Hindawi Publishing Corporation
EURASIP Journal on Advances in Signal Processing
Volume 2007, Article ID 32503, 16 pages
doi:10.1155/2007/32503

Research Article

Density-Based 3D Shape Descriptors

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

[1] *Electrical and Electronics Engineering Department, Boğaziçi University, 34342 Bebek, Istanbul, Turkey*
[2] *GET-Telecom Paris, CNRS UMR 5141, 75634 Paris Cedex 13, France*
[3] *Computer Engineering Department, Koç University, 34450 Sariyer, Istanbul, Turkey*

Received 1 February 2006; Revised 14 July 2006; Accepted 10 September 2006

Recommended by Petros Daras

We propose a novel probabilistic framework for the extraction of density-based 3D shape descriptors using kernel density estimation. Our descriptors are derived from the probability density functions (pdf) of local surface features characterizing the 3D object geometry. Assuming that the shape of the 3D object is represented as a mesh consisting of triangles with arbitrary size and shape, we provide efficient means to approximate the moments of geometric features on a triangle basis. Our framework produces a number of 3D shape descriptors that prove to be quite discriminative in retrieval applications. We test our descriptors and compare them with several other histogram-based methods on two 3D model databases, Princeton Shape Benchmark and Sculpteur, which are fundamentally different in semantic content and mesh quality. Experimental results show that our methodology not only improves the performance of existing descriptors, but also provides a rigorous framework to advance and to test new ones.

1. INTRODUCTION

The use of 3D models is becoming increasingly more commonplace with their distribution on the Internet and with the availability of 3D scanners. Many fields are focused on 3D object models: computer graphics, computer-aided design, medical imaging, molecular analysis, cultural heritage in virtual environments, movie industry, military target detection, or industrial quality control to name a few. Efficient organization and access to these databases demand effective tools for indexing, categorization, classification, and representation of 3D objects. All these database activities hinge on the development of 3D object similarity measures. There are two paradigms for 3D object database operations and design of similarity measures, namely, the feature vector approach and the nonfeature vector approach [1, 2]. The feature vector paradigm aims at obtaining numerical values of certain shape descriptors and measuring the distances between these vectors. A typical example of nonfeature-based approach is to describe the object as a graph and then use graph similarity metrics. In this work, we follow the feature vector paradigm, and furthermore we limit our scope to the subclass of histogram-based descriptors.

Representations used for shape matching are often referred to as 3D shape descriptors and they usually differ substantially from those intended for 3D object rendering and visualization [3]. Shape descriptors aim at encoding geometrical and topological properties of an object in a discriminative and compact manner. The diversity of shape descriptors range from 3D moments to shape distributions, from spherical harmonics to ray-based sampling, from point clouds to voxelized volume transforms [1, 2, 4–7]. In this work, inspired from histogram-based 3D shape descriptors [8–12], we propose a density-based approach that applies to local geometrical features of arbitrary dimension. Our interest in histogram-based 3D shape descriptors stems from their generality and their simplicity. They are global descriptors based on sets of local measurements and they have been shown to be effective in classifying shapes into broad categories [2]. Our objective is to show that, in addition to their categorization capability, they have also satisfactory retrieval performance.

Any histogram-based 3D shape descriptor must face the problem of estimating the histogram from any given mesh composed of triangles usually with arbitrary forms and sizes. In the previous histogram-based approaches, the surface samples are either chosen as the centers of gravity of the triangles or obtained by randomly sampling several points from the surface. A single sample from each triangle may not adequately represent the mesh. The random sampling of the

surface may compensate for the nonuniform distribution of triangles, provided that a sizeable number of surface points is taken. Although the random sampling approach proves to be useful for computing histograms of scalar features [10], it is not practical in the multidimensional case due to the curse of dimensionality: the number of samples required to fill in the multivariate histogram bins increases exponentially as dimensionality increases [13], resulting in a significant extra computational load which is not affordable for most applications such as retrieval.

Our density-based framework makes a more effective use of each triangle and also takes care of the nonuniformity of their areas and orientations without resorting to expensive random sampling. First, we do not use samples but exploit the information in the whole triangle area using an integration scheme, as described in Section 3.3. Second, we resort to nonparametric kernel density estimation (KDE) with rule-based bandwidth parameter assignment [13, 14]. In other words, local geometric information emanating from each mesh triangle contributes to the geometric feature density by the intermediary of a kernel. Thus local evidences about surface shape are accumulated at targeted density points to result in a global shape description. Third, we use a Gaussian kernel. Since the Gaussian density is completely determined by its first two moments, we only need to estimate the mean and the variance of the feature for each triangle. For certain cases, these moments can be approximated very accurately by making use of the geometry of a triangle in 3D space. The choice of Gaussian kernel brings in the additional advantage of alleviating the computational burden of calculating large sums of Gaussians, as occur in the proposed set of descriptors, by enabling the use of the efficient fast Gauss transform (FGT) [15, 16]. Thus the main contribution of our work is to propose an analytical framework for the extraction of 3D descriptors from local surface features that characterize the object geometry. This framework computes probability densities of local features instead of their conventional histograms. Here, we interpret histograms and densities in a broad sense: any descriptor that uses an accumulator scheme of measured quantities qualifies as a histogram-based descriptor. As a byproduct, we also introduce some novel local features.

The rest of the paper is structured as follows. In Section 2, we provide an overview of histogram-based 3D shape descriptors. Section 3 introduces the local geometric features we have considered and describes the KDE-based computational framework. In Section 4, we illustrate the retrieval performance of our method in comparison to other equivalent or similar histogram-based descriptors [8–12]. In Section 5, we draw conclusions and discuss further directions in density-based 3D shape descriptors.

2. PREVIOUS WORK ON 3D SHAPE DESCRIPTORS

There are two main paradigms of 3D shape description, namely, graph-based and vector-based. Graph-based representations are more elaborate and complex, harder to obtain, but represent shape properties in a more faithful and intuitive manner. Shock graphs [17], multiresolution Reeb graphs [6, 18, 19], and skeletal graphs [20] are methods that fall in this category. However, they do not generalize easily and hence they are not very convenient to use in unsupervised learning, for example, to search for natural shape classes in a database. Vector-based representations, on the other hand, are more easily computed. Although they do not necessarily conduce to plausible topological visualizations, they can be naturally employed in both supervised and unsupervised classification tasks. Typical vector-based representations are extended Gaussian images [8, 9], cord and angle histograms [11], 3D shape histograms [21], spherical harmonics [7, 22–24], and shape distributions [10]. In this work, we are exclusively interested in histogram-based 3D shape descriptors that constitute a particular branch of vector-based representations. In the following, we provide a brief overview of histogram-based descriptors. References [1, 2, 4] provide also excellent surveys.

In [11], Paquet and Rioux present *cord and angle histograms* for matching 3D objects. A "cord," which is actually a ray, joins the barycenter of the mesh with a triangle center. The histograms of the length and of the angles of these rays (with respect to a reference frame) are used as the 3D shape descriptors. Although automatic determination of a canonical reference frame for 3D meshes is still not totally solved [7], the common practice is to obtain the eigendecomposition of the covariance matrix of the surface points. The covariance matrix itself can be computed using the mesh vertices, the triangle centers, or in a "continuous" way as described in [7]. The resulting eigenvectors, which are the orthogonal directions along which the mesh has maximal spread, are taken as a reference frame. Notice that the eigendirections may not necessarily correspond to the "natural" pose of the object; however, they can serve as a canonical reference frame. In conclusion, Paquet and Rioux [11] consider the shape descriptors consisting of the ray length and the relative ray angles with respect to the largest two eigenvectors. One shortcoming of all such approaches that reduce the triangles to their center points is that they do not take into consideration the size and shape of the mesh triangles. First, because triangles of any size have equal weight in the final shape distribution; second, because the triangle shapes can be arbitrary, so that the center may not represent adequately the impact of the triangle on the shape distribution.

In the *shape distributions* approach, Osada et al. [10] use a collection of shape functions, which are geometrical quantities estimated by a random sampling of the surface of the 3D object. Their shape functions are defined as the distance of surface points to the center of mass of the model (D1), the distance between two surface points (D2), the area of the triangle defined by three surface points (D3), the volume of the tetrahedron defined by four surface points (D4), and so on. The descriptors of the object are then defined as the histograms of these shape functions. The randomization of the surface sampling process improves the estimation over Paquet and Rioux's approach [11], since a more representative and dense set of surface points is used. Obviously, the histogram accuracy can be controlled with the sample size.

Table 1: Invariance properties of histogram-based 3D shape descriptors.

Descriptor	Translation invariance	Rotation invariance	Scale invariance
Cord histogram [11]	No	Yes	No
Angle histogram [11]	No	No	Yes
D1-*distribution* [10]	No	Yes	No
D2-*distribution* [10]	Yes	Yes	No
Shape histogram (shells) [21]	No	Yes	No
Shape histogram (sectors) [21]	No	No	No
EGI [8]	Yes	No	No
CEGI [9]	No	No	No
3DHT [12]	No	No	No

Ankerst et al. use *shape histograms* for the purpose of molecular surface analysis [21]. A shape histogram is defined by partitioning the 3D space into concentric shells and sectors around the center of mass of a 3D model. The histogram is constructed by accumulating the surface points in the bins (in the form of shells, sectors, or both) based on a nearest-neighbor rule. Ankerst et al. [21] illustrate the shortcomings of Euclidean distance to compare two shape histograms and make use of a Mahalanobis-like quadratic distance measure taking into account the distances between histogram bins.

Extended Gaussian images (EGI), introduced by Horn [8], form another class of histogram-based 3D shape descriptors. An EGI consists of a spherical histogram with bins indexed by (θ_j, φ_k), where each bin corresponds to some quantum of the spherical azimuth and elevation angles (θ, φ) in the range $0 \leq \theta < 2\pi$ and $0 \leq \varphi < \pi$. The histogram bins accumulate the count of the spherical angles of the surface normal per triangle, usually weighted by the triangle area. Kang and Ikeuchi have extended the EGI approach by considering the normal distances of the triangles to the origin [9]. Accordingly, each histogram bin accumulates a complex number whose magnitude and phase are the area of the triangle and its signed distance to the origin, respectively. The resulting 3D shape descriptor is called complex extended Gaussian images (CEGI) [9].

In [12], Zaharia and Prêteux present the *3D Hough transform descriptor* (3DHT) as a histogram constructed by accumulating surface points over planes in 3D space. Each triangle of the mesh contributes to each plane with a weight equal to the projected area of the triangle on the plane but only if the scalar product between their normals is higher than a given threshold. Although we have not encountered in the literature a direct comparison between 3DHT and EGI, 3DHT can be considered as a generalized version of EGI, where concentric spherical shells of different radii are constructed around the object's center of mass. One can consequently conjecture that the 3DHT descriptor captures the shape information better than the EGI descriptor, as will be shown experimentally in Section 4.

An important property of a 3D shape descriptor is its invariance to similarity transformations, that is, translation (T), rotation (R), and scale (S) [1, 2, 4, 7]. In Table 1, we summarize invariance properties of the histogram-based shape descriptors discussed above.

3. THE PROPOSED FRAMEWORK FOR DENSITY-BASED DESCRIPTORS

3.1. Local geometric features

We assume that each 3D shape is represented as a triangular mesh and that its center of mass coincides with the origin of the coordinate system. In what follows, capital italic letter P stands for a point in 3D, a small case boldface letter $\mathbf{p} = (p_x, p_y, p_z)$ for its vector representation, $\hat{\mathbf{n}}_P = (\hat{n}_{P,x}, \hat{n}_{P,y}, \hat{n}_{P,z})$ for the unit surface normal vector at P when P belongs to a surface $\mathcal{M} \subset \mathbb{R}^3$, and $\langle \cdot, \cdot \rangle$ for the usual dot product.

We define a local geometric feature as a mapping S from the points of a surface $\mathcal{M} \subset \mathbb{R}^3$ into a d-dimensional space, generally a subspace of $\mathbb{R}^d$. Each dimension of this space corresponds to a specific geometric property that can be calculated at each point of the surface. For example, the distance of a surface point to the center of the 3D shape is a one-dimensional ($d = 1$) geometric feature, while the mesh triangle normal $\hat{\mathbf{n}}_P$ is a three-dimensional feature vector ($d = 3$). In this work, we consider three different multidimensional local geometric features that we describe in the sequel.

The radial feature S_r at a point P is a 4-tuple defined as

$$S_r(P) \triangleq (r_P, \hat{r}_{P,x}, \hat{r}_{P,y}, \hat{r}_{P,z}) \quad \text{with} \quad \begin{aligned} r_P &\triangleq \sqrt{p_x^2 + p_y^2 + p_z^2}, \\ \hat{r}_{P,x} &\triangleq \frac{p_x}{r_P}, \\ \hat{r}_{P,y} &\triangleq \frac{p_y}{r_P}, \\ \hat{r}_{P,z} &\triangleq \frac{p_z}{r_P}. \end{aligned} \tag{1}$$

Accordingly, S_r consists of a magnitude component r_P measuring the distance of the point P to the origin, and a direction component $\hat{\mathbf{r}}_P \triangleq (\hat{r}_{P,x}, \hat{r}_{P,y}, \hat{r}_{P,z})$ that gives the orientation of the point P (see Figure 1). Observe that we can write S_r also as $S_r(P) = (r_P, \hat{\mathbf{r}}_P)$. The direction component $\hat{\mathbf{r}}_P$ is a three-dimensional vector with unit norm; hence it lies on the unit sphere.

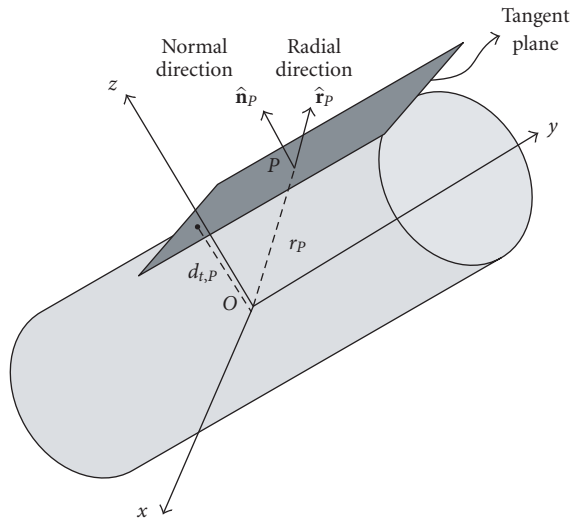


Figure 1: Radial and normal directions of a surface point.

The tangent plane-based feature S_t at a point P is a 4-tuple defined as

$$S_t(P) = (d_{t,P}, \hat{n}_{P,x}, \hat{n}_{P,y}, \hat{n}_{P,z}) \quad \text{with } d_{t,P} \triangleq r_P \left| \langle \hat{\mathbf{r}}_P, \hat{\mathbf{n}}_P \rangle \right|. \tag{2}$$

Similar to the S_r feature, S_t has a magnitude component $d_{t,P}$, which stands for the distance of the tangent plane at P to the origin, and a direction component $\hat{\mathbf{n}}_P = (\hat{n}_{P,x}, \hat{n}_{P,y}, \hat{n}_{P,z})$ (see Figure 1). Thus, we may write $S_t(P) = (d_{t,P}, \hat{\mathbf{n}}_P)$. The normal $\hat{\mathbf{n}}_P$ is a unit norm vector by definition and lies on the unit sphere.

The cross-product feature S_c aims at encoding the relationship between the former two features, namely, the radial feature S_r and the tangent plane-based feature S_t. To this end, we define S_c at a point P as

$$S_c(P) \triangleq (r_P, c_{P,x}, c_{P,y}, c_{P,z}) \triangleq (r_P, \mathbf{c}_P) \quad \text{with } \mathbf{c}_P \triangleq \hat{\mathbf{r}}_P \times \hat{\mathbf{n}}_P. \tag{3}$$

In much the same way as in S_r and S_t, S_c is decoupled into a magnitude component r_P and a direction component $\mathbf{c}_P$. Notice, however, that $\mathbf{c}_P$ is not a unit-norm vector unless the angle between the radial direction $\hat{\mathbf{r}}_P$ and the normal direction $\hat{\mathbf{n}}_P$ is $\pi/2$. Both $\hat{\mathbf{r}}_P$ and $\hat{\mathbf{n}}_P$ being unit norm vectors, the norm of $\mathbf{c}_P$ is lower than or equal to unity and it lies inside the unit ball.

The local geometric features presented above and their invariance properties are summarized in Table 2.

3.2. Kernel density estimation

Given a set of observations $\{s_k\}_{k=1}^K$ for a random variable (scalar or vector) S, the kernel approach to estimate the probability density of S is formulated in its most general form as

$$f_S(s) = \sum_{k=1}^{K} w_k \left| H_k \right|^{-1} \mathcal{K}\left(H_k^{-1}(s - s_k)\right), \tag{4}$$

Table 2: Local geometric features and their invariance properties (assuming that the barycenter of the surface $\mathcal{M}$ is at the origin).

Feature	Component-wise invariance	Overall invariance
Radial S_r	Magnitude r_P: rotation Direction $\hat{\mathbf{r}}_P$: scale	None
Tangent plane S_t	Magnitude $d_{t,P}$: rotation Direction $\hat{\mathbf{n}}_P$: scale	None
Cross-product S_c	Magnitude r_P: rotation Direction $\mathbf{c}_P$: scale	None

where $\mathcal{K} : \mathbb{R}^d \to \mathbb{R}$ is a kernel function, H_k is a $d \times d$ matrix composed of a set of design parameters called *bandwidth* parameters (smoothing parameters or scale parameters) for the kth observation, and w_k is the importance weight associated with the kth observation. The contribution of each data point s_k to the density function $f_S(s)$ at a target point s is computed through the kernel function $\mathcal{K}$ scaled by the matrix H_k and the weight w_k. Thus KDE involves a data set $\{s_k\}_{k=1}^K$ with the associated set of importance weights $\{w_k\}_{k=1}^K$, the choice of a kernel function $\mathcal{K}$ and the setting of bandwidth parameters $\{H_k\}_{k=1}^K$.

We compute the probability density values of a certain local geometric feature S from a set of observations $\{s_k\}_{k=1}^K$. We assume that the 3D shape is represented as a triangular mesh consisting of K triangles. Thus we can obtain an observation s_k from each of the triangles in the mesh, as will be explained in Section 3.3. Since, in general, the mesh is made up of nonuniformly sized triangles, the data should be weighted accordingly. A natural choice for the importance weight w_k of a data point s_k is the ratio of the kth triangle area to the total surface area, yielding $\sum_{k=1}^K w_k = 1$. It is known that the particular functional form of the kernel does not significantly affect the accuracy of the estimator [14]. The Gaussian kernel has become a popular choice, first because it lends itself more easily to asymptotic error analysis [14]; and second, for the existence of efficient algorithms to calculate large sums of Gaussians, as the fast Gauss transform (FGT) already mentioned in the introduction [15, 16]. Actually, FGT is the dominant reason why we choose the Gaussian kernel since computational efficiency is an important requirement for 3D object retrieval [1, 2] (see Section 3.6 for details).

The setting of the bandwidth parameters $\{H_k\}_{k=1}^K$ is critical for an accurate kernel density estimation [14, 25]. For the Gaussian kernel, the bandwidth matrix H_k simply corresponds to the feature covariance matrix. For setting/estimating the bandwidth parameters, there exist several guidelines and computational methods with varying complexity [14, 25]. We discuss different alternatives in Section 3.4. The probability density function $f_S(s)$, when computed over predefined target points using (4), results in the shape descriptor sought for a given triangular mesh. The methodology that we employ to choose the target points for each specific feature is explained in Section 3.5.

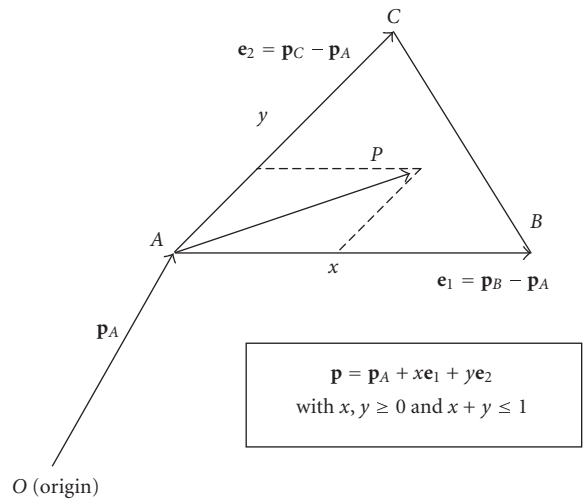


Figure 2: A local basis for a triangle in 3D.

3.3. Feature calculation

Given a d-dimensional local feature $S = (S_1, \ldots, S_d)$, the observation s_k can be obtained from the mesh triangle T_k by evaluating the value of S at the barycenter of the triangle. However, the mesh triangles having in general arbitrary shapes, the feature value at the barycenter may not be the most representative one. The shape of the triangle should be in some way taken into account in order to reflect the local feature characteristics more faithfully. The expected value of the local feature $E\{S \mid T\}$ over the triangle T is more informative than the feature value only sampled at a single point, the barycenter of the triangle.

Consider T as an arbitrary triangle in 3D space with vertices A, B, and C represented by $\mathbf{p_A}$, $\mathbf{p_B}$, and $\mathbf{p_C}$, respectively, (see Figure 2). By noting $\mathbf{e}_1 = \mathbf{p}_B - \mathbf{p}_A$ and $\mathbf{e}_2 = \mathbf{p}_C - \mathbf{p}_A$, we can obtain a parametric representation for a point P inside the triangle T as $\mathbf{p} = \mathbf{p}_A + x\mathbf{e}_1 + y\mathbf{e}_2$, where the two parameters x and y satisfy the constraints $x, y \geq 0$ and $x + y \leq 1$. We assume that the point P is uniformly distributed inside the triangle T. Thus, the expected value of the ith component of S, denoted by $E\{S_i \mid T\}$, is given by

$$E\{S_i \mid T\} = \iint_\Omega S_i(x, y) f(x, y) dx\, dy, \quad i = 1, \ldots, d, \quad (5)$$

where $S_i(x, y)$ is the feature value at (x, y) and $f(x, y)$ is the probability density function of the pair (x, y) over the domain $\Omega = \{(x, y) : x, y \geq 0,\ x + y \leq 1\}$. Accordingly, $f(x, y) = 2$ when $(x, y) \in \Omega$ or zero otherwise. The integration is performed over the domain Ω. To approximate (5), we apply Simpson's 1/3 numerical integration formula [26]. We avoid the arbitrariness in vertex labeling by considering the three permutations of the labels A, B, and C. This yields us three approximations, which are in turn averaged to yield

$$\begin{aligned} E\{S_i \mid T\} \approx{}& \left(\frac{1}{27}\right)(S_i(\mathbf{p}_A) + S_i(\mathbf{p}_B) + S_i(\mathbf{p}_C)) \\ &+ \left(\frac{4}{27}\right)\left(S_i\left(\frac{\mathbf{p}_A + \mathbf{p}_B}{2}\right) + S_i\left(\frac{\mathbf{p}_A + \mathbf{p}_C}{2}\right)\right. \\ &\left. + S_i\left(\frac{\mathbf{p}_B + \mathbf{p}_C}{2}\right)\right) \\ &+ \left(\frac{4}{27}\right)\left(S_i\left(\frac{2\mathbf{p}_A + \mathbf{p}_B + \mathbf{p}_C}{4}\right)\right. \\ &+ S_i\left(\frac{\mathbf{p}_A + 2\mathbf{p}_B + \mathbf{p}_C}{4}\right) \\ &\left. + S_i\left(\frac{\mathbf{p}_A + \mathbf{p}_B + 2\mathbf{p}_C}{2}\right)\right). \end{aligned} \quad (6)$$

Equation (6) boils down to take a weighted average of feature values calculated at 9 points on the triangle.

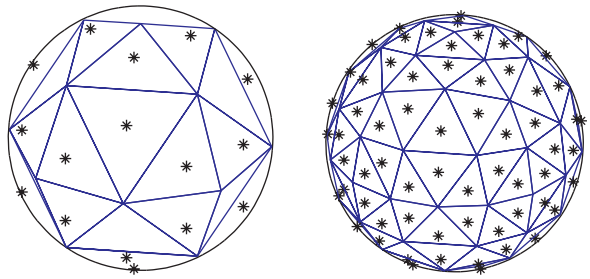
3.4. Bandwidth selection

There are three levels of analysis at which the parameters in the bandwidth matrix H_k involved in KDE can be chosen (see (4) in Section 3.2).

(1) *Triangle level*: this option allows a distinct bandwidth parameter for each triangle in the mesh. In principle, this choice is very flexible since it does not make any assumptions about the shape of the kernel function and hence about the shape of the kth triangle. In general, finding a KDE bandwidth matrix specific to each observation is a difficult problem [25]. For the Gaussian kernel, however, estimation of the bandwidth matrix H_k reduces to the estimation of the feature covariance matrix. The moment formula in (5) and its numerical approximation in (6) can directly be used for moments of any order. For example, the (i, j)th component h_{ij} of H is computed by

$$\begin{aligned} h_{ij} = {}& \iint_\Omega S_i(x, y) S_j(x, y) f(x, y) dx\, dy \\ &- \iint_\Omega S_i(x, y) f(x, y) dx\, dy \\ &\times \iint_\Omega S_j(x, y) f(x, y) dx\, dy, \quad i, j = 1, \ldots, d. \end{aligned} \quad (7)$$

(2) *Mesh level*: the second option is to use a fixed bandwidth matrix for all triangles in a given mesh, but different bandwidths for different meshes. In this case, the bandwidth matrix for a given feature can be obtained from its observations using Scott's rule of thumb [14]: $H_{\text{Scott}} = (\sum_k w_k^2)^{1/(d+4)} \hat{C}^{1/2}$, where d is the dimension of the feature, $\hat{C}$ is the estimate of the feature covariance matrix, and w_k is the weight associated to each observation. Scott's rule of thumb is proven to provide the optimal bandwidth in terms of estimation error when the kernel function and the unknown density are both Gaussian. Although, there is no guarantee that feature distributions to be Gaussian, Scott's rule of thumb is still used for its simplicity.



(a) (b)

Figure 3: Distribution of target points over the unit-sphere, obtained by subdividing an octahedron once (left: 32 points) and twice (right: 128 points).

(3) *Database level*: in the last option, the bandwidth parameter is fixed for all triangles and meshes, that is, $H_k = H$. Setting the bandwidth at database level has the implicit effect of smoothing the resulting densities. In this case, we estimate the bandwidth parameters from a representative subset of the database by averaging the Scott bandwidth matrices over the selected meshes.

3.5. Choice of the targets

Targets are defined as the points at which the feature density functions are explicitly calculated. The density values computed at these targets constitute the 3D shape feature vector. Selection of target points must result in parsimonious yet discriminative descriptors. For single-dimensional features, it suffices to uniformly sample the density function within its dynamic range. However, the multidimensional features, S_r, S_t, and S_c, which consist of magnitude and direction components, require more attention. We denote the target size by N_{mag} for the magnitude component and by N_{dir} for the direction component. The target points for these multidimensional features are then obtained by the Cartesian product of the two sets, yielding an overall target set size of $N = N_{\text{mag}} \times N_{\text{dir}}$. The magnitude components of S_r and S_t are uniformly quantized in the interval $[0, r_{\max}]$, while those of S_t in the $[0, d_{t,\max}]$ interval. The setting of $r_{\max}$ and $d_{t,\max}$ is discussed in Section 4.2. The direction components of S_r and S_t features, namely, $\hat{\mathbf{r}}_P$ and $\hat{\mathbf{n}}_P$, lie on the unit sphere. To complete the design of target points, following [12], we consider an octahedron circumscribed by the unit sphere and we subdivide each of its 8 triangles into four, twice, by radially projecting back the subdivided triangles to the surface of the sphere. As targets of the direction components of S_r and S_t, we select the barycenters of the resulting 128 triangles, 16 per each of the 8 faces of the octahedron. This leads to a uniform partitioning of the sphere, as shown in Figure 3.

The S_c feature has a direction component $\mathbf{c}_P$ with non-unit norm, which lies within the unit ball. For the target set of the direction component $\mathbf{c}_P$, we thus similarly consider octahedra, but circumscribed by spheres of various radii. We take four such octahedra within spheres of radial length 0.25, 0.5, 0.75, and 1. We subdivide the two inner octahedra once, each yielding 32 targets, and the two outer octahedra twice, each yielding 128 targets. This gives a total of $N_{\text{dir}} = 320$ regularly spaced targets for the $\mathbf{c}_P$-component of the S_c feature. The inner spheres have sparser targets to balance out the target densities of the outer spheres.

3.6. Computational complexity of KDE

The computational complexity of KDE using directly (4) 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, that is, targets. For applications such as content-based retrieval, the $O(KN)$-complexity is prohibitive. To give an 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 the S_r-descriptor (1024-point pdf) takes 125 seconds. However, when the kernel function in (4) is chosen as Gaussian, we can use the fast Gauss transform (FGT) [15, 16] to reduce the computational complexity by two orders of magnitude. For example, with FGT, the S_r-descriptor computation 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 $O(K + N)$. In our 3D shape description system, we have used an improved version of FGT implemented by Yang et al. [16].

For the sake of completeness, we provide the conceptual guidelines of the FGT algorithm (see [15, 16] for mathematical and implementation details). FGT is a special case of the more general fast multipole method [15], which trades off computational simplicity for acceptable loss of accuracy. The basic idea is to cluster the data points and target points using appropriate data structures and to replace the large sums with smaller ones that are equivalent up to a given precision. In the case of FGT, each exponential in the sum is shifted and expanded into a truncated Hermite series in $O(K)$ operations. The gain in complexity is achieved by avoiding the computation of every Gaussian at every evaluation point unlike the direct approach, which has $O(KN)$-complexity. The accuracy can be controlled by the truncation order. Truncated Hermite series are constructed about a small number of cluster centers formed by target points; the series are shifted to target cluster centers, and then evaluated at N targets in $O(N)$ operations. Since the two sets of operations are disjoint, the total complexity of FGT becomes $O(K + N)$.

3.7. Flow diagram of the algorithm

We summarize below the proposed algorithm to obtain a density-based 3D shape descriptor.

(1) For a chosen local feature S, specify a set of targets t_n, $n = 1, \ldots, N$.
(2) Normalize the 3D triangular mesh $\mathcal{M} = \bigcup_{k=1}^{K} T_k$ according to the invariance requirements of S.
(3) For each mesh triangle T_k, calculate its feature value s_k using (6) and its weight w_k.

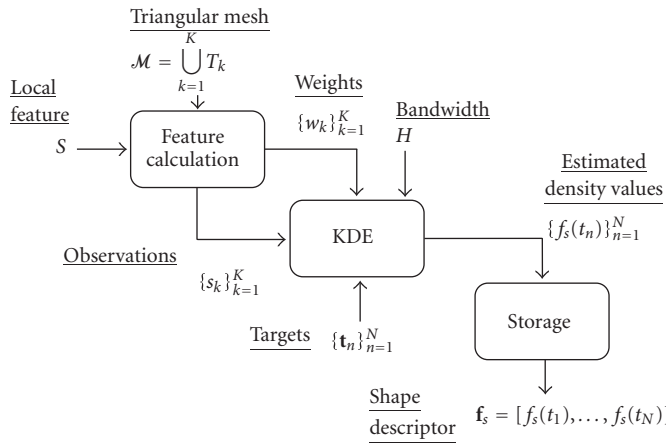


Figure 4: Flow diagram to compute a density-based 3D shape descriptor when the bandwidth is set at database level.

(4) Set the bandwidth parameters H_k according to the strategy chosen among the three options described in Section 3.4.
(5) For each target t_n, $n = 1, \ldots, N$, evaluate the local feature density $f_S(t_n)$, using (4).
(6) Store the resulting density values $f_S(t_n)$ in the shape descriptor $\mathbf{f}_S = [f_S(t_1), \ldots, f_S(t_N)]$.

Note that the descriptors corresponding to L different local features $S_1, \ldots, S_L$ can be concatenated to obtain a combined descriptor $\mathbf{f}_{S_1,\ldots,S_L} = [\mathbf{f}_{S_1}, \ldots, \mathbf{f}_{S_L}]$. Figure 4 depicts the flow diagram of the algorithm when the bandwidth parameters are set at database level. Alternatively, in the triangle or mesh level setting, a bandwidth matrix is to be computed for each triangle or for the entire mesh, respectively. Note that in Figure 4, we assume that the mesh $\mathcal{M}$ has already undergone a pose and/or scale normalization step depending on the missing invariance properties of the local feature S chosen.

4. EXPERIMENTAL RESULTS

In this section, we illustrate the performance of the proposed shape descriptors in 3D retrieval applications. When a query model is presented to the 3D object database, its descriptor is calculated and then compared to all the stored descriptors using a distance function. The outcome is a set of database models sorted in increasing distance. The models at the top of the list are expected to resemble the queried model more than those at the bottom of the list.

We have experimented on two different 3D model databases: the Princeton Shape Benchmark (PSB) [5] and the Sculpteur Database (SCUdb) [6, 27]. Both databases consist of objects described as triangular meshes, though they differ substantially in terms of content and mesh quality. PSB is a publicly available database containing a total of 1814 synthesis models, categorized into general classes such as animals, humans, plants, household objects, tools, vehicles, buildings, and so forth. An important feature of the database is the availability of two equally sized sets. One of them is a training set (90 classes) reserved for tuning the parameters involved in the computation of a particular shape descriptor, and the other for testing purposes (92 classes). By contrast, SCUdb is a private database containing over 800 models corresponding mostly to scanned archeological objects residing in museums [6, 27]. Presently, 513 of the models are classified into 53 categories with comparable set populations, which include utensils of ancient times (e.g., amphorae, vases, bottles, etc.), pavements, and artistic objects such as human statues (parts or as a whole), figurines, and moulds. The database has been augmented by artificially generated 3D objects such as spheres, tori, cubes, or cones in order to build a set of simple well-controlled classes. The meshes in SCUdb are highly detailed and reliable in terms of connectivity and orientation of triangles. To give an idea of the significant differences between PSB and SCUdb, we can quote average mesh resolution figures. The average number of triangles in SCUdb and in PSB is 175250 and 7460, respectively, corresponding to a ratio of 23. In terms of vertices, SCUdb meshes contain 87670 vertices on the average while for PSB this number is 4220. Furthermore, the average triangular area relative to the total mesh area is 33 times smaller in SCUdb than in PSB.

4.1. Evaluation tools

The most commonly used statistics for measuring the performance of a shape descriptor in a content-based retrieval application are summarized below [5].

(i) *Precision-recall curve*

For a query q that is a member of a certain class, *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 C_q:

$$\text{Precision} = \frac{K_q}{K_{\text{ret}}}, \qquad \text{Recall} = \frac{K_q}{C_q}. \tag{8}$$

(ii) *Nearest neighbor (NN)*

The percentage of the first-closest matches that belong to the query class.

(iii) *First-tier and second-tier*

First-tier (FT) is the recall when the number of retrieved models is the same as the size of the query class and second-tier (ST) is the recall when the number of retrieved models is two times the size of the query class.

(iv) *E-measure*

This is a composite measure of the precision and recall for a fixed number of retrieved models, for example, 32, based on the intuition that a user of a search engine is more interested in the first page of query results than in later pages. E-measure is given by

$$\mathrm{E} = \frac{2}{1/\mathrm{precision} + 1/\mathrm{recall}}. \tag{9}$$

(v) *Discounted cumulative gain*

A statistic that weights correct results near the front of the list more than correct results later in the ranked list under the assumption that a user is less likely to consider elements near the end of the list. Specifically, the ranked list of retrieved objects is converted to a list L, where an element L_k has value 1 if the kth object in the ranked list is in the same class as the query and otherwise has value 0. Discounted cumulative gain DCG_k is then defined as

$$\mathrm{DCG}_k = \begin{cases} L_k, & k = 1, \\ \mathrm{DCG}_{k-1} + \dfrac{L_k}{\log_2(k)}, & \text{otherwise.} \end{cases} \tag{10}$$

The final DCG score for a query q is obtained for $k = K_{\max}$, where $K_{\max}$ is the total number of objects in the database, and normalizing $\mathrm{DCG}_{K_{\max}}$ by the maximum possible DCG that would be achieved if the first C_q retrieved elements were in the class of the query q (C_q is the size of the query class). Thus DCG reads as

$$\mathrm{DCG} = \frac{\mathrm{DCG}_{K_{\max}}}{1 + \sum_{k=2}^{C_q} \left(1/\log_2(k)\right)}. \tag{11}$$

(vi) *Normalized DCG*

This is a very useful statistic based on averaging DCG values of a set of algorithms on a particular database. Normalized DCG (NDCG) gives the relative performance of an algorithm with respect to the other ones. A negative value means that the performance of the algorithm is below the average; similarly a positive value indicates above the average performance. Let $\mathrm{DCG}^{(A)}$ be the DCG of a certain algorithm A and let $\mathrm{DCG}^{(\mathrm{avg})}$ be the average DCG values of a series of algorithms on the same database, then NDCG for the algorithm A is defined as

$$\mathrm{NDCG}^{(A)} = \frac{\mathrm{DCG}^{(A)}}{\mathrm{DCG}^{(\mathrm{avg})}} - 1. \tag{12}$$

All these quantities are normalized within the range $[0, 1]$ (except NDCG) and higher values reflect better performance. In order to give the overall performance of a shape descriptor on a database, the values of a statistic for each query are averaged to yield a single performance figure. The retrieval statistics presented in the sequel are obtained using the utility software included in PSB [5].

Table 3: Histogram-based 3D shape descriptors and their sizes.

Descriptor	Acronym	Size N
Cord and angle histograms [11]	CAH	$4 \times 64 = 256$
D1-*distribution* [10]	D1	64
D2-*distribution* [10]	D2	64
EGI [8]	EGI	128
3DHT [12]	3DHT	$8 \times 128 = 1024$

4.2. Retrieval experiments

In all of our retrieval experiments, we use the Minkowski-l_1 distance measure to assess the similarity between descriptors since we have observed that this distance function gives better performance in most of the cases as compared to other distance measures such as l_2 or χ^2. We apply the following normalization to all the meshes of the database to secure RST invariance of the features. For translation invariance, the object's center of mass is translated to the origin. For scale invariance, the area-weighted average distance of surface points to the origin is set to unity. We have observed that, with this scaling operation, the frequency of the distance of a surface point to the mesh center exceeding 2 becomes negligible. This allows us to set empirical upper limits $r_{\max}$ and $d_{t,\max}$to the magnitude components r_P and $d_{t,P}$, respectively. Finally, to guarantee rotation and reflection invariance, we follow the "continuous" PCA approach of Vranić [7]. All the codes for our descriptors as well as for those proposed in the literature (cord and angle histograms [11], D1 and D2 shape distributions [10], EGI [8] and CEGI [9], 3DHT [12]) have been implemented in MATLAB 7.0 (R14) environment, using C MEX external interface for time-consuming jobs. For FGT, we have used the implementation provided by Yang et al. [16].

The acronyms of the descriptors we have experimented are listed in Tables 3 and 4. They will subsequently be used in graph annotations. The details about descriptor sizes are given in the corresponding sections.

There are two alternative ways of combining descriptors, by multivariate density evaluation or by concatenating estimated univariate densities. The multivariate descriptors (Sr, St, Sc, and Sn) that we consider in our experiments are derived from S_r, S_t, S_c, and S_n features as given in the first four rows of Table 4. Alternatively, descriptors for multiple scalar features, for example, S_{ri}, $i = 1, \ldots, 4$, can

Table 4: Density-based 3D shape descriptors and their sizes.

Descriptor	Acronym	Size N
Radial (S_r) *density*	Sr	$8 \times 128 = 1024$
Tangent pl. (S_t) *density*	St	$8 \times 128 = 1024$
Cross-product (S_c) *density*	Sc	$8 \times 320 = 2560$
Normal (S_n) *density*	Sn	128
Univ. dens. of S_r *components*	[Sr1,Sr2,Sr3,Sr4]	$4 \times 64 = 256$
Univ. dens. of S_t *components*	[St1,St2,St3,St4]	$4 \times 64 = 256$

Table 5: DCG values for possible bandwidth selection strategies on PSB training meshes.

Bandwidth setting	Sr	St	Sc
Triangle level	0.352	—	—
Mesh level	0.511	0.514	0.499
Database level	0.541	0.567	0.543

separately be computed by univariate density estimation and then concatenated in a joint vector, as in the last two rows of Table 4. Let $A_1, A_2, \ldots, A_L$ denote L generic (one- or multidimensional) features and let $\mathbf{f}_{A_1}, \mathbf{f}_{A_2}, \ldots, \mathbf{f}_{A_L}$ denote the corresponding density-based descriptors with $N_1, N_2, \ldots, N_L$ components, respectively, (N_i, $i = 1, \ldots, L$ corresponds to the number of target points on which the density of feature A_i has been evaluated or equivalently to the size of the vector $\mathbf{f}_{A_i}$). Square bracketing $[A_1, A_2, \ldots, A_L]$ that appears in subsequent graphs and tables indicates the concatenation of the shape descriptors $[\mathbf{f}_{A_1}, \mathbf{f}_{A_2}, \ldots, \mathbf{f}_{A_L}]$ resulting in a vector of size $N_1 + N_2 + \cdots + N_L$. For notational simplicity, we will refer to the descriptor $\mathbf{f}_{A_1}$ consisting of the density vector as A1-descriptor; similarly, [A1, A2] will be the shorthand notation for the descriptor $[\mathbf{f}_{A_1}, \mathbf{f}_{A_2}]$. Note finally that the generic feature A_i can be either a vector by construction or a scalar obtained by taking a component of some other multidimensional feature.

4.2.1. *Impact of bandwidth selection*

The KDE approach critically depends upon the judicious setting of the bandwidth parameters. We tested the triangle, mesh and database level alternatives presented in Section 3.4 on our multidimensional local features S_r, S_t, and S_c (the computationally expensive triangle-level setting was only tested for S_r). Since we have observed that the off-diagonal terms of the bandwidth matrices are negligible as compared to the diagonal terms, we use only diagonal bandwidth matrices $H = \text{diag}(h_1, \ldots, h_d)$. For the mesh level and database level, we apply the Scott's rule-of-thumb. For the triangle level, we employ the KDE toolbox developed by Ihler [28] since the available FGT implementation does not allow a different bandwidth per triangle [16]. The KDE toolbox makes use of kd-trees and reduces the computational burden considerably, though not to the extent achieved by FGT. Table 5 compares the DCG scores obtained with Sr, St, and Sc-descriptors on the PSB training set. Figure 5 shows the precision-recall plots corresponding to mesh and database level settings for Sr and St-descriptors. We clearly observe that setting the bandwidth H at database level is more advantageous as compared to triangle and mesh level settings. Any further results reported are therefore for the database level setting of H. In Table 6, we provide the average Scott bandwidth values obtained from PSB training meshes for S_r, S_t, and S_c features.

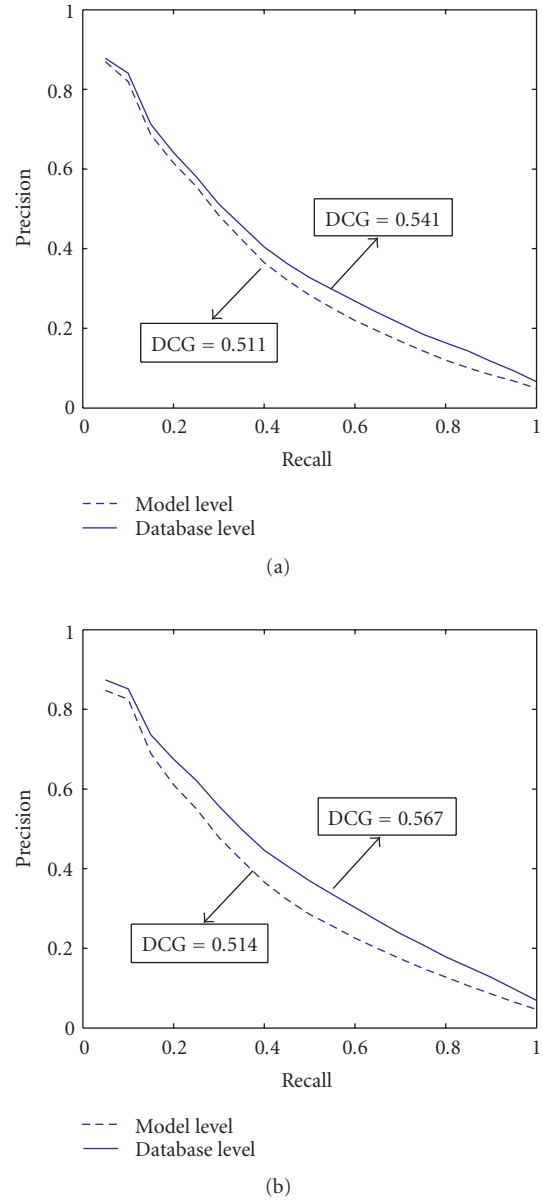


Figure 5: Precision-recall curves with a bandwidth selection made at mesh level versus database level for Sr-descriptor (a) and St-descriptor (b) on PSB training set.

4.2.2. *Univariate versus multivariate density-based descriptors*

In this section, we compare the impact of combining descriptors on the retrieval performance. As discussed before,

Table 6: The average Scott bandwidth obtained from the PSB training meshes.

Descriptor	h_1	h_2	h_3	h_4
Sr	0.20	0.35	0.25	0.15
St	0.20	0.25	0.25	0.30
Sc	0.20	0.15	0.25	0.25

descriptors can be compounded either by concatenating univariate descriptors or by multivariate density estimation. One can conjecture that the multivariate descriptors, resulting from the joint density functions of features, are richer in information content since component-wise dependencies are also taken into account. On the other hand, univariate densities are much simpler to estimate and do not incur into dimensionality problems. In our experiments, each univariate density is evaluated at 64 target points. Accordingly, a 4-tuple concatenation, such as [Sr1,Sr2,Sr3,Sr4], results in a descriptor of size $N = 4\times 64 = 256$. For multivariate density descriptors Sr and St, recall that $N_{\text{dir}} = 128$ and for Sc, $N_{\text{dir}} = 320$ (see Section 3.5). N_{mag} being chosen equal to 8 in all cases, the size of the Sr and St-descriptors is $N = 8 \times 128 = 1024$ and the size of the Sc-descriptor is $N = 8 \times 320 = 2560$. Figures 6 and 7 with Table 7 explicitly show that the multivariate density-based descriptors are superior to the descriptors obtained by the concatenation of univariate densities for all feature types on both databases.

4.2.3. *Comparison of density-based descriptors with their histogram-based peers*

One of the motivations of this work is to show that a considerable improvement in the retrieval performance can be obtained by more rigorous and accurate computation of shape distributions as compared to more practical ad hoc histogram approaches. Notice that we interpret the term "histogram-based descriptor" for any count-and-accumulate type of procedure. This way we can refer to analogous descriptors in the literature as histogram-based whenever they count-and-accumulate local information to obtain a global shape descriptor [8–12].

An interesting case in point is Cord and Angle Histograms (CAH) [11]. The features in CAH are identical to the individual scalar components $r_P, \hat{r}_{P,x}, \hat{r}_{P,y}$, and $\hat{r}_{P,z}$ of our S_r feature up to a parameterization. In [11], the authors consider the length of a cord (corresponding to r_P) and the two angles between a cord and the first two principal directions (corresponding to $\hat{r}_{P,x}$ and $\hat{r}_{P,y}$). Notice that in our parameterization of S_r, we consider the Cartesian coordinates rather than the angles. In order to compare with our [Sr1,Sr2,Sr3,Sr4]-descriptor, we implemented the CAH-descriptor by also considering the histogram of the angle with the third principal direction. The resulting CAH-descriptor is thus the concatenation of one cord length and three angle histograms. Each histogram consisting of 64 bins leads to a descriptor of total size $N = 4 \times 64 = 256$. [Sr1,Sr2,Sr3,Sr4]-descriptor, again of size 256,

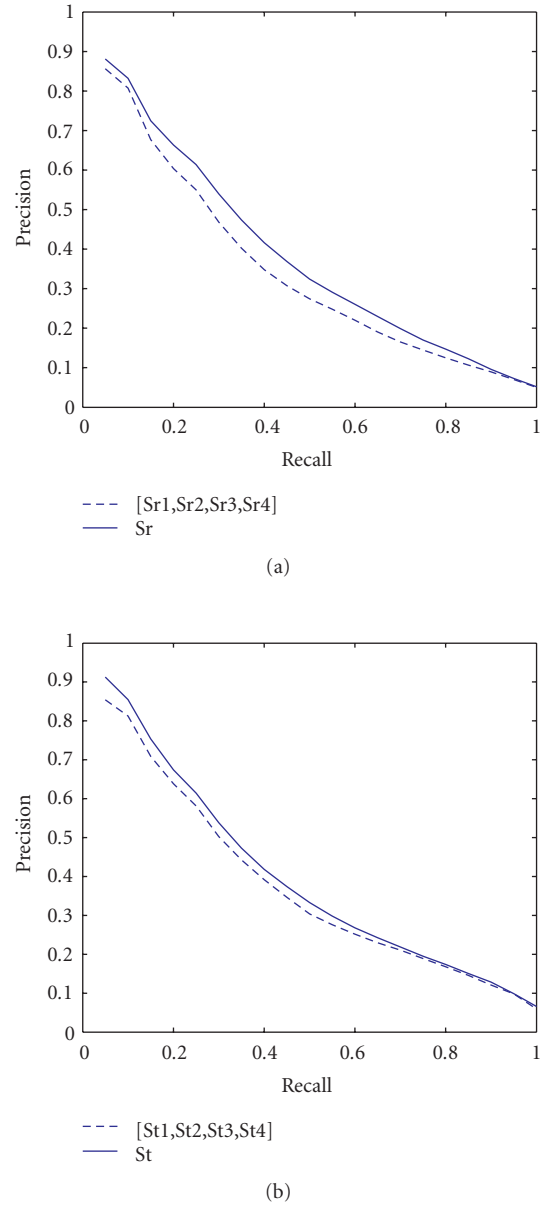


Figure 6: Precision-recall curves for [Sr1,Sr2,Sr3,Sr4] versus Sr (a) and [St1,St2,St3,St4] versus St (b) on PSB.

differs from CAH in three aspects: first, it uses a different parameterization of the angle (direction) components; second, the local feature values are calculated by (6) instead of using mere barycentric sampling; third, it employs KDE instead of histogram computation. In Figure 8, we provide the precision-recall curve corresponding to CAH and [Sr1,Sr2,Sr3,Sr4] on PSB test set and on SCUdb. The respective DCG values are 0.434 and 0.501 for PSB, 0.681 and 0.698 for SCUdb, indicating the superior performance of our framework under identical feature sets. An additional improvement can be gained by estimating the joint density of S_r, leading to the Sr-descriptor. That is, in contrast to the concatenation of univariate densities, we directly use the joint density of S_r as a descriptor. The DCG value

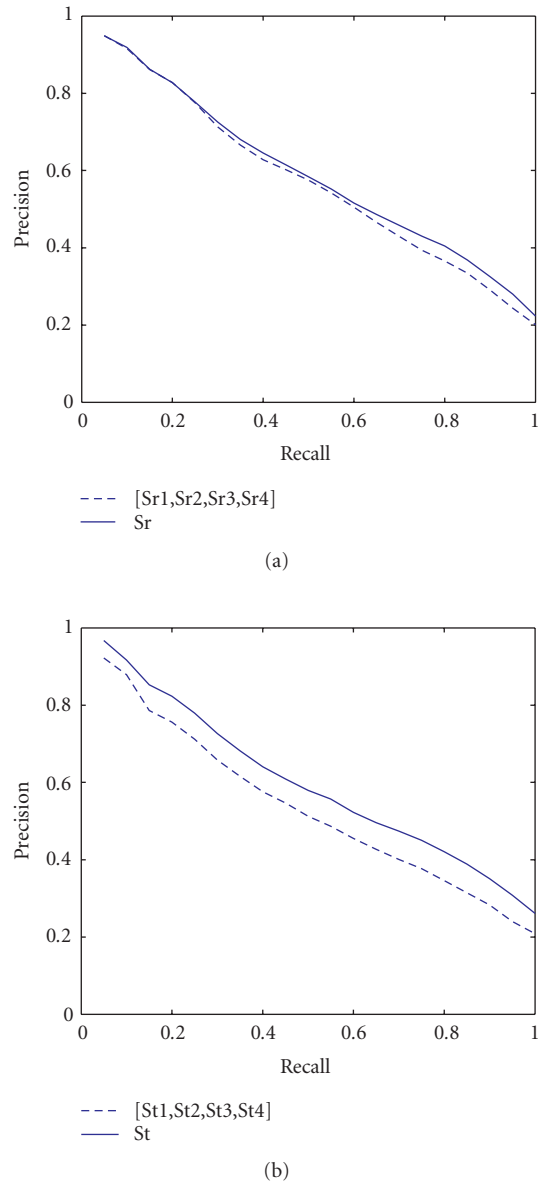


Figure 7: Precision-recall curves for [Sr1,Sr2,Sr3,Sr4] versus Sr (a) and [St1,St2,St3,St4] versus St on SCUdb (b).

of the Sr-descriptor is 0.533 on PSB and 0.708 on SCUdb, one more step of improvement as compared to the concatenated univariate case [Sr1,Sr2,Sr3,Sr4] (DCG = 0.501 on PSB and DCG = 0.698 on SCUdb). Note that the performance improvement using our scheme is less impressive over SCUdb than over PSB. This can be explained by the fact that SCUdb meshes are much denser than PSB meshes in number of triangles. As the number of observations increases, the accuracies of the histogram method and KDE become comparable and both methods result in similar descriptors. This also indicates that the KDE methodology is especially appropriate for coarser mesh resolutions as in PSB.

A second instance of our framework outperforming its competitor is with the EGI-descriptor [2, 5, 8], which

Table 7: Retrieval statistics for univariate and multivariate density-based descriptors.

	Descriptor	NN	FT	ST	E	DCG
PSB	[Sr1,Sr2,Sr3,Sr4]	0.436	0.222	0.306	0.180	0.501
	Sr	0.499	0.260	0.343	0.201	0.533
	[St1,St2,St3,St4]	0.451	0.250	0.348	0.202	0.533
	St	0.523	0.267	0.364	0.210	0.543
SCUdb	[Sr1,Sr2,Sr3,Sr4]	0.701	0.430	0.555	0.314	0.698
	Sr	0.745	0.452	0.568	0.323	0.709
	[St1,St2,St3,St4]	0.632	0.400	0.520	0.298	0.662
	St	0.754	0.473	0.575	0.324	0.712

consists of binning the surface normals. The density of our $S_n(P) = \hat{\mathbf{n}}_P$ feature is equivalent to the EGI-descriptor. There can be different choices for binning surface normals, for example, by mapping the normal of a certain mesh triangle to the closest bin over the unit sphere and augmenting that bin by the relative area of the triangle. Such an approach requires a very densely discretized unit sphere and the resulting descriptor is not very efficient in terms of storage. In the present work, similarly to [12], we preferred the following implementation for the EGI-descriptor. First, 128 unit norm vectors $\hat{\mathbf{n}}_{\text{bin},j}$, $j = 1,\ldots,128$, are obtained as histogram bin centers by octahedron subdivision, as described in Section 3.5. Then, the contribution of each triangle T_k, $k = 1,\ldots,K$, with normal vector $\hat{\mathbf{n}}_k$ to the nth bin center is computed as $w_k|\langle\hat{\mathbf{n}}_k,\hat{\mathbf{n}}_{\text{bin},j}\rangle|$ if $|\langle\hat{\mathbf{n}}_k,\hat{\mathbf{n}}_{\text{bin},j}\rangle| \geq 0.7$ or otherwise as zero (recall that w_k is the relative area of the kth triangle). The use of the absolute value is needed because some models as those in the PSB set cannot provide orientation information. The Sn-descriptor of the same size, that is, 128, achieves a superior DCG of 0.478 as compared to the DCG score of 0.438 for EGI on PSB (see Figure 9). For SCUdb, the DCG-performance differential is even more pronounced (DCG = 0.589 for Sn, DCG = 0.535 for EGI) noting that for low recall values (recall < 0.2), the EGI-descriptor is better than Sn (see Figure 9).

A third instance of comparison can be considered between our St-descriptor and the 3DHT-descriptor [12] since both of them use local tangent plane parameterization. The procedure for the 3DHT descriptor is carried out as follows. We first recall that the 3DHT-descriptor is a histogram constructed by accumulating mesh surface points over planes in 3D space. Each histogram bin corresponds to a plane $\mathcal{P}_{ij}$ parameterized by its normal distance $d_{t,i}$, $i = 1,\ldots,N_{\text{mag}}$, to the origin and its normal direction $\hat{\mathbf{n}}_{\text{bin},j}$, $j = 1,\ldots,N_{\text{dir}}$. Clearly, there can be $N_{\text{mag}} \times N_{\text{dir}}$ such planes and the resulting descriptor is of size $N = N_{\text{mag}} \times N_{\text{dir}}$. We can obtain such a family of planes exactly as described in Section 3.5 and in [12]. In our experiments, we have used $N_{\text{mag}} = 8$ distance bins sampled within the range $[0, 2]$ and $N_{\text{dir}} = 128$ uniformly sampled normal directions. This results in a 3DHT descriptor of size $N = 1024$. To construct the Hough array, one first takes a plane with normal direction $\hat{\mathbf{n}}_{\text{bin},j}$, $j = 1,\ldots,N_{\text{dir}}$, at each triangle barycenter $\mathbf{m}_k$, $k = 1,\ldots,K$,

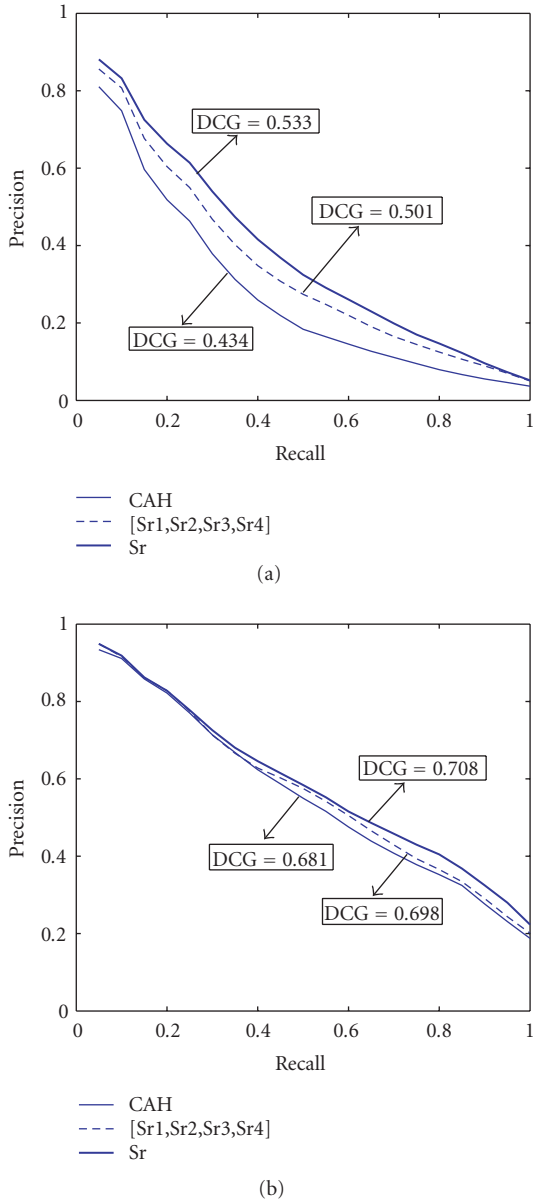


Figure 8: Precision-recall curves for CAH, [Sr1,Sr2,Sr3,Sr4] (concatenated) and Sr (joint) on PSB test set (a) and SCUdb (b).

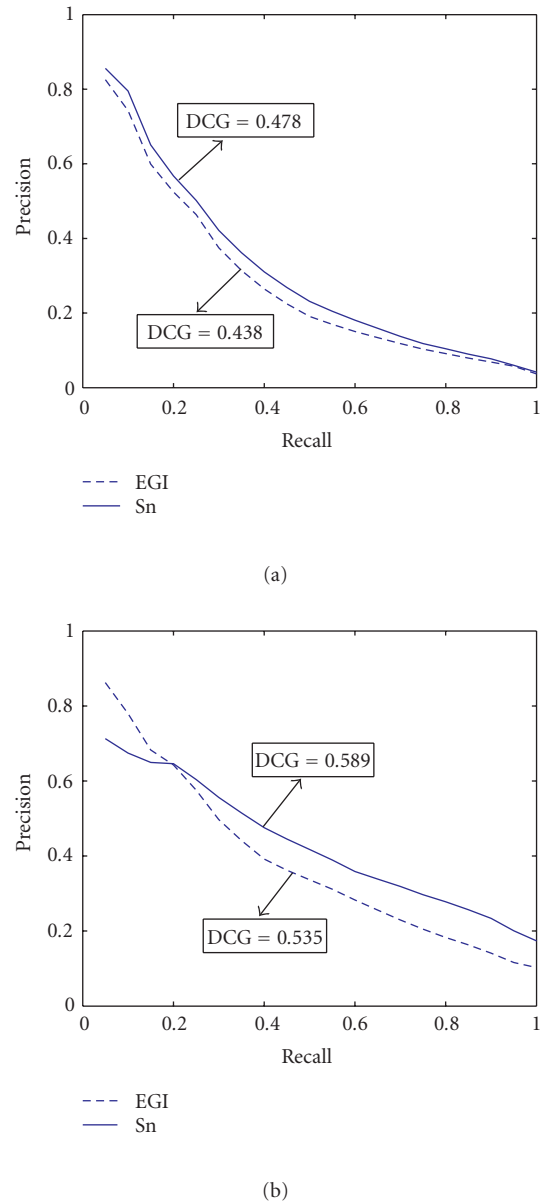


Figure 9: Precision-recall curves for EGI and Sn on PSB test set (a) and SCUdb (b).

and then calculates the normal distance of the plane to the origin by $|\langle \mathbf{m}_k, \hat{\mathbf{n}}_{\text{bin},j} \rangle|$. The resulting value is quantized to the closest $d_{t,i}$, $i = 1, \ldots, N_{\text{mag}}$, and then the bin corresponding to the plane $\mathcal{P}_{ij}$ is augmented by $w_k |\langle \hat{\mathbf{n}}_k, \hat{\mathbf{n}}_{\text{bin},j} \rangle|$ if $|\langle \hat{\mathbf{n}}_k, \hat{\mathbf{n}}_{\text{bin},j} \rangle| \geq 0.7$ (the value of 0.7 is suggested by Zaharia and Prêteux [12] and we have also verified its performance-wise optimality). In Figure 10, we compare the St- and the 3DHT-descriptors in terms of precision-recall curves. On PSB, the St-descriptor yields a DCG of 0.543, a worse score against 0.577 of the 3DHT-descriptor. This can be attributed largely to the fact that the 3DHT-descriptor employs an implicit correction for normal orientations by the weighting scheme $w_k |\langle \hat{\mathbf{n}}_k, \hat{\mathbf{n}}_{\text{bin},j} \rangle|$ according to which only normal direction $\hat{\mathbf{n}}_k$ matters but not its orientation. Our St-descriptor does not make use of such a correction and considers the normal orientations as they are provided by the list of triangles in the mesh. Accordingly, we explain the negative performance gap between St and 3DHT by the fact that, on PSB meshes, information regarding normal orientations might be compromised. On the other hand, for SCUdb, the performance of St (DCG = 0.712) parallels that of 3DHT noting that 3DHT remains slightly better (DCG = 0.727).

4.2.4. General performance comparison

In this section, we compare the descriptors that we propose (univariate, concatenated, or multivariate) first among

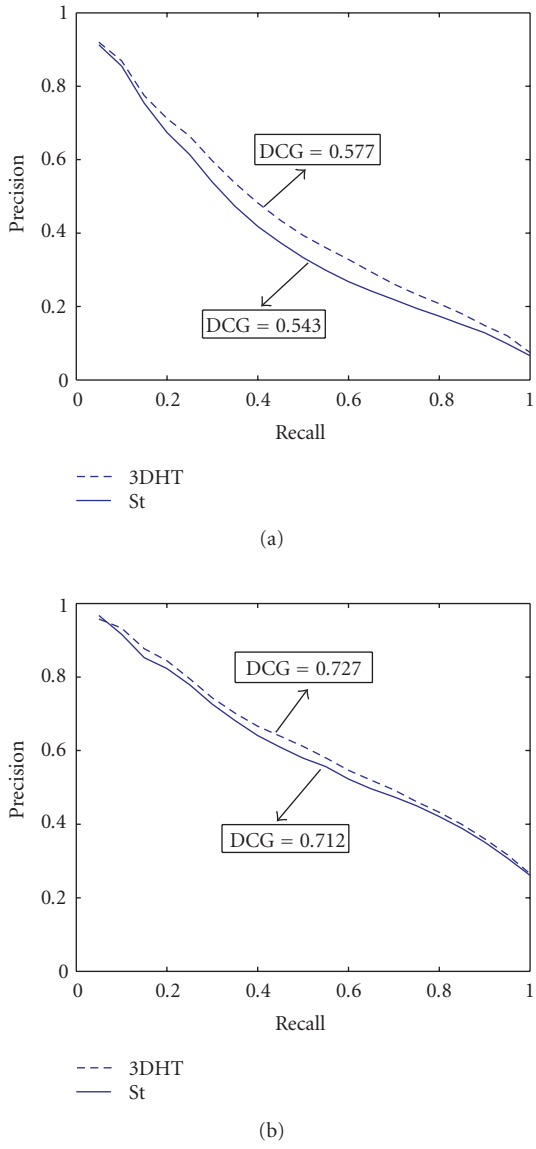


Figure 10: Precision-recall curves for 3DHT and St on PSB test set (a) and SCUdb (b).

themselves and then with various other descriptors existing in the literature.

In Table 8, we see the competition within the Sr, St, and Sc set and their various combinations. Since pairing the features results in higher dimensions (8 or 12) precluding multivariate density estimation, we use concatenation of the 4-variate densities. It is interesting to observe that the pairwise concatenations [Sr,St], [Sr,Sc], and [St,Sc] of size 2048, 3584 and 3584, respectively, increase the DCG and NN scores significantly. We can conclude that each local feature must be reporting aspects on the shape not covered by the remaining ones, albeit their similarity. Furthermore, the triplet concatenation [Sr,St,Sc] of size 4608 boosts the DCG and NN performance further. We also note that, on a Pentium 4 PC (2.4 GHz CPU, 2 GB RAM), the [Sr,St,Sc]-descriptor can be computed in less than one second on the average over PSB test set meshes, which indicates that our density-based descriptors are very time-efficient and suitable for practical online applications.

Table 9 finally summarizes the experimental results conducted to compare our density-based descriptors with other histogram-based descriptors. For both databases, PSB and SCUdb, the [Sr,St,Sc]-descriptor comes at the top in all performance fields. Furthermore, the second place is taken by a pairwise concatenation which is more storage-efficient and even more time-efficient than [Sr,St,Sc]: [Sr,St] for PSB and [St,Sc] for SCU.

The density-based framework does not only outperform histogram-based descriptors but also proves to be effective as compared to other more general state-of-the-art shape descriptors. In fact, based on the scores on PSB test set reported in [5], the [Sr,St,Sc]-descriptor has the highest DCG score among all other well-known 3D shape descriptors, as shown in Figure 11. Except for 3DHT [12] and CAH [11], all the descriptor scores shown in Figure 11 are taken from [5]. We refer the reader to [5] for brief descriptions and acronyms of these descriptors. The [Sr,St,Sc]-descriptor has a DCG value of 0.607, while the next best descriptor radialized extent function (REXT) [7, 24] has a DCG value of 0.601 [5]. Note also that the [Sr,St]-descriptor (DCG = 0.599) ranks third in the competition. The average REXT-descriptor size reported in [5] is 17.5 kilobytes, while for our [Sr,St,Sc]-descriptor this figure is 22 kilobytes. The average generation time for the REXT-descriptor is 2.2 seconds [5], while our [Sr,St,Sc]-descriptor can be computed in 0.9 seconds on the average on comparable hardware configurations.

5. CONCLUSION

We have proposed a novel methodology to obtain 3D shape descriptors and evaluated its impact in a retrieval scenario. We have shown that shape descriptors derived as kernel density estimates of local surface features prove more advantageous compared to the count-and-accumulate-based histogram descriptors. Firstly, one main advantage accrues from the fact that our descriptors are true probability density functions of geometrical quantities defined over the model surface. Secondly, our surface sampling is not as crude as just considering triangle barycenters or as profuse as random sampling, but judiciously chooses the triangle characteristics. Thirdly and most importantly, the KDE-based approach deals with multidimensional surface features as easily as with scalar features. The bandwidth parameters in KDE provide a more gracious control over finite sample-size and dimensionality problems, while with multivariate histograms one can only adjust the bin widths [13, 14]. The local surface information brought by multidimensional features proves to be more discriminating than scalar ones.

The proposed framework applies to 3D objects represented as triangular meshes but extension to point-cloud representations is straightforward. Concerning hidden triangles encountered in triangular "soups," we remark that we do not try to detect such degeneracies and process them as any other triangles. They introduce noise in the density

Table 8: DCG and NN scores for the combination of density-based descriptors.

		Sr	St	Sc	[Sr,St]	[Sr,Sc]	[St,Sc]	[Sr,St,Sc]
PSB	DCG	0.533	0.543	0.533	0.599	0.579	0.585	0.607
	NN	0.500	0.527	0.487	0.606	0.572	0.584	0.615
SCUdb	DCG	0.708	0.712	0.732	0.731	0.742	0.744	0.746
	NN	0.745	0.754	0.733	0.788	0.776	0.774	0.786

Table 9: General performances of histogram and density-based descriptors.

	Descriptor	NN	FT	ST	E	DCG	NDCG
PSB	[Sr,St,Sc]	0.615	0.339	0.434	0.251	0.607	0.214
	[Sr,St]	0.606	0.333	0.423	0.245	0.599	0.199
	3DHT	0.588	0.311	0.396	0.230	0.577	0.154
	D2	0.363	0.168	0.245	0.145	0.448	−0.103
	EGI	0.311	0.165	0.245	0.145	0.438	−0.124
	CAH	0.332	0.159	0.229	0.137	0.433	−0.133
	D1	0.256	0.119	0.185	0.107	0.397	−0.207
SCUdb	[Sr,St,Sc]	0.786	0.518	0.617	0.355	0.746	0.106
	[St,Sc]	0.774	0.513	0.622	0.355	0.744	0.103
	3DHT	0.778	0.485	0.603	0.336	0.727	0.079
	CAH	0.678	0.427	0.536	0.309	0.681	0.010
	D1	0.643	0.366	0.486	0.272	0.646	−0.042
	D2	0.643	0.355	0.467	0.264	0.643	−0.048
	EGI	0.489	0.252	0.349	0.203	0.535	−0.207

estimation but not to the extent to alter the density-based descriptor drastically. Furthermore, hidden triangles present in PSB remain in small proportion and SCUdb models are manifold and free of hidden triangles.

Our framework should be viewed as an application of kernel density estimation [13, 14] with either variable (triangle or mesh levels) or fixed (database level) bandwidth parameters selection [25]. We have also demonstrated that density-based descriptors are much more discriminative in retrieval when the bandwidth parameters are set at database level as compared to mesh or triangle level setting. We think that the database level strategy smoothes out individual shape details and emphasizes global shape properties as appropriate for object retrieval and classification tasks; while the other two options, especially the triangle level strategy, result in an overfitting of the feature density and hamper the descriptor's discrimination ability. Furthermore, the computational advantage of density-based descriptors enabled by FGT with a database-dependent bandwidth matrix is very promising for practical online applications.

When combined together, the multivariate density-based 3D shape descriptors introduced in this work outperform the existing histogram-based techniques in the literature. The retrieval competition took place on two databases, PSB and SCUdb, which are fundamentally different in semantic content and mesh quality. In addition, the performance advantage of density-based descriptors over its competitors is not limited to histogram-based ones, as shown in the more general comparison where our [Sr,St,Sc]-descriptor reaches the top position in the category of purely 3D descriptors reported in [5]. As a side remark, based on nearest-neighbor scores of our descriptors, we conjecture that they would also perform well in recognition applications.

In summary, a general framework using KDE has been developed, that covers existing and novel descriptors. Our method enables the use of arbitrary one- or multidimensional surface features for retrieval, recognition, and classification of 3D objects. Future research will concentrate on potential improvements of decision fusion. For example, several retrievers can operate in parallel and one can consider rank-weighted reordering of the retrieved objects. A second natural avenue of research is in the direction of second-order features. We will tackle the problem of designing second-order features that would serve as natural proxies for curvature-like quantities. Curvature is in fact difficult to work with because of the estimation inaccuracies involved in its computation. Nevertheless, it can be conjectured that the kernel-based approach, thanks to its smoothing behavior, may be useful in deriving curvature-driven 3D shape descriptors. One of our future objectives is thus to arrive at an exhaustive set of first- and second-order features and to discover computational limits of the density-based approach. A side issue is to render the proposed descriptors more effective in discrimination and more efficient in terms of storage size by adequately sampling the local feature domains for target evaluation points. A further question that should be considered is to which extent

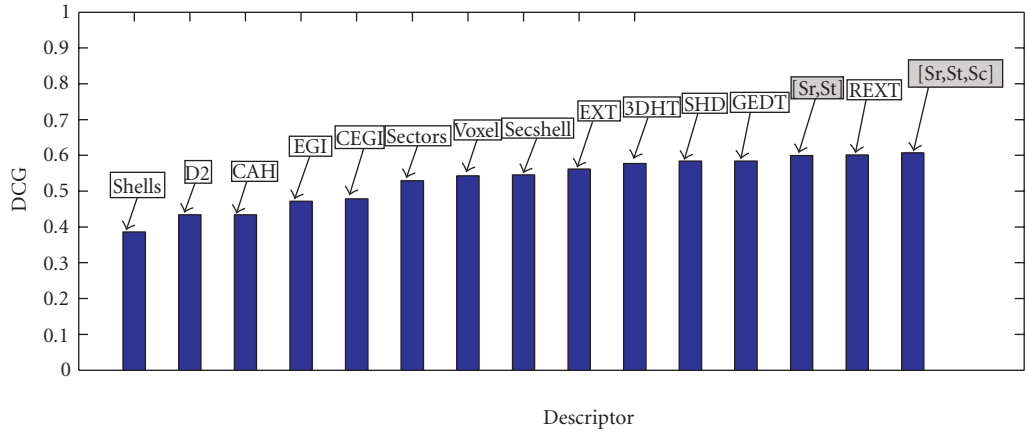


Figure 11: Comparison of 3D shape descriptors on PSB test set. (Except CAH, 3DHT, and our descriptors, DCG values are taken from [5].)

the combination of the available features can be exploited, that is, how large the feature dimension of the multivariate densities can be.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their helpful comments and suggestions on the earlier version of the manuscript. This research was supported by BU Project 03A203 and TUBITAK Project 103E038.

REFERENCES

[1] B. Bustos, D. A. Keim, D. Saupe, T. Schreck, and D. V. Vranić, "Feature-based similarity search in 3D object databases," *ACM Computing Surveys*, vol. 37, no. 4, pp. 345–387, 2005.

[2] J. W. H. Tangelder and R. C. Veltkamp, "A survey of content based 3D shape retrieval methods," in *Proceedings of International Conference on Shape Modeling and Applications (SMI '04)*, pp. 145–156, Genova, Italy, June 2004.

[3] R. J. Campbell and P. J. Flynn, "A survey of free-form object representation and recognition techniques," *Computer Vision and Image Understanding*, vol. 81, no. 2, pp. 166–210, 2001.

[4] 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, 2005.

[5] P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser, "The Princeton shape Benchmark," in *Proceedings of International Conference on Shape Modeling and Applications (SMI '04)*, pp. 167–178, Genova, Italy, June 2004.

[6] T. Tung, *Indexation 3D de bases de données d'objets par graphes de Reeb améliorés*, Ph.D. thesis, Ecole Nationale Supérieure des Télécommunications (ENST), Paris, France, 2005.

[7] D. V. Vranić, *3D model retrieval*, Ph.D. thesis, University of Leipzig, Leipzig, Germany, 2004.

[8] B. K. P. Horn, "Extended Gaussian images," *Proceedings of the IEEE*, vol. 72, no. 12, pp. 1671–1686, 1984.

[9] S. Kang and K. Ikeuchi, "The complex EGI: a new representation for 3-D pose determination," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 7, pp. 707–721, 1993.

[10] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin, "Shape distributions," *ACM Transactions on Graphics*, vol. 21, no. 4, pp. 807–832, 2002.

[11] E. Paquet and M. Rioux, "Nefertiti: a query by content software for three-dimensional models databases management," in *Proceedings of the 1st International Conference on Recent Advances in 3-D Digital Imaging and Modeling (3DIM '97)*, pp. 345–352, Washington, DC, USA, May 1997.

[12] T. Zaharia and F. Prêteux, "Indexation de maillages 3D par descripteurs de forme," in *Actes 13ème Congrès Francophone AFRIF-AFIA Reconnaissance des Formes et Intelligence Artificielle (RFIA '02)*, pp. 48–57, Angers, France, January 2002.

[13] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, Wiley-Interscience, New York, NY, USA, 2000.

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

[15] L. Greengard and J. Strain, "The fast Gauss transform," *SIAM Journal on Scientific and Statistical Computing*, vol. 12, no. 1, pp. 79–94, 1991.

[16] C. Yang, R. Duraiswami, N. A. Gumerov, and L. Davis, "Improved fast Gauss transform and efficient kernel density estimation," in *Proceedings of the 9th IEEE International Conference on Computer Vision (ICCV '03)*, vol. 1, pp. 464–471, Nice, France, October 2003.

[17] K. Siddiqi, A. Shokoufandeh, S. J. Dickinson, and S. W. Zucker, "Shock graphs and shape matching," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV '98)*, pp. 222–229, Bombay, India, January 1998.

[18] M. Hilaga, Y. Shinagawa, T. Kohmura, and T. L. Kunii, "Topology matching for fully automatic similarity estimation of 3D shapes," in *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '01)*, pp. 203–212, Los Angeles, Calif, USA, August 2001.

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

[20] H. Sundar, D. Silver, N. Gagvani, and S. J. Dickinson, "Skeleton based shape matching and retrieval," in *Proceedings of International Conference on Shape Modeling and Applications (SMI '03)*, pp. 130–139, Seoul, Korea, May 2003.

[21] M. Ankerst, G. Kastenmüller, H.-P. Kriegel, and T. Seidl, "3D shape histograms for similarity search and classification in

spatial databases," in *Proceedings of the 6th International Symposium on Advances in Spatial Databases (SSD '99)*, vol. 1651 of *Lecture Notes in Computer Science*, pp. 207–226, Hong Kong, July 1999.

[22] T. Funkhouser, P. Min, M. Kazhdan, et al., "A search engine for 3D models," *ACM Transactions on Graphics*, vol. 22, no. 1, pp. 83–105, 2003.

[23] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz, "Rotation invariant spherical harmonic representation of 3D shape descriptors," in *Proceedings of the Eurographics/ACM SIGGRAPH Symposium on Geometry Processing (SGP '03)*, pp. 156–164, Aachen, Germany, June 2003.

[24] D. V. Vranić, "An improvement of rotation invariant 3D-shape based on functions on concentric spheres," in *Proceedings of the IEEE International Conference on Image Processing (ICIP '03)*, vol. 3, pp. 757–760, Barcelona, Spain, September 2003.

[25] D. Comaniciu, V. Ramesh, and P. Meer, "The variable bandwidth mean shift and data-driven scale selection," in *Proceedings of the 8th International Conference on Computer Vision (ICCV '01)*, vol. 1, pp. 438–445, Vancouver, BC, Canada, July 2001.

[26] W. H. Press, B. P. Flannery, and S. A. Teukolsky, *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge University Press, Cambridge, UK, 1992.

[27] S. Goodall, P. H. Lewis, K. Martinez, et al., "SCULPTEUR: multimedia retrieval for museums," in *Proceedings of the Image and Video Retrieval: 3rd International Conference (CIVR '04)*, vol. 3115 of *Lecture Notes in Computer Science*, pp. 638–646, Dublin, Ireland, July 2004.

[28] A. Ihler, *Kernel density estimation toolbox for MATLAB (R13)*, 2003.

Ceyhun Burak Akgül received the B.S. and M.S. degrees in electrical and electronics engineering from Boğaziçi University, Istanbul, in 2002 and 2004, respectively. He has been pursuing his Ph.D. degree jointly at Boğaziçi University and Télécom Paris (Ecole Nationale Supérieure des Télécommunications) since 2004. In the framework of his Ph.D. thesis, he is currently working on 3D shape descriptors and statistical similarity learning for object retrieval and classification. His main research interests are 2D/3D image analysis and statistical pattern recognition with applications on multimedia data.

Bülent Sankur received his B.S. degree in electrical engineering at Robert College, Istanbul, and completed his M.S. and Ph.D. degrees at Rensselaer Polytechnic Institute, NY, USA. He has been teaching at Boğaziçi University in the Department of Electric and Electronics Engineering. His research interests are in the areas of digital signal processing, image and video compression, biometry, cognition, and multimedia systems. He held visiting positions at University of Ottawa, Technical University of Delft, and Ecole Nationale Supérieure des Télécommunications, Paris. He was the Chairman of ICT96 (International Conference on Telecommunications) and of EUSIPCO05 (The European Conference on Signal Processing) as well as Technical Chairman of ICASSP00.

Yücel Yemez received the B.S. degree from Middle East Technical University, Ankara, Turkey, in 1989, and the M.S. and Ph.D. degrees from Boğaziçi University, Istanbul, Turkey, respectively, in 1992 and 1997, all in electrical engineering. From 1997 to 2000, he was a Postdoctoral Researcher in the Image and Signal Processing Department of Télécom Paris (Ecole Nationale Supérieure des Télécommunications). Currently, he is an Assistant Professor of the Computer Engineering Department at Koç University, Istanbul, Turkey. His current research is focused on various fields of computer vision and graphics.

Francis Schmitt received an Engineering degree from Ecole Centrale de Lyon, France, in 1973 and received a Ph.D. degree in applied physics from the University Pierre et Marie Curie, Paris VI, France, in 1979. He has been a Member of Télécom Paris (Ecole Nationale Supérieure des Télécommunications) since 1973. He is currently Full Professor at the Image and Signal Processing Department and Head of the image processing group. His main interests are in computer vision, 3D modeling, image and 3D object indexing, computational geometry, multispectral imagery, and colorimetry. He is the author or coauthor of about 150 publications in these fields.