Multivariate Density-Based 3D Shape Descriptors

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

Abstract

In this brief communication, we provide an overview of the density-based shape description framework and the details of the runs we have submitted to the SHREC'07 Watertight Models Track. We also present the results of an improved matching scheme leading to more effective 3D retrieval.

1 Introduction

Density-based shape description is a framework to extract 3D shape descriptors from local surface features characterizing the object geometry [1, 2]. The feature information is processed with the kernel methodology for density estimation (KDE) [3] and the probability density function (pdf) of the local feature is estimated at chosen target points. The shape descriptor vector is then simply a discretized version of this probability density. This density-based approach provides a mechanism to convert local shape evidences, using KDE, into a global shape description. Our recent work on density-based shape descriptors [1, 2] for 3D object retrieval has proven that this scheme is both computationally rapid and effective compared to other state-of-the-art descriptors. In this brief communication, we provide an overview of the density-based shape description framework in Section 2 and the details of the runs we have submitted to the SHREC'07 Watertight Models Track in Section 3. More details about the methods and its performance can be found in [1, 2].

2 Overview of the Description Framework

A density-based descriptor of a 3D shape is defined as the discretized pdf of some surface feature S taking values within a subspace $\mathcal{R}_S$ of $\mathbb{R}^m$. The feature S is local to the surface patch and treated as a random variable. For an object O_l represented as a triangular mesh, evaluating the feature S at each triangle and/or vertex, we can obtain a set of observations about S, that we call *the source set*, denoted as $\{s_k \in \mathcal{R}_S\}_{k=1}^K$. Let $f(\cdot|O_l)$ be the pdf of S. Using the source set $\{s_k \in \mathcal{R}_S\}_{k=1}^K$, the value of this pdf at an arbitrary m-dimensional point t (which is in the range space $\mathcal{R}_S$ of the feature S) can be estimated by the following KDE equation [3]:

$$f(t|O_l) = \sum_{k=1}^{K} w_k \, |H|^{-1} \, \mathcal{K}(H^{-1}(t - s_k)), \qquad (1)$$

where $\mathcal{K} : \mathbb{R}^m \rightarrow \mathbb{R}$ is a kernel function, H is a $m \times m$ matrix composed of a set of design parameters called bandwidth parameters, and w_k is the importance weight associated with the kth observation s_k. Suppose that we have specified a finite set of points $\overline{\mathcal{R}}_S = \{t_n \in \mathcal{R}_S\}_{n=1}^N$, called *the target set*, within $\mathcal{R}_S$. The density-based descriptor $\mathbf{f}_{S|O_l}$ for the object O_l (with respect to the feature S) is then simply an N-dimensional vector whose entries consist of the pdf values computed at the target set $\overline{\mathcal{R}}_S = \{t_n \in \mathcal{R}_S\}_{n=1}^N$, that is, $\mathbf{f}_{S|O_l} = [f(t_1|O_l), \ldots, f(t_N|O_l)]$. To convert this general framework into a practical description scheme, we have to address the following issues:

Feature Design: Which surface features should be used to capture local characteristics of the surface? (See Section 2.1.)

Target Selection: How to determine the targets $t_n \in \overline{\mathcal{R}}_S$ at which we will evaluate the KDE equation in (1)? (See Section 2.2.)

Density Estimation: How to choose the kernel function $\mathcal{K}$, how to set the bandwidth parameter matrix H in (1) and how to evaluate this equation in a computationally efficient manner? (See Section 2.3.)

2.1 Feature Design

In the following, we assume that the 3D object is embedded in a canonical reference frame of $\mathbb{R}^3$ whose origin

coincides with the center of mass of the object. The reference frame can be computed using the continuous PCA approach [4].

1. *Radial distance* R measures the distance of a surface point Q to the origin (centroid). Though not an effective shape feature all by itself; when coupled to other local surface features, it helps to manifest their distribution at varying radii.

2. *Radial direction* $\hat{\mathbf{R}}$ is a unit-norm vector $(\hat{R}_x, \hat{R}_y, \hat{R}_z)$ collinear with the ray traced from the origin to a surface point Q. This unit-norm vector is obviously scale-invariant.

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

4. *Radial-normal alignment* A is the absolute cosine of the angle between the radial and normal directions and is computed as $A = \left|\left\langle \hat{\mathbf{R}}, \hat{\mathbf{N}} \right\rangle\right| \in [0, 1]$. This feature measures crudely how the surface deviates locally from sphericity. For example, if the local surface approximates a spherical cap, then the radial and normal directions align, and the alignment A approaches unity.

5. *Tangent-plane distance* D stands for the absolute value of the distance between the tangent plane at a surface point Q and the origin. This scalar feature D is related to the radial distance R by $D = R.A$.

6. *Shape index* SI provides a local categorization of the shape into primitive forms such as spherical cap and cup, rut, ridge, trough, or saddle. In the present work, we consider the following parameterization

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

where κ_1 and κ_2 are the principal curvatures at the surface point. SI is confined within the range $[0, 1]$ and not defined when $\kappa_1 = \kappa_1 = 0$ (planar patch).

Our previous work [1, 2] has shown that the density-based framework is more effective when we consider the joining of the local features described above, to obtain even higher dimensional features. In the present work, we consider three such instances:

Radial feature S_1 is obtained by augmenting the scale-invariant radial direction vector $\hat{\mathbf{R}}$ with the rotation-invariant radial distance R. The resulting 4-tuple $S_1 \triangleq (R, \hat{R}_x, \hat{R}_y, \hat{R}_z)$ can be viewed as an alternative to Cartesian coordinate representation of the surface point. In this parameterization, however, the distance and direction information are decoupled. With this decoupling, the range of individual features can be determined independently. In fact, $\hat{\mathbf{R}}$ lies on the unit 2-sphere, and the scalar R lies on the interval $(0, r_{max})$, where r_{max} depends on the size of the surface.

Tangent plane feature S_2 is obtained by joining the tangent plane distance D with the normal direction $\hat{\mathbf{N}}$, providing a 4-dimensional vector $S_2 \triangleq (D, \hat{N}_x, \hat{N}_y, \hat{N}_z)$ that corresponds to the representation of the local tangent plane. As in the radial case, this representation also separates the distance and direction information concerning the tangent plane.

Finally, we define *a third feature* $S_3 \triangleq (R, A, SI)$, which aims at encoding the interaction between the radial and normal directions through the alignment feature A and at adding further local surface information through the shape index SI. Again the radial distance R augments the two-tuple (A, SI) to capture the characteristics of the shape at different distances from the center of the object.

In our runs, we will use the discretized pdfs of all these three features as descriptors and combine their individual discrimination capabilities (see Section 3).

2.2 Target Selection

The targets sets for our local features occur as the Cartesian products of their individual constituents. For instance, the S_1-feature is composed of a scalar feature $R \in (0, r_{max})$ and a unit-norm 3-vector $\hat{\mathbf{R}} \in \mathcal{S}^2$. Accordingly to determine the target set $\overline{\mathcal{R}}_{S_1}$, we first uniformly sample the interval $(0, r_{max})$ to obtain a distance set $\overline{\mathcal{R}}_R$, then partition the unit 2-sphere using the octahedron subdivision scheme described in [1, 6] to obtain a direction set $\overline{\mathcal{R}}_{\hat{\mathbf{R}}}$, and finally take their Cartesian products to obtain $\overline{\mathcal{R}}_{S_1} = \overline{\mathcal{R}}_R \times \overline{\mathcal{R}}_{\hat{\mathbf{R}}}$. Note that r_{max} depends on the type of scale normalization applied to the object. The target set for $S_2 = (D, \hat{N}_x, \hat{N}_y, \hat{N}_z)$ can be obtained likewise. Finally the target set for S_3 is given by $\overline{\mathcal{R}}_{S_3} = \overline{\mathcal{R}}_R \times \overline{\mathcal{R}}_A \times \overline{\mathcal{R}}_{SI}$, where both $\overline{\mathcal{R}}_A$ and $\overline{\mathcal{R}}_{SI}$ are uniformly sampled versions of the interval $[0, 1]$ since both A and SI share the same unit-interval as range.

2.3 Density Estimation

It is known that the particular functional form of the kernel does not significantly affect the accuracy of KDE [3]. In our scheme, we choose the Gaussian kernel since there exists a fast algorithm, the fast Gauss transform (FGT) [5], to rapidly evaluate large KDE sums in $O(K + N)$ instead of $O(KN)$-complexity of direct evaluation, where K is the number of sources and N is the number of targets.

The setting of the bandwidth matrix H has been shown to be critical for accurate density estimation [3], which in turn affects shape discrimination and retrieval performance [1]. The optimal bandwidth for KDE depends on the unknown density itself [3], making the appropriate choice of the bandwidth parameter a challenging problem. Guided by the results of our previous work in [1], we set the bandwidth parameter by averaging all object-level bandwidths given by Scott's rule [3] or by averaging the covariance matrix of the observations over the objects.

3 Experiments

Prior to descriptor computation, all models have been normalized so that descriptors are translation, rotation, flipping and scale invariant. For translation invariance, the object's center of mass is considered as the origin of the coordinate frame. For rotation and flipping invariance, we applied the continuous PCA algorithm [4]. For isotropic scale invariance, we calculate a scale factor so that the average point-to-origin distance is unity.

In each of the three runs that we have submitted to the SHREC'07 Watertight Models Track, we have used three different density-based descriptors: $S_1 = (R, \hat{R}_x, \hat{R}_y, \hat{R}_z)$, $S_2 \triangleq (D, \hat{N}_x, \hat{N}_y, \hat{N}_z)$, and $S_3 = (R, A, SI)$. Again in each of the runs, we have calculated the bandwidth matrix, required by the KDE, by averaging the covariance matrices of the observations over the meshes. We have used the l^1-metric to obtain three distance measures (each of which corresponds to one of S_1, S_2, and S_3) that we sum up to obtain a final dissimilarity value between each pair of objects. The runs differ in the following aspects:

Run 1. The mesh is reoriented so that the angle between radial and normal directions at a surface point is always acute.

Run 2. The orientation of the mesh is kept as given by the list of triangles and the *radial-normal alignment* feature is defined *with* the absolute value, i.e., $A = \left|\left\langle \hat{\mathbf{R}}, \hat{\mathbf{N}} \right\rangle\right| \in [0, 1]$.

Run 3. The orientation of the mesh is kept as given by the list of triangles and the *radial-normal alignment* feature is defined *without* the absolute value, i.e., $A = \left\langle \hat{\mathbf{R}}, \hat{\mathbf{N}} \right\rangle \in [-1, 1]$.

The DCG performance of these runs on the SHREC'07 Watertight Database are 85.0%, 84.0%, and 84.9% respectively. As we have pointed out earlier in this section, these results have been obtained using the l^1-metric. However, as we explain in the sequel, the discrimination ability of S_1- and S_2-descriptors can be further increased by exploiting a certain property of the directional features $\hat{\mathbf{R}}$ and $\hat{\mathbf{N}}$ involved in these descriptors respectively. Since these features are explicity parameterized by Cartesian coordinates, i.e., $\hat{\mathbf{R}} = (\hat{R}_x, \hat{R}_y, \hat{R}_z)$ and $\hat{\mathbf{N}} = (\hat{N}_x, \hat{N}_y, \hat{N}_z)$, an error in the labeling or the sign of x-,y- or z-axes alters the resulting pdf completely, leading to an unreliable descriptor. PCA-based normalization methods usually assign the axes based on the eigenvalue of the covariance matrix, and if two eigenvalues are close to each other there remains an ambiguity about which is the correct one. Assigning a sign to an axis is also problematic.

A matching scheme to circumvent discrimination errors to such undesirable situations can be implemented by considering all possible configurations of the Cartesian coordinate system. The first axis can be labeled as one of 6 the elements of the set $\{x+, x-, y+, y-, z+, z-\}$, let this be $x+$; the second axis can be labeled as one of the 4 elements of the set $\{y+, y-, z+, z-\}$, let this be $y+$; finally the remaining axis can be labeled as either $z+$ or $z-$. Clearly, there are $6 \times 4 \times 2 = 48$ such configurations. A naive way to achieve invariance against mislabelings is to compute, for a given mesh, 48 different descriptors each of which corresponding to a fixed coordinate labeling. The dissimilarity between two objects can then be calculated as the minimum of 48 distance values obtained by holding one descriptor fixed and alternating the other over the 48 configurations. Such a naive approach is computationally and memory-wise impractical.

Hopefully, since the target sets of the $\hat{\mathbf{R}}$- and $\hat{\mathbf{N}}$-features arise from a uniform sampling of the unit 2-sphere by octahedron subdivision [6], a descriptor corresponding to a certain labeling of the coordinate system can be derived from one another by just permuting the entries appropriately. To give an example, let $t_n = (t_{n,x}, t_{n,y}, t_{n,z})$ be a fixed target in the target set $\overline{\mathcal{R}}_{\hat{\mathbf{R}}} \in \mathcal{S}^2$ (or $\overline{\mathcal{R}}_{\hat{\mathbf{N}}} \in \mathcal{S}^2$ identically). Octahedron subdivision ensures that, for instance, $t_{n'} = (t_{n,y}, t_{n,x}, t_{n,z})$ is also in the target set (actually, all the 48 possibilities are in the target set). Thus, evaluating 48 distance values between two descriptors as described above, is just a matter of holding one descriptor fixed and permuting the entries of the other appropriately. The final dissimilarity value is simply the minimum among these 48 distance values. This metric adds a factor of 48 to the overall matching complexity, way smarter then calculating and storing 48 descriptors each of which corresponding to a different labeling of the coordinate system.

The improvements gained by the matching scheme are tabulated in Table 1 for each of the runs. In this table, we provide DCG figures corresponding to single descriptors S_1 and S_2 as well. Observe that, in all runs/cases, the improved matching scheme does indeed improve the discrimination as measured by the DCG gain, percent increase in DCG. It appears that S_1 benefits more from this improvement than S_2, DCG gains being 4.8% and 2.8% respectively. Enriched by the S_3-descriptor (which is invariant to all affine transformations but scale through the R-component), the overall

Table 1. DCG Performance of the Improved Matching Scheme

Run	Desc.	l^1	Imp. l^1	Gain (%)
1	S_1	74.4	78.0	4.8
	S_2	80.3	82.6	2.9
	$S_1 + S_2$	80.9	84.2	4.1
	$S_1 + S_2 + S_3$	85.0	86.7	2
2	S_1	73.2	76.8	4.9
	S_2	78.7	81.0	2.8
	$S_1 + S_2$	80.1	83.1	3.7
	$S_1 + S_2 + S_3$	84.0	85.4	1.7
3	S_1	73.3	76.8	4.8
	S_2	78.9	81.1	2.8
	$S_1 + S_2$	80.1	83.1	3.7
	$S_1 + S_2 + S_3$	84.9	86.3	1.6

descriptor in Run 1 achieves a DCG of 86.7% on the Watertight Models database.

It's also instructive to take a classwise look to the DCG performance in order to see what kind of discrimination can be achieved by the density-based framework on a database where classification semantics are topologically-guided (see Figure 1 in the main paper in this proceedings). In Table 2, we display classwise averaged DCGs and their standard deviations. From this table, we observe that the density-based descriptor performs nearly perfectly for the "Ant" and "Plier" classes with DCGs of 99.3% and 99.2% respectively. Worst DCG is obtained for the "Vase" class (DCG=60.1%) with a rather high standard deviation of 12.5%, indicating an inhomogeneity of the shapes within the class. The latter fact corroborates with the visual investigation of Figure 1 in the the main paper in this proceedings: the "Vase" instances differ significantly in geometry and appearence.

Table 2. Classwise DCG Performance of the Density-Based Framework

Class	Avg. DCG (%)	St. Dev. DCG (%)
Ant	99.3	1.1
Armadillo	93.8	10.5
Bearing	74.8	14.9
Bird	73.0	11.5
Bust	82.7	9.9
Chair	98.3	1.7
Cup	85.8	16.3
Fish	94.7	6.7
Four-legged	93.2	10.3
Glasses	88.9	11.1
Hand	79.3	13.0
Human	85.2	11.6
Mechanic	95.9	10.8
Octopus	70.0	14.0
Plane	93.1	14.5
Plier	99.2	1.2
Spring	80.0	17.1
Table	84.4	21.6
Teddy	98.7	1.8
Vase	61.4	12.5

4 Conclusion

Discrimination performance of density-based descriptors have already been established in [1, 2] through experiments on two different databases: Princeton Shape Benchmark and Sculpteur (see references therein for these databases). In the present paper, we have experimentally demonstrated that the density-based framework provides adequate discrimination also on the SHREC'07 Watertight database. The density-based description framework being geometrical by foundation, it's comforting to see that it enables effective retrieval on a database where shape equivalences are topologically-induced. We also remark that our framework is computationally very efficient [1, 2] thanks to the fast Gauss transform [5], it's not demanding in terms of mesh quality and degeneracies, and is able to support multiple 3D modalities be they 3D point cloud, mesh, or parametric surfaces. Our research on effective 3D retrieval continues with score fusion and similarity learning, guided by statistical techniques, which are able to incorporate domain-specific classification semantics to the matching scheme.

References

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

[2] C. B. Akgül, B. Sankur, Y. Yemez, and F. Schmitt. Multivariate density-based 3D shape descriptors. In *Shape Modeling International (SMI '07)*, submitted 2007.

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

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

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

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