical Performance (DCG Percent) of Alternative Bandwidth Selection Strategies

Bandwidth Setting	**R**-Descriptor	**T**-Descriptor
Mesh-level	51.1	51.4
Database-level	57.0	59.8

Targets $\{t_n \in \mathbb{R}^m\}_{n=1}^N$ are the pdf evaluation points so that $f_S(t_n|O)$ values constitute the descriptor vector $\mathbf{f}_{S|O} = [f_S(t_1|O), \ldots, f_S(t_N|O)]$.

Weights $\{w_k \in \mathbb{R}\}_{k=1}^K$ stand for the importance of the sources. Naturally, a large triangle induces a more important observation. Accordingly, we set the weight w_k as the relative area of the kth mesh triangle (w.r.t. the total surface area).

Bandwidth parameter matrix $H \in \mathbb{R}^{m\times m}$ models the degree of uncertainty about the observations and controls the smoothing behavior of the KDE. Appropriate bandwidth selection is a critical issue for all applications using the KDE scheme. We deal with the effect of the bandwidth and the associated selection problem in Section 4.1.

Note that the nonparametric pdf KDE scheme makes fewer assumptions about the underlying generative model and, hence, provides further flexibility as compared to parametric approaches such as Gaussian mixture models. From the practical side, by choosing KDE, we can avoid computationally intensive parametric model estimation procedures like Expectation-Maximization or Markov Chain Monte Carlo methods, which would not be feasible in a practical retrieval application. Our choice of the Gaussian kernel in KDE is mainly motivated by such computational concerns. As discussed in Section 4.3 in detail, the complexity of KDE might be prohibitive unless one uses fast computational approximation schemes such as the FGT algorithm. FGT enables one to two order decrease in the computational load of evaluating large sums of Gaussians as in (1). Previous studies have indicated that the shape of the kernel does not critically affect the statistical accuracy [16]; in fact, it is stated in [41] that "the choice of the kernel function is almost irrelevant for the efficiency of the estimate." Therefore, in view of the computational advantage of the FGT algorithm, we find that the Gaussian kernel is a sound and well-motivated choice in our context.

4 Properties of DBF

In this section, we investigate the sensitivity of DBF to various perturbations, its pose invariance properties, and its computational complexity.

4.1 Bandwidth Selection and Sensitivity Analysis

KDE imbues the pdf estimation with smoothing property, so that it overcomes some of the inherent handicaps of histogram estimators [16]. Histograms are affected by the repositioning of the grid, hence of the setting of the origin, and they tend to have jaggy appearance especially in high dimensions. In KDE, on the other hand, the density estimate does not depend on the choice of origin and can be obtained in a continuum of target points. More importantly, KDE makes better use of the available observations $\{s_k\}_{k=1}^K$ as they all contribute, in principle, to the density estimate at all target points $\{t_n\}_{n=1}^N$ by a soft assignment strategy as can be seen from (1). This smoothing action of KDE is controlled by the bandwidth parameter.

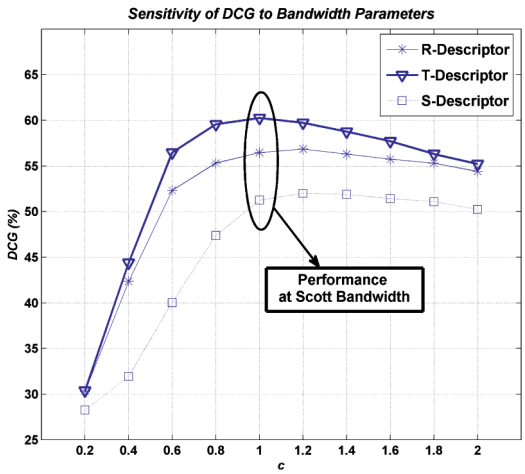

Fig. 3. Sensitivity of DCG to bandwidth parameters.

To set the ith diagonal entry of the $m \times m$ bandwidth matrix H, we use a commonly used rule-of-thumb selector, the Scott estimate [16] given by $h_i = (\frac{4}{K(m+2)})^{1/(m+4)}\sigma_i$, where σ_i is the standard deviation of the ith feature component, estimated from a given 3D mesh. The Scott bandwidth has been shown to be the optimal choice when features follow a Gaussian density [16]. Although this assumption is rarely fulfilled in practice, we adopt this rule-of-thumb as it provides us with an explicit and straightforward formula. Note that the Scott bandwidth matrices can be estimated for each mesh separately. However, we have found out that averaging the Scott bandwidth matrices over a representative set of models gives better empirical performance, as shown in Table 2 (see Section 5.1 for the definition of the DCG retrieval performance measure). Accordingly, we recommend the use of an averaged bandwidth matrix for all meshes in a training database (such as the PSB Training Set, see Section 5.1). This database dependence is further investigated in Section 5.4. Note also that we work with diagonal bandwidth matrices of the form $H = diag(h_1, \ldots, h_m)$ since off-diagonal entries become negligible after averaging.

To analyze the sensitivity of the empirical performance as a function of the bandwidth parameters, we perturbed the average Scott bandwidths h_i by scaling them as ch_i, with a common scalar factor $c \in [0.2, 2.0]$. In Fig. 3, we provide the DCG profiles as a function of c with steps of 0.2 for the **R**, **T,** and **S** descriptors. The best performances are attained for $c \in \{1.0, 1.2\}$, showing that average Scott bandwidth gives a sensible operational point. As expected, DCG degrades for undersmoothed ($c \leq 0.8$) and oversmoothed ($c > 1.2$) cases. Performance degradation for undersmoothed estimates is much steeper than the oversmoothed ones. Undersmoothed estimates reflect specific details about features; hence, descriptors become too much object-specific. In a follow-up experiment, we perturbed the component-wise Scott

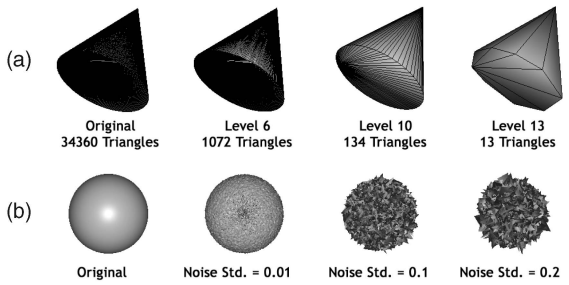

Fig. 4. (a) A cone model at different mesh resolutions and (b) a sphere model at various levels of noise contamination.

bandwidths h_i independently by choosing 100 uniformly distributed values within the interval $[0.8h_i, 1.2h_i]$ for each. The average DCG statistics over these 100 descriptor sets for the **R**, **T**, and **S** descriptors were very close to the Scott bandwidth performances, 57 percent, 60 percent, and 52 percent respectively, with negligible variations. This is a further evidence of the breadth of optimality, which extends over 40 percent (± 20 percent) of the Scott bandwidth in terms of empirical DCG performance.

Increasing the bandwidth leads to smoother density estimates. In the context of 3D shape description, this helps to gloss over uncertainties that may arise from changes in the mesh resolution and/or mesh degeneracies, measurement noise and small shape variations, and pose normalization errors. To illustrate this, we carried out a sensitivity analysis of our descriptors against three types of nuisances: changing mesh resolution, additive Gaussian noise, and small rotation errors. We considered three bandwidth choices, set as the Scott bandwith, one-fifth the Scott bandwidth (undersmoothed), and twice the Scott bandwidth (oversmoothed).

4.1.1 Sensitivity against Low Mesh Resolution

Since mesh resolution is not an intrinsic shape property, descriptors should be insensitive to changes in the level of detail. To investigate the sensitivity of the density-based descriptors to resolution, we computed them on successively subdivided versions of a 3D cone model (see Fig. 4). The left column of Fig. 5 shows the changes in the descriptor according to the L^1-distance (with respect to the original cone mesh) as a function of decreasing resolution for all descriptors. These sensitivity profiles put forth a general pattern, which corroborates the fact that the variation can be reduced by increasing the bandwidth (which leads to oversmoothed density estimates). In fact, the profiles for the R, D, A, and $\hat{\mathbf{N}}$ features exhibit very low or negligible variation even for low resolution versions of the cone model. On the other hand, controlling the variation of the notoriously sensitive SI descriptor is somewhat problematic and the $2 \times \text{Scott}$ bandwidth option is not sufficient. Note, however, that the smoothing principle does not change and an even larger bandwidth can be employed to make the SI descriptor less sensitive.

4.1.2 Sensitivity against Noise

The middle column of Fig. 5 depicts the sensitivity of our descriptors when Gaussian noise at various levels is added to vertex points of a sphere mesh (see Fig. 4). Here again, a large bandwidth ($2 \times \text{Scott}$) reduces the effects of noise on descriptor variation. The difficulty of coping with the SI descriptor is recurrent. Even for very low levels of noise, its variation is dramatically high.[1] In the presence of noise, we can no longer rely on the SI descriptor.

4.1.3 Sensitivity against Pose Normalization Errors

Small pose uncertainties can arise due to the imperfections of standard pose normalization methods that work best for relatively compact and elliptical objects [24]. For example, slight changes in postures of articulated objects, even though they may not be semantically relevant, can give rise to pose perturbations. The smoothing behavior of KDE can be exploited to combat such deficiencies. To make this point, we have generated randomly rotated versions of a cylinder at increasing levels of angular deviation and evaluated the descriptor variations. Variation profiles for the $\hat{\mathbf{R}}$ and $\hat{\mathbf{N}}$ descriptors are displayed in the rightmost column of Fig. 5. The remaining features are not considered as they all are rotation-invariant by definition. We see that, even for deviations as large as 30°, it is possible to keep the descriptor variation negligible (when $h = 2 \times \text{Scott}$).

With a judicious choice of the KDE bandwidth parameter, one can mitigate the effects of low mesh resolution, slight pose perturbations, and measurement noise. Note, however, that these robustness advantages should be assessed to the degree they compromise the discrimination ability of the descriptor. The empirical performance curves in Fig. 3 tells us that while a slight oversmoothing of the descriptors can even provide some performance gain, undersmoothing is completely detrimental to the descriptor's classification power.

4.2 Invariance Properties

Geometric transformations of 3D objects, such as translation, rotation, reflection, and isotropic rescaling (collectively denominated as similarity transformations), are often viewed as nuisance effects that must be eliminated as they do not contribute to the semantic classification of the shape. In DBF, invariance is achieved in two stages: 1) by preprocessing the object before descriptor extraction and then 2) by postprocessing the descriptor before shape matching. Note that there exist methods in the 3D shape description literature where certain invariances are guaranteed by the design of features, for example, taking only the magnitudes of spherical harmonics coefficients against rotational effects [31]. However, such invariances come with some loss of shape information. We think this can be avoided by normalizing objects before descriptor extraction and postprocessing descriptors at matching stage. Automatic derivation of a canonical 3D reference frame against rotations and reflections still remains as an open problem, notwithstanding all the efforts in the literature [24].

1. Variations cannot go beyond 2 since we work with normalized descriptors, in which case the L^1-distance between two descriptors is upper-bounded by 2.

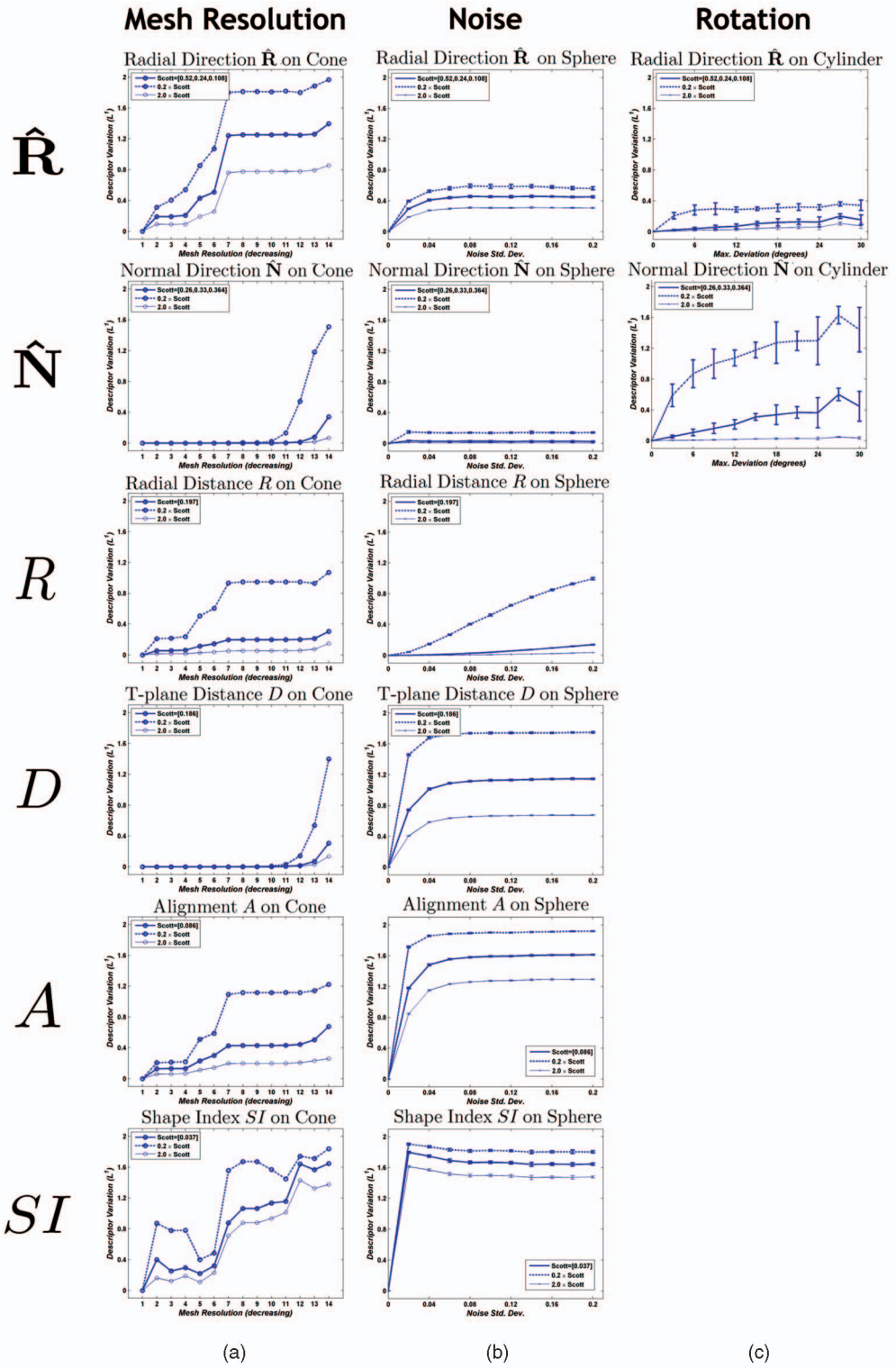

Fig. 5. L^1-distance of descriptors with respect to the baseline versus (a) decreasing mesh resolution, (b) increasing additive Gaussian noise, and (c) increasing amounts of random rotation errors. The estimated Scott bandwidth(s) is written in brackets in the legend. The other two graphs are obtained at 20 percent and 200 percent of the estimated Scott bandwidth(s).

4.2.1 Preprocessing for Invariance

Principal component analysis (PCA) and its variants [24] constitute practically a universal tool for 3D pose normalization although they are not always very stable to variations of the object's shape even in a semantically well-defined class and might result in counterintuitive alignments. From an operational point of view, we opt to carry out the following normalization steps:

Translation. The object's center of mass is considered as the origin of the 3D coordinate system. We calculate the center of mass as the area-weighted average of triangle barycenters.

Rotation and Reflection. We use Vranic's "continuous" PCA approach, where the covariance matrix of the surface points is evaluated by integrating over triangles instead of mere area-weighted averaging [24]. The axes of the coordinate system are found by the eigendecomposition of the estimated covariance matrix and the x, y, z labeling of the axes is assigned according to the decreasing rank of the eigenvalues (invariance to axis relabelings), while the polarities are estimated by Vranic's moments-based approach (invariance to mirror reflections) [24]. However, this axis labeling and polarity assignment strategy does not always yield consistent results. We specifically address this shortcoming in the following postprocessing section.

Isotropic Rescaling. We calculate a scale factor as the area-weighted average of the surface point-to-origin distances and rescale the objects by dividing the surface point coordinates by this factor.

4.2.2 Postprocessing for Invariance

Given the equivocation of the axis labeling and polarity assignments, the invariance of DBF scheme can be corroborated by exhaustively testing within an appropriate set of 3D transformations what the object might undergo. In DBF, the matching stage becomes only slightly computationally more intensive since we do not have to recalculate the descriptor for each transformation but just to permute its entries, as we describe in the sequel.

Consider a generic pose-dependent random surface feature $S \in \mathcal{R}_S$ with pdf f_S and a bijection Γ on $\mathcal{R}_S$. The pdf of S is related to the pdf of its transformed version $\Gamma(S)$ by $f_S(s) = f_{\Gamma(S)}(\Gamma(s))|J_\Gamma(s)|$, where J_Γ is the Jacobian of Γ. When we restrict Γ to orthogonal transformations accounting for rotations, relabelings, and mirror reflections of the coordinate axes, neither the intrinsic shape properties of the object O nor the shape information in the feature S is changed. For orthogonal transformations, we have $|J_\Gamma(s)| = |\Gamma| = 1$, thus $f_S(s|O) = f_{\Gamma(S)}(\Gamma(s)|O),\ \forall s \in \mathcal{R}_S$. Matching of a test object O' to a reference object O invariantly to any orthogonal transformation Γ can then be carried out by searching the minimum distance

$$Inv\text{-}L^1(O, O') = \min_{\Gamma \in \mathbf{O}(3)} \int_{s \in \mathcal{R}_S} \left| f_{\Gamma(S)}(\Gamma(s)|O) - f_S(s|O') \right| ds, \tag{2}$$

where $\mathbf{O}(3)$ is the set of orthogonal transformations in $\mathbb{R}^3$ and where L^1-distance is used as the base distance. Given two DBF descriptors $\mathbf{f}_{S|O}$ and $\mathbf{f}_{S|O'}$, the invariant measure in (2) gets the form:

$$Inv\text{-}L^1(O, O') = \min_{\Gamma \in \mathbf{O}(3)} \left\| \mathbf{f}_{\Gamma(S)|O} - \mathbf{f}_{S|O'} \right\|_{L^1}, \tag{3}$$

provided that the target set $\overline{\mathcal{R}}_S$ provides a uniform partitioning of the feature space $\mathcal{R}_S$. The search for the minimum in (3) is practical only when the pdf values $f_{\Gamma(S)}(\Gamma(t_n)|O)$, i.e., at the new targets $\{\Gamma(t_n)\}$, can be obtained directly from the stored pdf values $f_S(t_n|O)$. This is feasible whenever $\overline{\mathcal{R}}_S$ is closed under the action of Γ, i.e., when $\forall t \in \overline{\mathcal{R}}_S, \Gamma(t) = t' \in \overline{\mathcal{R}}_S$. In such a case, Γ maps the target t to another target t' so that we have $f_S(t|O) = f_{\Gamma(S)}(t'|O)$. Accordingly, for a given descriptor $\mathbf{f}_{S|O}$ and any orthogonal Γ, the descriptor $\mathbf{f}_{\Gamma(S)|O}$ is obtained simply by permuting the components $f_S(t_n|O)$ of the vector $\mathbf{f}_{S|O}$. The minimum in (3) can then be found by exhaustively testing all the admissible transformations Γ that leave the target set closed.

In this work, we restrict the set of admissible transformations to axis relabelings and mirror reflections, forming a subset of $\mathbf{O}(3)$ that we denote by $\mathbf{PS}(3)$. This restriction is mainly computational since the cardinality of $\mathbf{PS}(3)$ is only 48. Notice that the three coordinate axes can be labeled in $3! = 6$ possible ways and, for any given labeling, there are $2^3 = 8$ possible polarity assignments, which result in $6 \times 8 = 48$ possible xyz-configurations. The target set $\overline{\mathcal{R}}_S$ for pose-dependent features $\hat{\mathbf{R}}$ or $\hat{\mathbf{N}}$ obtained by octahedron subdivision (cf. Section 3.3) remains closed under $\Gamma \in \mathbf{PS}(3)$. If the regular octahedron has its center placed at the origin and its six vertices at $\{(\pm 1, 0, 0), (0, \pm 1, 0), (0, 0, \pm 1)\}$, first, its appearance will not depend on the axis labeling: After an axis permutation, we will recover the same vertex coordinates. Second, the octahedron is symmetric with respect to each of the xy, xz, and yz-planes: After a reflection, we again recover the same coordinates. This nice property of the octahedron holds for all of its successive subdivisions of any order, making the resulting target set $\overline{\mathcal{R}}_S$ closed under $\Gamma \in \mathbf{PS}(3)$. Furthermore, each such Γ corresponds to a unique permutation of descriptor vector entries so that the invariant metric over $\Gamma \in \mathbf{PS}(3)$ can be very efficiently implemented via a look-up table. In fact, assuming that the cost of a permutation is negligible, the complexity of the invariant metric is just 48 times the complexity of a vector-to-vector comparison by L^1-distance. The matching runtimes of this invariant scheme are provided in Table 7 at the end of Section 5.5.

Once mislabelings and/or erroneous polarity assignments are compensated for, any small rotation errors after normalization can be taken care of by the smoothing effect of the KDE-based scheme as discussed in Section 4.1. The significant performance improvements of this invariant matching scheme are presented in Section 5.2.

4.3 Computational Complexity

The computational complexity of KDE using (1) directly is $O(KN)$, where K is the number of observations (the number of triangles in our case) and N is the number of density evaluation points, i.e., targets. For applications like content-based retrieval, this $O(KN)$-complexity is prohibitive. Hopefully, the FGT algorithm is able to reduce the computational complexity significantly. For example, on a Pentium 4 PC (2.4 GHz CPU, 2 GB RAM) and for a mesh of 130,000 triangles, the direct evaluation of a 1,024-point pdf-descriptor takes 125 seconds, while FGT takes only 2.5 seconds. FGT is an approximation scheme enabling the calculation of large sums of Gaussians within reasonable accuracy and reducing the complexity down to $O(K + N)$ [20], [21]. In our 3D shape description system, we have used an improved version of FGT implemented by Yang et al. [21].

TABLE 3
DCG (Percent) Performance of L^1 versus *Inv*-L^1 on Different Databases for the **R** and **T**-Descriptors

	R-Descriptor			**T**-Descriptor		
Database	L^1	*Inv*-L^1	**Gain**	L^1	*Inv*-L^1	**Gain**
PSB Train	57.0	61.2	4.2	59.8	64.9	5.1
PSB Test	54.9	57.9	3.0	57.8	61.4	3.6
SCU	71.3	74.4	3.1	72.0	76.2	4.2
SHREC-W	74.4	78.0	3.6	80.3	82.6	2.3
ESB	68.8	70.1	1.3	73.7	75.4	1.7

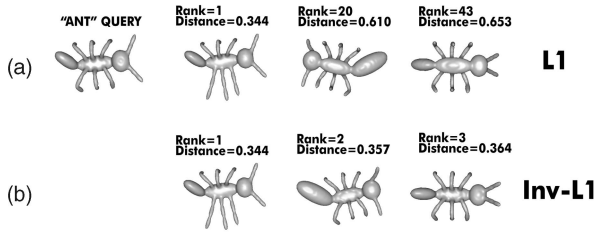

Fig. 6. A sample "ant" query from SHREC-W database retrieved (a) using L^1-distance and (b) using *Inv*-L^1-distance.

5 Experimental Results

5.1 Databases and Evaluation Tools

To demonstrate the retrieval potential of DBF in a wide range of applications, we have experimented with four 3D databases selected from different domains.

Princeton Shape Benchmark (PSB) [23] contains 1,814 general-purpose low-quality 3D models. The base ground-truth classification consists of a training set (907 models in 90 classes) and a test set (907 models in 92 classes). Classification is induced by functionality as well as by form. In general, PSB meshes have low resolution; they are nonregular, nonsmooth, and contain degeneracies such as nonmanifold, nonconnected triangles of varying size and shape. These models are usually referred as "triangular soups."

Sculpteur (SCU) [2] contains 513 high-quality 3D models in 53 classes consisting mainly of archaeological models. SCU meshes are regular, smooth, and highly detailed in terms of resolution.

SHREC Watertight (SHREC-W) [42] contains 400 high-quality 3D models in 20 classes. This database is special in that the classification takes into account not only the geometrical similarities between shapes but also their topological equivalences. Hence, it constitutes a challenging test environment for geometry-induced description methods. SHREC-W meshes are regular and smooth.

Purdue Engineering Shape Benchmark (ESB) [1] contains 865 3D models of engineering parts in 45 classes. ESB meshes are regular but in general nonsmooth due to the general crisp geometrical nature of engineering parts, composed of large flat patches along with many joints and sharp ridges.

PSB and SHREC-W represent two extremes in terms of mesh regularity and smoothness. In SCU and SHREC-W meshes, second-order differential structure is locally present at every mesh point in contrast to PSB and ESB where second-order geometry is either not very informative or difficult to be analyzed.

In our comparative analyses, we have used the following statistics to measure the retrieval performance:

Precision-Recall curve. For a query q which is a member of a certain class $\mathcal{C}$ of size $|\mathcal{C}|$, *Precision* (vertical axis) is the ratio of the relevant matches K_q (matches that are within the same class as the query) to the number of retrieved models K_{ret}, and *Recall* (horizontal axis) is the ratio of relevant matches K_q to the size of the query class $|\mathcal{C}|$. Ideally, this curve should be a horizontal line at unit precision.

Nearest Neighbor (NN). The percentage of the first-closest matches that belong to the query class. A high NN score indicates the potential of the algorithm in a classification application.

Discounted Cumulative Gain (DCG). DCG is a statistic weighting correct results at the top of the list more than those appearing later. To calculate this measure, the ranked list of retrieved objects is converted to a list L, where an element L_k has value 1 if the kth object is in the same class as the query, and otherwise, has value 0. Discounted cumulative gain at the kth rank is then defined as $\mathrm{DCG}_k = \mathrm{DCG}_{k-1} + \frac{L_k}{log_2(k)}, k \geq 2; \mathrm{DCG}_1 = L_1$. The final DCG score for a query $q \in \mathcal{C}$ is the ratio of $\mathrm{DCG}_{K_{max}}$ to the maximum possible DCG that would be achieved if the first $|\mathcal{C}|$ retrieved elements were in the class $\mathcal{C}$, where K_{max} is the total number of objects in the database.

Normalized DCG (NDCG). This statistic is based on averaging DCG values of a set of algorithms on a given database. A positive NDCG indicates an above-the-average performance. Let $\mathrm{DCG}^{(A)}$ be the DCG of a certain algorithm A and $\mathrm{DCG}^{(avg)}$ be the averaged DCG over a set of algorithms tested on the same database, then NDCG for the algorithm A is defined as $\mathrm{NDCG}^{(A)} = \mathrm{DCG}^{(A)} / \mathrm{DCG}^{(avg)} - 1$.

All of these quantities are normalized within the range $[0, 1]$ (except NDCG) and higher values reflect better performance. The retrieval statistics presented in this work are obtained using the utility software included in PSB [23].

5.2 Invariant Matching Results

In this section, we prove experimentally that the invariant matching scheme described in Section 4.2 improves the retrieval effectiveness for all databases. Table 3 shows that additive DCG gains corresponding to **R** and **T**-descriptors are significant, e.g., a 5.1 percent DCG point improvement is obtained using the **T**-descriptor on PSB Training set. The performance increase for ESB is more modest (1.3 percent for **R**-descriptor and 1.7 percent for **T**-descriptor) compared to other databases.

The sample "ant" query depicted in Fig. 6 illustrates how invariance against coordinate axis relabelings and mirror reflections can provide better matches. Top row displays correct models with their ranks and L^1-distances to the query. Consider, for instance, the second correct item retrieved at as far as the 20th position. Clearly, the problem here is that PCA was not able to align the query and database models coherently: The "head" parts of the ants look opposite sides after normalization. Invariant matching aligns the models correctly, consequently lifting the position of the database model to second rank under *Inv*-L^1 as compared to its 20th rank order under L^1 (see the bottom

TABLE 4
Effect of the Invariant Scheme on DCG Performance

Descriptor	Database	# Classes	(+) Classes	(-) Classes	Best Class-Wise DCG Gain (%)	Worst Class-Wise DCG Loss (%)
R	**PSB Train**	90	64	25	35.0	4.1
	PSB Test	92	62	30	24.5	8.8
	SCU	53	31	20	26.2	4.9
	SHREC-W	20	16	4	11.6	1.9
	ESB	45	12	7	16.7	5.9
T	**PSB Train**	90	66	24	41.1	14.1
	PSB Test	92	66	25	30.9	11.3
	SCU	53	30	19	30.6	4.8
	SHREC-W	20	13	7	11.3	7.6
	ESB	45	24	18	16.1	3.4

 # Classes: Total number of shape classes
(+) Classes: Number of classes for which the invariant scheme improved DCG on average
(-) Classes: Number of classes for which the invariant scheme degraded DCG on average

row of Fig. 6). Similar observations hold for the remaining matches and *Inv-*L^1 provides much better results (the first four matches are correct). Note also that the "ant" class is a difficult shape category for geometry-induced descriptors (like ours) due the intrinsic geometric variability of its members sharing the same topology. In this example, the invariant scheme alleviates the problem noticeably.

Table 4 provides class-wise performance gain-loss results. We see that roughly two thirds of shape classes profit from the exhaustive search for coordinate axis relabelings and mirror reflections. However, one third of classes incur into performance loss, though the overall balance remains positive for all classes and databases. A plausible explanation for the performance loss of the metric is the semantic gap between the geometry information encoded by the descriptor and the purported functionality of the 3D model. In other words, geometric invariance cannot always resolve the semantics of the shape.

5.3 Discriminativeness of Features on Different Databases

In this section, we address the following two questions: 1) Given a database, which local feature is the most effective? 2) Given a feature, which database is the most challenging for the retrieval task?

To illustrate the effectiveness of various feature sets, we introduce a graphical tool, called *DCG-eye diagrams*. As shown in Figs. 7 and 8, a DCG eye diagram is the graphical representation of the DCG score represented as an inscribed painted disk within a unit circle. The radius of the disk is the DCG score while the unit circle represents the maximum achievable DCG (100 percent). Obviously, the bigger the "blue eye" or the smaller the residual area, the higher the performance. These eye diagrams help us visualize the effectiveness of features and their database dependence. The **R** and **T**-features are evaluated via *Inv-*L^1-distance while the **S**-feature, which is pose-invariant by definition, is evaluated under the L^1-distance. From Fig. 7, we see that, for all databases, the **T**-feature performs better than the **R** and **S**-features, indicating that first-order surface information is more discriminative than zero or second-order. The **S**-feature is inferior on all three databases, except for SHREC-W, where it is on a par with the **T**-feature. The performance of the **S**-feature depends upon whether the database shapes allow for the reliable computation of the second-order features. In fact, for rough meshes as in PSB or manufactured surfaces as in ESB, the shape index *SI* feature is either not defined everywhere or unreliable. Therefore, wherever curvature estimation (required for *SI*) is not reliable, the **S**-feature (an augmented version of *SI*) has unstable performance across different databases.

The **R**, **T,** and **S**-features encode zero, first, and second-order local surface information, respectively; hence, they probe different aspects of shapes. Thus, when joined together into a larger multivariate feature, they would be more

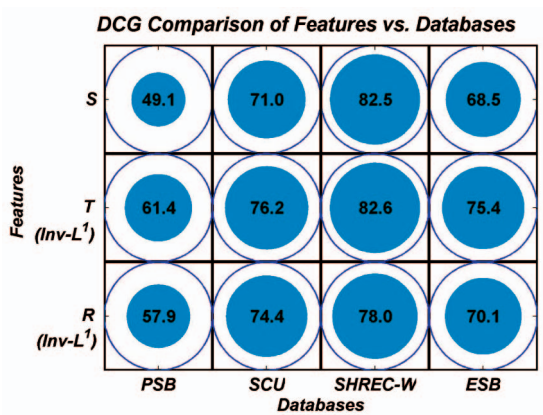

Fig. 7. Eye diagram illustration of DCG scores in comparing features and databases. The iris (inner circle) is proportional to the inscribed DCG score.

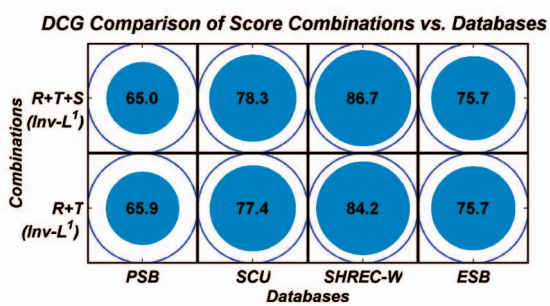

Fig. 8. Eye diagram illustration of DCG scores in comparing score combinations and databases. The iris (inner circle) is proportional to the inscribed DCG score.

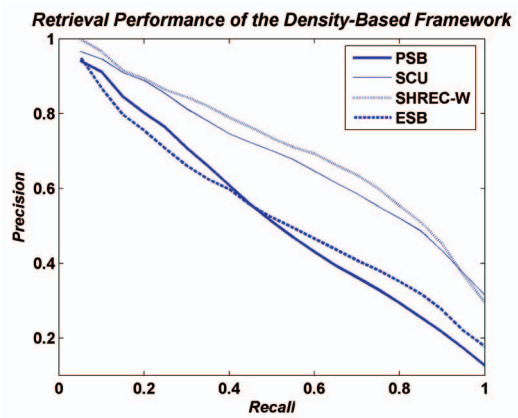

Fig. 9. Precision-recall curves for the four databases under the best score combinations

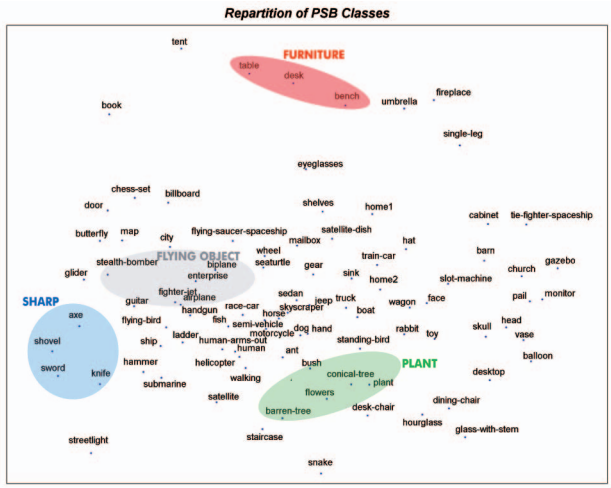

Fig. 10. Multidimensional scaling map for PSB Test Set classes.

effective. Ideally, these features should be fused by calculating their joint density function at designated target points. However, we do not advise the use of multivariate features with dimensions greater than five because the quality of KDE degrades due to curse of dimensionality and also descriptors become prohibitively large. An alternative way to benefit from them simultaneously is score fusion, for example, by summing their individual distance values. Although this unsupervised score fusion approach cannot exploit potential correlations between features, it constitutes a working alternative to combine shape similarity information from different descriptors. There are more sophisticated approaches that use some supervision to learn the optimal set of combination weights [43]. Fig. 8 depicts the DCG performance of this basic score fusion scheme. We have denoted score combinations with a "+" sign, e.g., **R**+**T**, since we sum their distances. From Fig. 8, we observe that it is always beneficial to sum the scores of the **R** and **T**-features. **S**-score contributes only to SHREC-W and SCU databases, and it actually worsens the performance for PSB. This is not surprising given the unrealiable calculation of second-order surface characteristics in low-quality meshes.

In Fig. 9, we display the precision-recall curves for the four databases under the best score combinations. The DCG performance ordering of DBF is $\text{SHREC-W} \succ \text{SCU} \succ \text{ESB} \succ \text{PSB}$, where $\succ$ denotes the performance order relation. PSB is the most challenging database not only because the numbers of objects (907) and classes (92) are higher than the others, but, more importantly, its classification is induced mostly by functionality-driven semantics. Since form does not always follow functionality, discrimination relying solely on global shape description has its limitations. It has been shown that such databases can profit from some user interaction that incorporates functionality-driven semantics in the search criteria [43]. In Fig. 10, we depict the repartition of PSB classes obtained by multidimensional scaling (MDS), applied on the average between-class distances given by our descriptors. MDS is a method to map pairwise dissimilarities to a lower dimensional display where the distance ordering between pairs of entities is preserved as much as possible [44]. The MDS mapping of PSB classes is open to many interesting observations showing that DBF captures functionality-driven semantics to some extent. For instance, in the southwestern part of the map, we see a clustering of *sharp* object classes, such as axe, knife, shovel, and sword. Fig. 10 provides many more of such visible groupings.

The least challenging database seems to be SHREC-W. Although it contains a small number of shape classes relative to the remainder, the fact that its classification is induced by topological equivalences makes it a difficult database to describe by geometry-based methods like ours. It is interesting to observe that the geometry-based DBF performs so well on SHREC-W. We also note that, in the Watertight Track of the SHREC'07 event, our shape description scheme had overall the second rank, outperforming other geometry-based methods [42]. The performance on ESB as measured by precision-recall is less impressive than the performance reflected by the DCG score (75.7 percent). In fact, the performance on ESB for $Recall < 0.4$ is worse than on the more difficult PSB where our method attains 65.9 percent DCG. It seems that on ESB, DBF does not always find the correct match in the upper part of the list but that it is eventually capable of retrieving them, thus lifting the DCG to a satisfactory value.

5.4 Database Dependence of the Bandwidth Parameters

Table 5 summarizes the performance changes due to bandwidth parameters estimated in a given database, but used to compute descriptors in another database. In this table, the row of a cell indicates the database on which descriptors are computed and its column indicates the database from which bandwidth parameters are estimated. Accordingly, diagonal cells corresponding to a certain descriptor must have zeroes while other cells contain performance drops or increases as indicated by the algebraic sign of the performance differential. It is encouraging to observe that, in general, the performance does not significantly depend on the training database, that is, the database where bandwidth parameters are estimated. Some minor gains or losses may occur. Exceptions are the **S**-descriptor when the ESB is used for bandwidth estimation and, to a lesser extent, the **T**-descriptor when SCU or SHREC-W are chosen. In fact, we can anticipate the degradation in the performance of the **S**-descriptor when the bandwidth is estimated from ESB

TABLE 5
DCG Differences due to Bandwidth Parameters Estimated from a Given Database for the **R**, **T**, and **S**-Descriptors

		PSB Tr.	PSB Te.	SCU	SHREC	ESB
R	**PSB Tr.**	0.0	+0.2	-1.8	-0.2	-1.4
	PSB Te	-0.1	0.0	-1.0	-0.1	-1.3
	SCU	+0.1	+0.3	0.0	+0.6	+0.2
	SHREC	-0.9	-0.8	0.1	0.0	-0.7
	ESB	0.7	0.4	-0.1	0.6	0.0
		PSB Tr.	**PSB Te.**	**SCU**	**SHREC**	**ESB**
T	**PSB Tr.**	0.0	-0.1	-4.2	-3.8	-0.3
	PSB Te.	0.1	0.0	-2.3	-2.0	-0.2
	SCU	+1.9	+2.0	0.0	+0.4	+1.6
	SHREC	+1.9	+1.8	-0.1	0.0	+1.3
	ESB	-0.3	-0.3	-3.0	-2.5	0.0
		PSB Tr.	**PSB Te.**	**SCU**	**SHREC**	**ESB**
S	**PSB Tr.**	0.0	+1.7	+0.9	+0.7	-4.5
	PSB Te.	-2.0	0.0	-0.2	-1.0	-5.2
	SCU	-2.8	-0.9	0.0	-1.6	-6.4
	SHREC	-0.2	+0.4	-0.8	0.0	-4.8
	ESB	+1.4	+2.2	+2.4	+2.1	0.0

because this database consists of machine part models for which the second order SI-feature cannot faithfully describe the local shape. If bandwidth parameters are to be estimated from one database and then used over several others, intuition tells that the training database must possess as much variety as possible: In this context, we can recommend bandwidths estimated from PSB.

5.5 Comparative Performance Analysis

5.5.1 Retrieval Comparisons on PSB

In this section, we first compare the retrieval performance of DBF against histogram-based, transform-based, and 2D view-based shape description methods on PSB. The statistics given in Table 6 are taken from either their original works or the survey study in [23]. For DBI, SIL, and DSR, we have used the executables provided in [25]; for CAH and 3DHT, we have used our own implementation. For DBF, we have taken the $Inv\text{-}L^1$ score combination of **R** and **T** descriptors. Fig. 11 illustrates the performance landscape using a DCG versus NN scatter plot, where the methodological category of a descriptor is indicated by a symbol, e.g., transform-based methods are marked as black circles. Top descriptors are CRSP, DSR, and DBF with very close performance scores, within a DCG difference less than 1 percent. DCG and NN performances are, in general, positively correlated except for the Spherical Wavelet Descriptor (SWD), which has a very good DCG but a poor NN score. It seems that SWD is able to retrieve models correctly, but with a lower ranking. We also observe that our DBF method has the best NN performance among all.

Joint consideration of DCG and NN performances suggests the following methodological clusters: $\{\text{CRSP}, \text{DSR}, \text{DBF}\} \succ 2D\ Views \succ Transforms \succ Histograms$. We draw the following conclusions concerning the effectiveness of these methods:

Invariant Matching after PCA Normalization Improves the Performance. Top performing methods {CRSP,DSR,DBF} all use PCA-based pose normalization rather than adhering to descriptors that are pose-invariant by definition. However, they also take additional measures to enhance invariance. For example, CRSP adopts a double pose normalization strategy incorporating an additional PCA procedure applied to surface normals [33], while DBF benefits from an invariant matching scheme for coordinate

TABLE 6
Retrieval Statistics (Percent): State-of-the-Art 3D Shape Descriptors on PSB Test Set

Desc.	CRSP	DSR	**DBF**	SWD	LFD	DBI	REXT	SIL	RISH	3DHT	SHIST	GSD	EGI	D2	CAH
NN	67.9	66.5	**68.6**	46.9	65.7	60.9	60.2	55.7	55.6	58.8	54.6	43.4	37.7	31.1	33.2
DCG	66.8	66.5	**65.9**	65.4	64.3	61.4	60.1	59.7	58.4	57.7	54.5	49.3	47.2	43.4	43.3
NDCG	16.4	15.9	**14.9**	14.0	12.1	7.0	4.7	4.1	1.8	0.6	-5.0	-14.1	-17.7	-24.4	-24.5

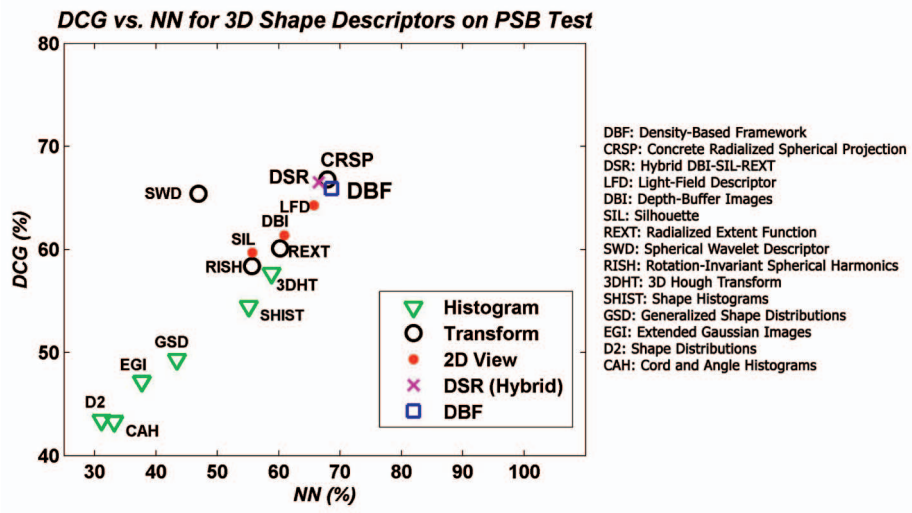

Fig. 11. Performance plot of DCG versus NN of the 3D shape descriptors evaluated on PSB Test Set. The legends in the inset denote the methodological categories, e.g., the circle represents histogram-based methods.

TABLE 7
Average Extraction, Comparison, and Total Matching Times (Seconds) for the **R**+**T**+**S** Descriptor Combination (on a Pentium 4 PC 2.4 GHz CPU, 2 GB RAM)

	PSB Train	PSB Test	SCU	SHREC-W	ESB
Number of Ovjects	907	907	513	400	865
Avg. # Observations K	7326	7960	175448	20548	3729
Extraction	0.2546	0.2838	3.1514	0.4370	0.1751
Comparison	0.001286	0.001276	0.001231	0.001074	0.001246
Total Matching	1.1652	1.1561	0.6303	0.4284	1.0769

axis labelings and mirror reflections. Comparison of performances indicates that it is preferable to work with absolute features (embedded in a canonical coordinate frame like the one obtained with PCA) rather than with features that are invariant by definition (as in RISH [31]).

Information Fusion Boosts the Performance. Two of the top performing methods, DSR and DBF, combine different types of shape information to boost the retrieval performance. This suggests that the performance can be improved even further by using a diverse set of descriptors with complementary shape description capabilities, as will be further addressed in detail in the next section.

Multiview 2D Data Capture 3D Shape Characteristics Well. The hybrid DSR-descriptor from the top cluster is based upon two 2D view-based descriptors (DBI and SIL) which, along with LFD, have generally better performance than 3D transform-based and histogram-based methods. This proves that describing a 3D object by a collection of 2D views leads to effective retrieval algorithms.

Distribution-Based Methods can be Effective. Histogram-based descriptors can be viewed as methodological ancestors of DBF, as both approaches rely on the idea of accumulating feature information to obtain a global shape description. While histogram-based methods are placed at the lower end of the performance landscape, DBF is in the top cluster. The lessons learned are twofold: Applying the right density estimation scheme (i.e., KDE) captures shape information that is missed by histograms and a multivariate local surface characterization is essential for effective retrieval.

Table 7 provides average extraction, comparison, and total matching times for the **R**+**T**+**S** descriptor combination on a Pentium 4 PC (2.4 GHz CPU, 2 GB RAM) over all databases. It can be observed from this table that DBF descriptors can be very efficiently extracted and compared. First, descriptor extraction, which is dominated by density estimation, can be performed rapidly thanks to FGT (cf. Section 4.3). Notice that the density estimation time is proportional to the mesh resolution, so that this step can be performed in $O(K+N) \approx O(K)$ since the number of observations K (i.e., the number of triangles per mesh) is usually much higher than the number of targets N (i.e., the size of the descriptor). Second, the complexity of comparing two descriptors (of type **R** or **T**) is just 48 times the complexity of a vector-to-vector distance computation (cf. Section 4.2). Consequently, it takes only $\sim$1 second to match a query descriptor against a database of $\sim$1,000 objects (see the sixth row of Table 7). Furthermore, even without any dimensionality reduction and compression, the storage of these descriptors coded with 16-bit double precision costs only 12 KB on average per object.

As a final remark, we note that even better retrieval performance on the PSB data set has been reported using the priority-driven search (PDS) method [26]. PDS belongs to the paradigm of matching by feature correspondences with no underlying global and compact shape description. However, as the authors report in [26], this algorithm demands computationally intensive database preprocessing (4-5 minutes per object, 200-300 times slower than DBF), considerable storage size (100 KB per object, 10 times more than DBF), and more time to find matches than descriptor-based methods. The DCG score of PDS is reported as 75.9 percent on PSB database, 10 percent better than the cluster {CRSP,DSR,DBF}, indicating that, performance-wise, there is more way to go for descriptor-based 3D shape retrieval schemes.

TABLE 8
DCG (Percent) Precision versus Recall of DBF, DSR, and Their Combinations DBF+DSR on the Four Tested Databases

Descriptor	PSB Test	SCU	SHREC-W	ESB
DBF	65.9	78.3	86.7	75.7
DSR	66.5	76.6	83.2	74.1
DBF+DSR	70.2	78.8	87.2	76.9

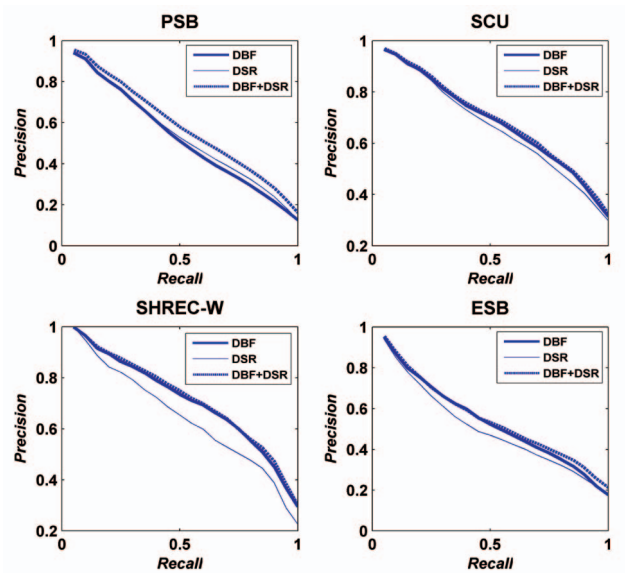

Fig. 12. Precision Versus Recall of DBF, DSR, and their combinations DBF+DSR on all databases tested.

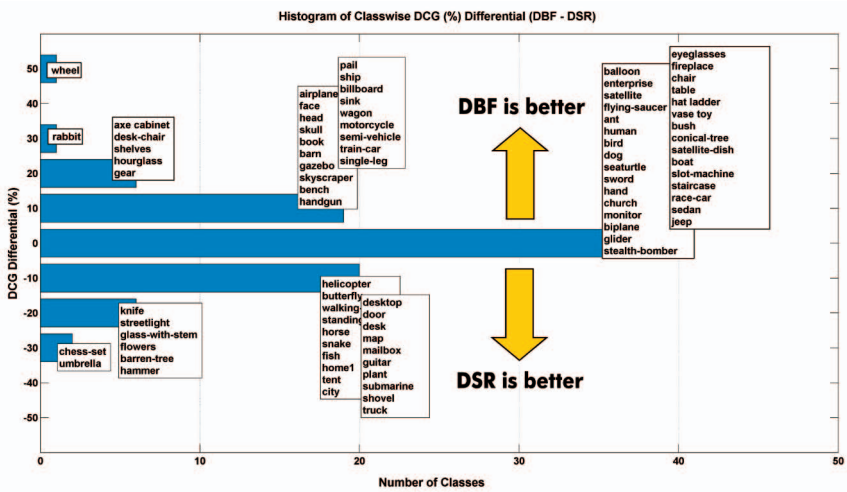

Fig. 13. Histogram of class-wise DCG differentials between DBF and DSR ($\mathrm{DCG}_{\mathrm{DBF}} - \mathrm{DCG}_{\mathrm{DSR}}$).

5.5.2 DBF versus DSR: A Closer Look over All Databases

Now, we carry out a more detailed comparison of DBF with DSR, our nearest competitor, over all databases. As can be seen from Tables 6 and 8, while DSR is better than DBF on PSB by 0.6 percent DCG, our method outperforms DSR on the other databases (on SCU by 1.7 percent, on SHREC by 3.5 percent, on ESB by 1.7 percent). Precision-recall curves in Fig. 12 show that 1) on PSB, performances are more or less equivalent for $Recall < 0.5$; 2) on SCU, a noticeable difference in favor of DBF occurs after $Recall \approx 0.4$, and 3) on SHREC-W and ESB, there are clear performance gaps in favor of DBF for all recall values.

As depicted in Fig. 13, the distribution of class-wise DCG differences between DBF and DSR ($\mathrm{DCG}_{\mathrm{DBF}} - \mathrm{DCG}_{\mathrm{DSR}}$) is symmetric, indicating that, in half of the PSB shape categories, DBF is better than DSR and vice versa (category names can be seen from Fig. 13). This suggests that DBF and DSR are of complementary nature performance-wise; hence, a combination of these two powerful descriptors can be beneficial to further improve the performance on PSB. Table 8 and Fig. 12 show the effect of summing the similarity scores of DBF with those of DSR, in which case we obtain a DCG of 70.2 percent on PSB, a 4 percent gain over DBF or DSR alone. For other databases, however, the improvement remains rather limited. Fig. 14 illustrates the benefits of the combination on a sample query from PSB.

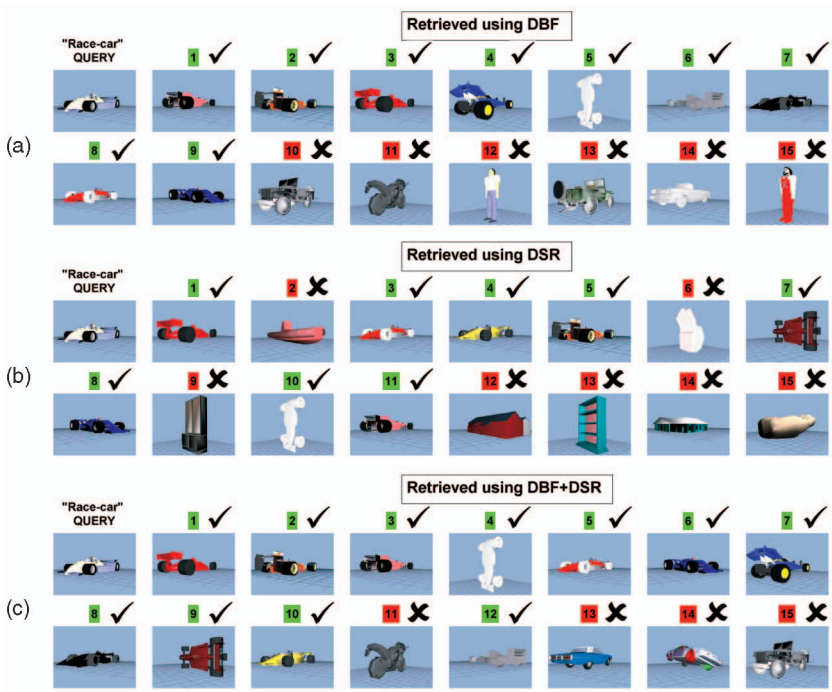

Fig. 14. A "race-car" query retrieved using DBF (a), using DSR (b). and (c) using the combination DBF+DSR.

6 Conclusion

In this work, we have provided a detailed analysis of the density-based shape descriptors for 3D model retrieval. Our framework decisively outperforms its histogram-based ancestors and is also placed in the top league of 3D shape descriptors as proven by extensive retrieval experiments on several 3D model databases with varying mesh quality, semantic content, and classification granularity. For instance, on PSB database, DBF has the DCG value of 65.9 percent, which is approximately as good as the best methods reported so far and 8.2 percent better than the closest histogram-based 3DHT descriptor. Our main conclusion is that for distribution-based descriptors to be effective in 3D model retrieval, they should rely on exhaustive local characterizations via multivariate surface features and they should employ kernel smoothing for pdf estimation. A serendipitous observation concerns the complementariness of two highly discriminative methods: our DBF approach and the hybrid DSR descriptor (66.5 percent DCG on PSB). After combining these two descriptors in a totally unsupervised manner, we have obtained 70.2 percent DCG on PSB. To our knowledge, this is the best retrieval performance on PSB reported so far. Our research in this domain continues with supervised similarity learning schemes adapted to the retrieval problem as in [43].

We have also shown that the pdf structure enjoys a permutation property which can be used to guarantee invariance against orthogonal transformations at the shape matching stage. Even if, for the purpose of computational efficiency, the set of orthogonal transformations is restricted to coordinate axis relabelings and mirror reflections, the proposed matching scheme yields better retrieval performance than merely using L^1-distance. A potential research direction is toward developing algorithms which make use of this permutation property of pdf-based descriptors for correspondence-free 3D shape alignment.

Acknowledgments

The authors would like to thank the anonymous reviewers whose insightful comments and suggestions helped improve this paper significantly. This research was supported by BU Project 03A203, TUBITAK Project 103E038, and TUBITAK Project 107E001. The authors dedicate this paper to the memory of their friend and colleague Francis Schmitt, one of the authors of the paper, whom they unfortunately lost before the appearance of this paper. This research was carried out during C.B. Akgül's PhD studies at Boğaziçi University, Istanbul, Turkey, and Télécom ParisTech, Paris, France.

References

[1] S. Jayanti, K. Kalyanaraman, N. Iyer, and K. Ramani, "Developing an Engineering Shape Benchmark for CAD Models," *Computer-Aided Design,* vol. 38, no. 9, pp. 939-953, Sept. 2006.

[2] 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," *Proc. Third Int'l Conf. Image and Video Retrieval,* pp. 638-646, 2004.

[3] P. Daras, D. Zarpalas, A. Axenopoulos, D. Tzovaras, and M.G. Strintzis, "Three-Dimensional Shape-Structure Comparison Method for Protein Classification," *IEEE/ACM Trans. Computational Biology and Bioinformatics,* vol. 3, no. 3, pp. 193-207, July-Sept. 2006.

[4] B. Bustos, D.A. Keim, D. Saupe, T. Schreck, and D.V. Vranic, "Feature-Based Similarity Search in 3D Object Databases," *ACM Computing Surveys,* vol. 37, no. 4, pp. 345-387, 2005.

[5] *Real-Time 3D Models,* http://www.3drt.com/, 2009.

[6] N. Iyer, S. Jayanti, K. Lou, Y. Kalyanaraman, and K. Ramani, "Three-Dimensional Shape Searching: State-of-the-Art Review and Future Trends," *Computer-Aided Design,* vol. 37, no. 5, pp. 509-530, Apr. 2005.

[7] J.W.H. Tangelder and R.C. Veltkamp, "A Survey of Content Based 3D Shape Retrieval Methods," *Multimedia Tools and Applications,* 2008.

[8] C.B. Akgül, B. Sankur, Y. Yemez, and F. Schmitt, "Density-Based 3D Shape Descriptors," *EURASIP J. Advances in Signal Processing,* vol. 2007, Article ID 32,503, p.16, 2007, doi: 10.1155/2007/32503.

[9] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin, "Shape Distributions," *ACM Trans. Graphics,* vol. 21, no. 4, pp. 807-832, 2002.

[10] Y. Liu, H. Zha, and H. Qin, "The Generalized Shape Distributions for Shape Matching and Analysis," *Proc. IEEE Int'l Conf. Shape Modeling and Applications,* June 2006.

[11] E. Paquet and M. Rioux, "Nefertiti: A Query by Content Software for Three-Dimensional Models Databases Management," *Proc. Int'l Conf. Recent Advances in 3D Digital Imaging and Modeling,* p. 345, 1997.

[12] M. Ankerst, G. Kastenmüller, H.-P. Kriegel, and T. Seidl, "3D Shape Histograms for Similarity Search and Classification in Spatial Databases," *Proc. Sixth Int'l Symp. Advances in Spatial Databases,* pp. 207-226, 1999.

[13] B.K.P. Horn, "Extended Gaussian Images," *Proc. IEEE,* vol. 72, pp. 1671-1686, 1984.

[14] S.B. Kang and K. Ikeuchi, "The Complex EGI: A New Representation for 3D Pose Determination," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 15, no. 7, pp. 707-721, July 1993.

[15] T. Zaharia and F. Prteux, "Shape-Based Retrieval of 3D Mesh Models," *Proc. IEEE Int'l Conf. Multimedia and Expo,* Aug. 2002.

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

[17] S. Belongie, J. Malik, and J. Puzicha, "Shape Matching and Object Recognition Using Shape Contexts," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 24, no. 4, pp. 509-522, Apr. 2002.

[18] A. Johnson and M. Hebert, "Using Spin Images for Efficient Object Recognition in Cluttered 3D Scenes," *IEEE Trans. Pattern Analysis and Machine Intelligence* vol. 21, no. 5, pp. 433-449, May 1999.

[19] C.B. Akgül, B. Sankur, Y. Yemez, and F. Schmitt, "Multivariate Density-Based 3D Shape Descriptors," *Proc. Shape Modeling Int'l (SMI '07),* June 2007.

[20] L. Greengard and J. Strain, "The Fast Gauss Transform," *SIAM J. Scientific and Statistical Computing,* vol. 12, pp. 79-94, 1991.

[21] C. Yang, R. Duraiswami, N.A. Gumerov, and L. Davis, "Improved Fast Gauss Transform and Efficient Kernel Density Estimation," *Proc. Int'l Conf. Computer Vision,* vol. 1, p. 464, 2003.

[22] D.-Y. Chen, X.-P. Tian, Y.-T. Shen, and M. Ouhyoung, "On Visual Similarity Based 3D Model Retrieval," *Computer Graphics Forum,* vol. 22, pp. 223-232, Sept. 2003.

[23] P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser, "The Princeton Shape Benchmark," *Proc. Shape Modeling Int'l,* pp. 167-178, 2004.

[24] D.V. Vranić, "3D Model Retrieval," PhD dissertation, Univ. of Leipzig, 2004.

[25] D.V. Vranić, *Tools for 3D Model Retrieval,* 2005. http://merkur01.inf.uni-konstanz.de/3Dtools/.

[26] T. Funkhouser and P. Shilane, "Partial Matching of 3D Shapes with Priority-Driven Search," *Proc. Symp. Geometry Processing,* June 2006.

[27] H. Dutağaci, B. Sankur, and Y. Yemez, "Transform-Based Methods for Indexing and Retrieval of 3D Objects," *Proc. Fifth Int'l Conf. 3D Digital Imaging and Modeling,* June 2005.

[28] J. Ricard, D. Coeurjolly, and A. Baskurt, "Generalizations of Angular Radial Transform for 2D and 3D Shape Retrieval," *Pattern Recognition Letters,* vol. 26, no. 14, pp. 2174-2186, 2005.

[29] P. Daras, D. Zarpalas, D. Tzovaras, and M.G. Strintzis, "Shape Matching Using the 3D Radon Transform," *Proc. Second Int'l Symp. 3D Data Processing, Visualization, and Transmission,* pp. 953-960, 2004.

[30] D. Zarpalas, P. Daras, A. Axenopoulos, D. Tzovaras, and M.G. Strintzis, "3D Model Search and Retrieval Using the Spherical Trace Transform," *EURASIP J. Advances in Signal Processing*, vol. 2007, Article ID 23 912, p.14, 2007, doi:10.1155/2007/23912.
[31] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz, "Rotation Invariant Spherical Harmonic Representation of 3D Shape Descriptors," *Proc. 2003 Eurographics/ACM SIGGRAPH Symp. Geometry Processing*, pp. 156-164, 2003.
[32] D.V. Vranić, "An Improvement of Rotation Invariant 3D Shape Descriptor Based on Functions on Concentric Spheres," *Proc. IEEE Int'l Conf. Image Processing*, pp. 757-760, Sept. 2003.
[33] S.P.P. Papadakis, I. Pratikakis, and T. Theoharis, "Efficient 3D Shape Matching and Retrieval Using a Concrete Radialized Spherical Projection Representation," *Pattern Recognition*, vol. 40, no. 9, pp. 2437-2452, 2007.
[34] H. Laga, H. Takahashi, and M. Nakajima, "Spherical Wavelet Descriptors for Content-Based 3D Model Retrieval" *Proc. IEEE Int'l Conf. Shape Modeling and Applications*, pp. 15-25, 2006.
[35] M. Hilaga, Y. Shinagawa, T. Kohmura, and T.L. Kunii, "Topology Matching for Fully Automatic Similarity Estimation of 3D Shapes," *Proc. ACM SIGGRAPH*, pp. 203-212, Aug. 2001.
[36] T. Tung and F. Schmitt, "The Augmented Multiresolution Reeb Graph Approach for Content-Based Retrieval of 3D Shapes," *Int'l J. Shape Modeling*, vol. 11, no. 1, June 2005.
[37] H. Sundar, D. Silver, N. Gagvani, and S. Dickinson, "Skeleton Based Shape Matching and Retrieval," *Proc. Shape Modeling Int'l*, p. 130, 2003.
[38] G. Taubin, "Estimating the Tensor of Curvature of a Surface from a Polyhedral Approximation," *Proc. Fifth Int'l Conf. Computer Vision*, p. 902, 1995.
[39] M.P. do Carmo, *Differential Geometry of Curves and Surfaces.* Prentice-Hall, 1976.
[40] C. Dorai and A.K. Jain, "COSMOS—A Representation Scheme for 3D Free-Form Objects," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 10, pp. 1115-1130, Oct. 1997.
[41] W. Härdle, M. Müller, S. Sperlich, and A. Werwatz, *Nonparametric and Semiparametric Models.* Springer, 2004.
[42] D. Giorgi, S. Biasotti, and L. Paraboschi, "Shape Retrieval Contest 2007: Watertight Models Track," R.C. Veltkamp and F.B. ter Haar, eds., *SHREC 2007: 3D Shape Retrieval Contest, Technical Report UU-CS-2007-015*, pp. 5-10, June 2007.
[43] C.B. Akgül, B. Sankur, Y. Yemez, and F. Schmitt, "Similarity Score Fusion by Ranking Risk Minimization for 3D Object Retrieval," *Proc. Eurographics Workshop 3D Object Retrieval*, Apr. 2008.
[44] R.O. Duda, P.E. Hart, and D.G. Stork, *Pattern Classification.* Wiley Interscience, 2000.

Ceyhun Burak Akgül received the BS and MS degrees in electrical engineering from Boğaziçi University, Istanbul, in 2002 and 2004, respectively, and the PhD degree from both Boğaziçi University, Istanbul, and Télécom ParisTech (École Nationale Supérieure des Télécommunications, Paris) in 2007. His research is focused on content-based retrieval, image and shape description, and machine learning. He is the recipient of the Marie Curie Research Scientist position at the Video Processing Analysis Group of Philips Research Europe, Eindhoven. He currently works on the development of a medical diagnostic support system within the framework of the EU-funded IRonDB Project. He is a student member of the IEEE.

Bülent Sankur received the BS degree in electrical engineering from Robert College, Istanbul, and the MSc and PhD degrees from Rensselaer Polytechnic Institute, New York. He is presently at Boğaziçi (Bosphorus) University in the Department of Electrical-Electronic Engineering. His research interests are in the areas of digital signal processing, image and video compression, biometry, cognition, and multimedia systems. He is the founder and leader of the Image and Signal Processing Laboratory at Boğaziçi, and has been serving in various industrial consulting tasks. He has held visiting positions at the University of Ottawa, Technical University of Delft, and École Nationale Supérieure des Télécommunications, Paris. He was the chairman of the International Conference on Telecommunications and of the European Conference on Signal Processing, as well as technical chairman of ICASSP '00. He is a senior member of the IEEE.

Yücel Yemez received the BS degree from Middle East Technical University, Ankara, Turkey, in 1989, and the MS and PhD degrees from Boğaziçi University, Istanbul, Turkey, in 1992 and 1997, respectively, all in electrical engineering. From 1997 to 2000, he was a postdoctoral researcher in the Image and Signal Processing Department of Télécom Paris (École Nationale Supérieure des Télécommunications). Currently, he is an assistant professor in the Computer Engineering Department at Koç University, Istanbul, Turkey. His current research is focused on various fields of computer vision and graphics. He is a member of the IEEE.

Francis Schmitt received the engineering diploma from the École Centrale, Lyon, France, in 1973, and the PhD degree from from the University of Paris VI (Pierre et Marie Curie) in 1979. From 1973 up to his sudden death in October 2008, he was a member of the faculty at Télécom Paris (École Nationale Supérieure des Télécommunications), last holding the rank of full professor in the Image and Signal Processing Department in which he headed the Image Processing Group. His main interests were in computer vision, 3D modeling, image and 3D object indexing, computational geometry, multispectral imagery, and colorimetry. He authored or coauthored nearly 150 publications in these fields.