Learning Distinctive Local Object Characteristics for 3D Shape Retrieval

Raoul Wessel[1], Rafael Baranowski[2], Reinhard Klein[3]

Computer Graphics Group
University of Bonn
Römerstraße 164, 53117 Bonn, Germany
Email: {wesselr[1], baranows[2], rk[3]}@cs.uni-bonn.de

Abstract

While supervised learning approaches for 3D shape retrieval have been successfully used to incorporate human knowledge about object classes based on global shape features, the incorporation of local features still remains a difficult task. First, it is not obvious how to measure the similarity between two objects each represented by a set of local features, and second, it is not clear how to choose local feature scales such that they are most distinctive. In this paper, we tackle both of these problems and present a supervised learning approach that uses arbitrary local features for 3D shape retrieval. It avoids the problem of establishing feature correspondences and automatically detects discriminating feature scales. Our experiments on the Princeton Shape Benchmark show that our method is superior to state-of-the-art shape retrieval techniques.

1 Introduction

Advances in geometrical modeling, 3D acquisition and graphics hardware have led to an ever increasing use of 3D models in various fields like entertainment, science, engineering or architecture. In spite of these advances, the generation of 3D content still remains a time- and cost-intensive task. Content-based 3D shape retrieval has therefore become a more and more important challenge in order to reuse the increasing number of existing 3D models.

The first methods developed for 3D shape retrieval are based on global shape representations like Spherical Harmonics descriptors [21], Zernike moments descriptors [15] or view-based descriptors [25, 2, 13], allowing fast and easy object comparison by computing the distance between two global descriptors given as vectors. As the affiliation of many 3D objects to a certain category is not solely defined by global shape attributes, local features have been introduced. The results presented in [14, 5], and especially in [18] indicate that local features are able to further enhance retrieval performance. Additionally, to bridge the semantic gap between low level geometric descriptors and high level user intended object categories, supervised learning approaches have been successfully used to improve global and local feature based approaches [7, 22, 4, 1].

Summarizing the results of recent work in 3D shape retrieval, it has become obvious that *"...no single descriptor is capable of providing fine grain discrimination required by prospective 3D search engines"*[1]. Additionally, the use of local features leads to improved retrieval results. Furthermore, the incorporation of supervised learning methods increases the performance of shape retrieval algorithms. Therefore, in this work we follow the general approach of unsupervised learning based on local features of different types that are combined in a single descriptor.

Looking at approaches relying on local features, there are mainly two different ways to measure the similarity of two sets of local descriptors. The first one relies on determining descriptor correspondences and then measuring the distance between corresponding descriptors [4, 17]. The second one uses histograms computed over the local descriptor distribution for object comparison [18, 11]. Both of these methods face drawbacks. Establishing feature correspondences usually involves the use of not easy to determine thresholds on descriptor similarity, position, orientation and spatial arrangement. Additionally, it can become quite time-consuming due to combinatorial reasons. Histogram-based approaches, although allowing easy comparison of

two objects using e.g. Kullback-Leibler divergence, suffer from the drawback that certain highly discriminating local features might have a relative low impact due to the descriptor agglomeration in the histogram.

Another problem arising from the use of local features is the impact of scale. A priori it is not obvious *how local* a feature descriptor should be in order to be most distinctive. Common approaches either rely on the usage of a single fixed scale [14], a combination of several fixed scales [22, 4], or on the detection of a built-in feature scale [18, 5]. Most of these approaches face the problem that accidentally choosing a less distinctive scale might lead to decreased retrieval performance.

To overcome these drawbacks we suggest the new *Class Distribution Descriptor* (CDD). The CDD is a meta-descriptor that transforms geometric features like Spherical Harmonics[21], Zernike moments [15], or spin-images [9] into a representation stating how characteristic this feature is for certain object classes like animals, humans, planes, etc. The transformation of local features into CDDs requires a preprocessing step conducted on a set of training features: Given a set of preclassified 3D objects, our algorithm first learns how discriminating certain local features are for a set of object classes. In the query step, local features are computed for an unclassified unknown query object and each of them is transformed into a CDD with respect to the knowledge acquired in the training phase. The CDD is uncoupled from the geometric feature it was derived from: First, it is independent of the feature type, and second it is rather independent of the actual local feature geometry. Both of these properties render CDDs appropriate for easy combination and comparison of different local features without the need to take care of position and spatial arrangement.

Note that in contrast to other approaches (e.g. [4, 10]) our method is not restricted to similarity measurements between unknown query objects and the objects contained in the training database, but it also allows comparison of two unknown query objects. Additionally, our CDD combination scheme enables the usage of multiple local feature scales and at the same time solves the above mentioned problems arising from less distinctive scales. In our experiments using different types of local shape features we show that our method is superior to common 3D shape retrieval approaches.

Summarizing the key contributions of our work, they are:

- A supervised learning approach allowing the easy use of arbitrary features for 3D shape retrieval by avoiding the problem of generating feature correspondences.
- Combination of arbitrary features of different scales with no drawbacks caused by accidentally chosen less distinctive scales.
- Experimental evaluation of our approach on the Princeton Shape Benchmark [21] showing that our method is superior to state-of-the-art 3D shape retrieval techniques.

2 Related Work

Amongst the large amount of literature on content-based 3D shape retrieval we concentrate on approaches that are most relevant to our work, that is local feature-based approaches and supervised learning based approaches.

2.1 Approaches using local features

Ohbuchi et al. [18] recently introduced salient local visual features extracted as SIFT-features [12] from rendered depth images. A *Bag-Of-Features* approach is used to build a histogram of these local features for each object. Two objects are compared by computing the Kullback-Leibler divergence between the associated histograms. Mitra et al. [14] locally characterize 3D shapes by probabilistic shape signatures based on spin-images [9]. Providing good results for automatic scan alignment, the retrieval performance of this method highly depends on the chosen local spin-image scale. In [5] local surface descriptors for salient triangle patches are introduced. Similarity between a query object and the database is determined according to the overlap of similar local surface descriptors. Novotni et al. [17] detect local salient points on a 3D voxel grid. For each salient point they compute a local Spherical Harmonics descriptor. Given two objects, they establish correspondences between the descriptors and determine the similarity with respect to the spatial arrangement of corresponding points and descriptor similarities.

2.2 Approaches using supervised learning

In [16] Novotni et al. use a Support Vector Machine (SVM) and Kernel-based methods to realize relevance feedback for 3D shape retrieval using global Zernike descriptors. In an iterative process, the user selects relevant and nonrelevant retrieved objects and is afterwards presented results that are improved by the learning process. Relevance feedback is also addressed in [1], where Akgül et al. present a linear score fusion approach. In the training step they use a SVM in order to learn how to optimally combine similarity values between two objects arising from different global shape descriptors. Hou et al. [7, 8] also make use of SVMs. In the training step they learn conditional object class probabilities given a set of global shape descriptors. For a query object, global shape descriptors are computed and the conditional class probabilities are predicted according to the learnt knowledge. Training objects belonging to the most likely class are returned as retrieval results. The approach presented in [10] combines elements of the supervised learning method presented in [8] with view-based Light Field Descriptors introduced in [2]. In [22], the appearance likelihoods of local Spherical Harmonics descriptors are mapped to their distinctiveness in a training step. Given a query object, distinctive descriptors are selected according to the learnt mapping. The similarity between two objects is then determined by the minimum Euclidean distance between all pairs of selected descriptors. Taking into account spatial relationships between local features, Funkhouser et al. further improve this method [4]. Amongst a large number of randomly chosen surface features, a small set of most distinctive ones is determined in a learning step for a database of training objects. Given a query object, randomly located surface features are computed and correspondences to the distinctive database features are established with respect to certain spatial constraints. While still showing to be superior to most other 3D shape retrieval approaches, this method as well as the above described [10] are restricted to searching similar objects only amongst the set of preclassified training objects.

3 Class Distribution Descriptors

Overview The CDD is an abstract meta-descriptor that can be computed for arbitrary descriptors. It states how characteristic the underlying descriptor is for object classes of a given set in terms of conditional probabilities. The idea is related to the approach presented in [8]. There, supervised learning methods are used to predict conditional object class probabilities given global descriptors computed for a query object. The preclassified objects in the database that belong to the most likely predicted class are returned as results. In our approach, we follow a different way enabling us to take advantage of local feature descriptors. Instead of using the conditional class probabilities as a classifier, we treat them as a new meta-descriptor: Given a set of object classes $\mathcal{C} = \{c_1, ..., c_n\}$, the Class Distribution Descriptor $cdd(d_i)$ of an arbitrary local descriptor $d_i \in \mathbb{R}^k$ is the result of a transformation Φ mapping d_i into a space of conditional probabilities:

$$cdd(d_i) = \Phi(d_i) = \begin{pmatrix} p(c_1|d_i) \\ \vdots \\ p(c_n|d_i) \end{pmatrix}. \qquad (1)$$

To point out the advantage of this representation over an underlying geometric descriptor like Spherical Harmonics or Zernike moments, let us consider a 3D model of an airplane, for which two local descriptors were computed, one approximately located at the tail and one in the cockpit. Although the geometry at these locations largely differs, the two resulting CDDs will probably look similar, as both, the tail as well as the cockpit, are highly significant for an airplane. This example also demonstrates how the CDDs are uncoupled from local geometry and feature localization, rendering feature correspondence determination unnecessary.

3.1 Learning Conditional Class Probabilities

Computing CDDs for local descriptors d_i of unknown objects requires the prediction of conditional class probabilities $p(c_j|d_i)$ with respect to a set of given classes $\mathcal{C} = \{c_1, ..., c_n\}$. We solve this problem using a supervised learning approach presented in [19] which was originally proposed for classification of hand-printed Japanese characters. We will briefly describe this method in the following.

Consider two object classes c_a and c_b, and two sets of local training descriptors D_a and D_b

that were computed on objects from the according classes. In a training step, a classification method referred to as Nonlinear Kernel Discriminant Analysis (NKDA) is used to determine discriminant functions $g_{ab}(d)$ for every $n(n-1)/2$ pairs of classes c_a and c_b, learning whether the descriptor $d \in D_a \cup D_b$ indicates an object of class c_a or c_b. Let N be the total number of training descriptors and consider a kernel function $k(d_i, d_j)$ satisfying the Mercer condition [20]. Then, the kernel matrix K of size $N \times N$ built from the local training descriptors reads

$$K_{ij} = k(d_i, d_j), \tag{2}$$

with $d_i, d_j \in D_a \cup D_b$. Let now y^{ab} be the class label vector of size N with entries $y_i^{ab} = 1$ if $d_i \in D_a$ and $y_i^{ab} = -1$ if $d_i \in D_b$. Then the discriminant vector $\hat{\alpha}^{ab}$ can be computed as

$$\hat{\alpha}^{ab} = K^{-1} y^{ab}, \tag{3}$$

and the discriminating function g_{ab} reads

$$g_{ab}(d) = \sum_{i=1}^{N} k(d, d_i) \hat{\alpha}_i^{ab}. \tag{4}$$

Note that in contrast to most SVMs, NKDA produces probabilistic class label outputs due to the underlying Bayesian model. This will be important for the construction of conditional class probabilities using the discriminant functions. For further explanations and details we refer to [19].

So far, the $n(n-1)/2$ learnt discriminant functions are able to predict pairwise class probability estimates for unknown local descriptors. To predict the conditional probabilities $p(c_i|d_i)$ given the discriminant functions $g_{ab}(d_i)$, we use a technique known as *Pairwise Coupling* [6]. It is an iterative method that is designed to couple $n(n-1)/2$ pairwise class probability estimates into a joint estimate for all n classes. For further details on this method especially when used together with the NKDA we refer to [19] and [6].

3.2 Combining CDDs

Once CDDs for all local descriptors have been computed they need to be aggregated into one single CDD in order to allow object comparison. Combinations of probabilistic classifier outputs have been widely discussed in literature, see e.g. [24, 23, 8]. Supposing the local features to be statistically independent and adopting the Bayesian point of view leads to a widely used combination method known as product rule:

$$\begin{aligned} cdd(d_1, ..., d_l) &= \begin{pmatrix} p(c_1|d_1, ..., d_l) \\ \vdots \\ p(c_n|d_1, ..., d_l) \end{pmatrix} \\ &= \begin{pmatrix} \frac{\prod_{i=1}^{l} p(c_1|d_i)}{\sum_{j=1}^{n} \prod_{i=1}^{l} p(c_j|d_i)} \\ \vdots \\ \frac{\prod_{i=1}^{l} p(c_n|d_i)}{\sum_{j=1}^{n} \prod_{i=1}^{l} p(c_j|d_i)} \end{pmatrix} \\ &= \frac{\prod_{i=1}^{l} cdd(d_i)}{\sum_{j=1}^{n} \prod_{i=1}^{l} cdd(d_i)}, \end{aligned}$$

in case of equal a priori class probabilities. As no other combination method seems to clearly outperform this strategy [23, 8] we compute combinations of CDDs in this simple way.

Choosing the product rule for CDD combination additionally offers an essential advantage over methods like probability averaging, which is another commonly used combination method. It automatically favors distinctive CDDs (those containing low entropy in the object class distribution) over less distinctive CDDs (those containing high entropy in the object class distribution). Consider the following example: Given two different object classes, suppose one computes l local descriptors on an unknown object among which only one captures a distinctive characteristic feature. The remaining $l-1$ CDDs are not able to distinguish between the two classes, their entries read $p(c_1|d_i) = p(c_2|d_i) = 0.5, i = 2, .., l$. The first descriptor's entries read $p(c_1|d_1) = 0.7$ and $p(c_2|d_1) = 0.3$. Averaging the CDDs would then lead to the following combined CDD:

$$cdd_{avg}(d_1, ..., d_l) = \begin{pmatrix} \frac{0.7+(l-1)\cdot 0.5}{l} \\ \frac{0.3+(l-1)\cdot 0.5}{l} \end{pmatrix}.$$

For large l, the influence of the one distinctive descriptor d_1 vanishes more and more. In contrast, combination by the product rule delivers

$$cdd_{prod}(d_1, ..., d_l) = \begin{pmatrix} \frac{0.7\cdot 0.5^{l-1}}{(0.7+0.3)\cdot 0.5^{l-1}} \\ \frac{0.3\cdot 0.5^{l-1}}{(0.7+0.3)\cdot 0.5^{l-1}} \end{pmatrix},$$

preserving the information provided by the distinctive descriptor. By that, less distinctive descriptors that were for example computed on a non-discriminating feature scale do not decrease the overall distinctiveness of the combined CDD. This allows to include multiple scales in feature computation amongst which one or several distinctive ones should be found.

3.3 Computing CDD representations of 3D objects

With the learning algorithm and the combination rule at hand we can now compute CDD representations for an arbitrary 3D object. Consider a set of object training classes $\mathcal{C} = \{c_1, ..., c_n\}$, a set of local descriptor types $DT = \{type_1, .., type_t\}$ and sets of descriptor scales $DS(type_i) = \{scale_1(type_i), .., scale_s(type_i)\}$ associated with each descriptor type. We first compute a number of local descriptors of any type on all associated scales $DS(type_1), .., DS(type_t)$ for every training object. Second, for each descriptor type and scale combination we learn $n(n-1)/2$ discriminant vectors $\hat{\alpha}^{ab}(type_i, scale_j(type_i))$ (see Equation 3) regarding every pair of classes c_a and c_b like explained in 3.1.

Given an unknown query object, we repeat the descriptor computation described above. For each local descriptor d we then evaluate the according discriminant functions $g_{ab}(d)$ (see Equation 4) and derive the CDD representation using pairwise coupling. We then combine all CDD descriptors of the particular object using the product rule described in Section 3.2.

3.4 Comparison of CDDs

To compute the similarity of two 3D objects, we must define a distance measure between the two according combined CDDs. As CDDs represent a discrete probability distribution over the object classes, the natural similarity measure between them is given by the Kullback-Leibler (KL) divergence, also known as relative entropy [3]. Let $p(c)$ and $q(c)$ denote two probability distributions over the set of object classes $\mathcal{C}$, then the KL-divergence reads:

$$D_{KL}(p,q) = \sum_{a=1}^{n} p(c_a) \log \frac{p(c_a)}{q(c_a)}. \quad (5)$$

As we are interested in a symmetric object similarity measure, we use the modified version of KL-divergence in our experiments :

$$D_{KLsym}(p,q) = D_{KL}(p,q) + D_{KL}(q,p). \quad (6)$$

4 Results

In this Section we will first give an overview of our experimental setup describing the 3D objects and descriptors we use. We will then present our results and compare them to the state-of-the-art method described in [4]. We will further discuss the influence of descriptor types and feature scales.

4.1 Experimental Setup

Dataset For our experiments we used a subset of the 3D objects contained in the Princeton Shape Benchmark (PSB). Originally, the PSB is separated into two subsets. The training set contains 907 objects divided into 90 classes and the test set contains 907 objects divided into 92 classes. Merging both subsets leads to 1814 objects in 161 classes. To ensure that the learnt discriminant functions (see 3.1) achieve appropriate generalization performance, we restricted ourselves to those PSB classes from the merged set containing at least 20 objects. This resulted in 739 objects divided into 21 classes, see Table 1. We divided this set into a training set and

class	objects	class	objects
biplane	28	commercial airplane	21
fighter jet	100	helicopter	35
enterprise	22	human	100
like		standing	
human arms	41	sword	31
out			
face	33	head	32
two story	21	city	20
home			
dining chair	22	shelves	26
table rectangular	51	handgun	20
vase	22	potted plant	51
tree barren	22	ship	21
sedan	20		

Table 1: PSB classes used in our experiments.

a test set. For the training set, we randomly selected 16 objects of each class, resulting in a total number of 336 training objects. The remaining

403 objects were put into the test set. For all experiments involving our CDDs we used the training set for learning the discriminant functions. Retrieval was done exclusively on the test set. Three methods were used for comparison, global Spherical Harmonics descriptors, global Zernike descriptors and partial matching with priority driven search [4]. The two methods using global shape descriptors do not require any training and were only evaluated on the test set. Partial matching requires a leave-one-out classification. Query results can only be found amongst the training objects, so for each query object, the remaining 402 models of the test set were used as training data.

Descriptors For the experiments involving CDDs we used three different types of local descriptors: Spherical Harmonics [21], Zernike moments [15] and spin-images [9]. Descriptor locations were randomly selected on the underlying triangle mesh with respect to the triangle areas. All triangle meshes were normalized to their bounding boxes.

To compute Spherical Harmonics and Zernike moments, the triangle meshes were transformed into voxel grids of size $128 \times 128 \times 128$ in a preprocessing step. Spherical Harmonics and Zernike moments were computed on the voxel grid around all randomly selected points. The resolution for the Spherical Harmonics descriptors was 16 shells, each of it represented by 8 coefficients. For each Zernike moment we used 156 coefficients. To generate spin-images, the meshes were transformed into point clouds, each of it containing 250000 points per surface unit, leading to approximately 100000 to 700000 points per object. Spin-images were computed on this point cloud around all randomly selected points. The required normals were derived from the underlying triangles. Note that we did not use oriented normals to build the spin-image, as the PSB objects do not provide consistent orientation in general. The spin-image resolution was set to 16×16 bins. All feature scales mentioned in the next section describe the radius of the computed local descriptor. All scales are measured with respect to the radius of the underlying object. We computed 64 local shape descriptors of every type and scale.

For comparison to other methods and to show the impact of local CDDs, we also computed global Spherical Harmonics and Zernike moments located at the object centers. The resolution for both of them was the same as for the according local descriptors.

Training and Retrieval For learning the conditional class probabilities and for evaluating the resulting discriminant functions (see Equations 3 and 4), we used a Gaussian kernel

$$k(d_i, d_j) = \exp\left(-\frac{|d_i - d_j|^2}{2\sigma^2}\right).$$

The optimal kernel width σ was determined for every of the $n(n-1)/2$ NKDA classifiers separately using 8-fold cross validation, each time leaving out 2 of the 16 training models of both object classes.

4.2 Evaluation

Comparison to other methods We compared our method to three other shape retrieval approaches. On the one hand, classic approaches relying on global features without any supervised learning are represented by global Spherical Harmonics (GSH) and global Zernike moments (GZM). Partial matching with priority driven search (PDS) on the other hand represents a state-of-the-art method and is mostly related to our approach as it incorporates local descriptors and supervised learning. As can be seen from the precision-recall plot in Figure 1, PDS and our method perform at almost the same precision for the first recall value of 0.05. For larger recall values, our approach clearly outperforms the PDS as well as the other methods. Table 2 shows the performance of all four methods regarding additional quality criteria for shape retrieval. The 1-NN value describes the performance achieved by a nearest neighbor classifier. The tiers denote the fraction of objects belonging to the class of the query object among the top T results. For a class containing n objects, $T = n-1$ for the 1-Tier and $T = 2(n-1)$ for the 2-Tier. Average discounted cumulative gain (DCG) gives an impression of how the overall retrieval would be viewed by a human. For further explanations on DCG we refer to [21]. The retrieval measures given in Table 2 additionally show the superior performance of our method.

Impact of local descriptors In order to show the importance of using local features we compare the performance of a classic global Zernike moment

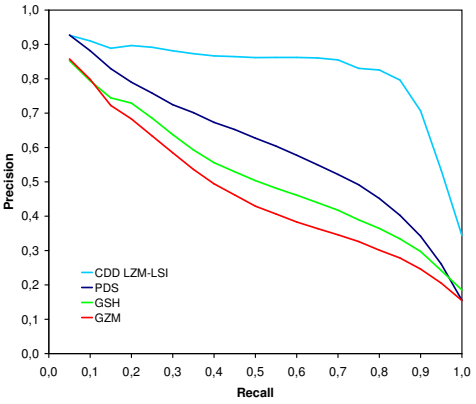

Figure 1: **Comparison to other methods.** The precision-recall plot shows a comparison of our approach (CDD LZM-LSI) using a combination of local Zernike moments descriptors and spin-images to three other shape retrieval methods. The state-of-the-art method *Partial Matching with Priority Driven Search* is denoted by PDS. Retrieval by one single global descriptor is represented by GSH (using Spherical Harmonics) and GZM (using Zernike moments).

Method	1-NN	1-Tier	2-Tier	DCG
CDD LZM-LSI	0.844	0.745	0.850	0.892
PDS	0.789	0.543	0.677	0.795
GSH	0.705	0.427	0.598	0.731
GZM	0.732	0.390	0.540	0.714

Table 2: Comparison of our approach (CDD LZM-LSI) to three other shape retrieval methods with respect to standard retrieval metrics. Our algorithm shows to be superior regarding all quality criteria.

(GZM) and a CDD derived from one global Zernike moment (CDD GZM) with our approach based on CDDs derived from local Zernike moments (CDD LZM). We include the CDD derived from the global descriptor in order to guarantee a fair comparison such that both, the local descriptors as well as the global one benefit from the supervised learning step. The results of the comparison are shown in Figure 2 and in Table 3. Although the performance of the global descriptor is definitely boosted by the CDD, it is still inferior to the CDDs derived from local descriptors.

Combination of different descriptor types To investigate the influence of descriptor type combination, we selected the best performing feature scales for every single descriptor type and com-

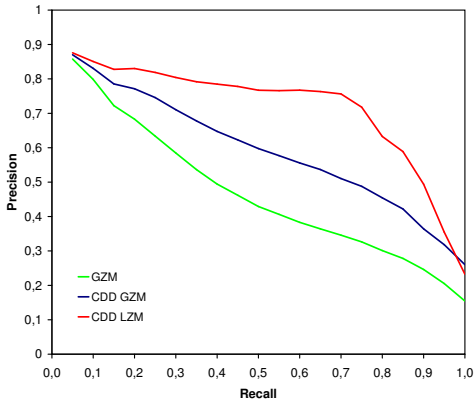

Figure 2: **Impact of local descriptors.** GZM denotes retrieval with one global Zernike moment. CDD GZM represents the same Zernike moment boosted by the CDD. CDD LZM shows retrieval results for 64 local Zernike moments at scale 1.0 using CDDs.

Method	1-NN	1-Tier	2-Tier	DCG
CDD LZM	0.757	0.636	0.767	0.834
CDD GZM	0.747	0.499	0.677	0.782
GZM	0.732	0.390	0.540	0.714

Table 3: Results of experiments regarding the impact of local descriptors. The table shows three different methods using CDDs based on local Zernike moments (CDD LZM) computed on scale 1.0, CDDs based on one global Zernike moment (CDD GZM) and one global Zernike moment (GZM).

bined the according CDDs with each other, see Table 4. It shows that the combination of Zernike moments and spin-images (LZM-LSI) and the combination of Spherical Harmonics and spin-images (LSH-LSI) perform best and further improve the retrieval performance of each single descriptor type (see Figure 3 for the LZM-LSI combination). Although not reaching the quality that is achieved if spin-images are involved, the combination of Spherical Harmonics and Zernike moments (LZM-LSH) is superior to using these descriptors alone. The combination of all three descriptor types (LSH-LSI-LZM) does not improve the performance any further.

Selection of the right scale Figure 4 and Table 5 show retrieval results using CDDs from Spherical Harmonics computed on three different scales (LSH 0.25, LSH 0.5, LSH 1.0). It also shows the retrieval results that were achieved by combining the resulting three CDDs into a single one (LSH com-

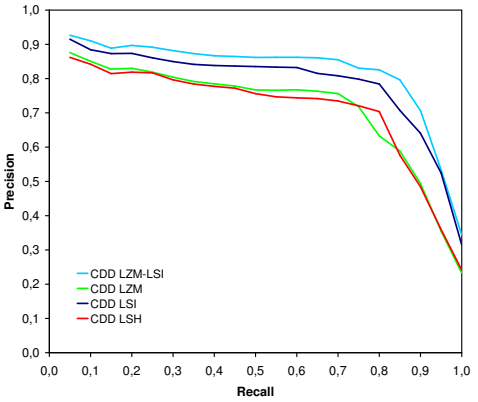

Figure 3: **Combining different descriptor types.** The combination of CDDs derived from Zernike moments and CDDs derived from spin-images futher improves the retrieval results.

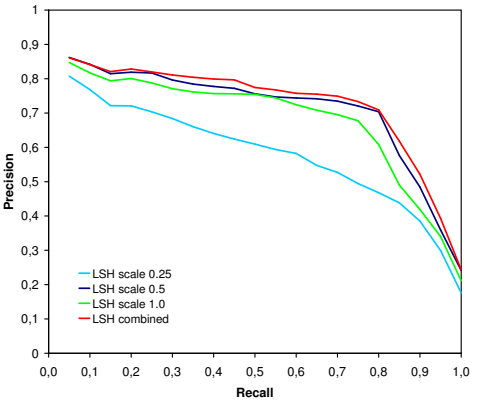

Figure 4: **Selecting distinctive feature scales.** The precision-recall plot shows CDDs derived from local Spherical Harmonics descriptors computed at three different scales. Combination of the CDDs using the product rule leads to a result close to that achieved at the best scale.

CDD Method	1-NN	1-Tier	2-Tier	DCG
LZM 0.5	0.757	0.636	0.767	0.834
LSI 2.0	0.824	0.716	0.810	0.880
LSH 0.5	0.742	0.633	0.746	0.835
LZM-LSI	0.844	0.745	0.850	0.892
LZM-LSH	0.769	0.678	0.782	0.856
LSH-LSI	0.861	0.744	0.845	0.896
LSH-LSI-LZM	0.821	0.751	0.841	0.893

Table 4: Results of experiments regarding combinations of CDDs derived from different types of local descriptors. The value behind the descriptor name denotes the scale on which it was computed.

bined). Like described in Section 3, the results show that choosing the product rule as combination strategy automatically favors distinctive feature. Without knowing which feature scale is most distinctive, combining leads to a result that is even a bit better than the one achieved by the optimal scale. This is due to the fact that although the overall performance of the scales 0.25 and 1.0 is worse than that of scale 0.5, these scales might be more distinctive for certain objects and thereby improve the performance if combined.

CDD Method	1-NN	1-Tier	2-Tier	DCG
LSH 0.25	0.630	0.464	0.604	0.747
LSH 0.5	0.742	0.633	0.746	0.835
LSH 1.0	0.737	0.587	0.712	0.818
LSH combined	0.777	0.655	0.771	0.847

Table 5: Retrieval results using CDDs derived from local Spherical Harmonics descriptors computed at three different scales.

Timings In Table 6 we provide information about the time consumption of our approach. All experiments were run on an AMD Athlon^{TM}X2 Dual Core with 2.21 GHz and 2 GB RAM using Windows XP operating system (32 Bit). Parts of the training and the CDD computation were accelerated using a NVIDIA® GeForce®8800. The timings for preprocessing include the voxel grid computation (for Spherical Harmonics and Zernike moments) and the point cloud generation (for spin-images), respectively, as well as the computation of 64 local descriptors of a certain type. The timings for training and CDD computation are computed with respect to objects represented by 64 descriptors of one type on the scales shown in Table 4. The time required to query an unknown object given in a mesh representation is the sum of required preprocessing time and the CDD computation time. So far, the query time for one object is dominated by the descriptor computation during the preprocessing step. Note that the focus of our work was rather on retrieval quality, we did not optimize the descriptor computation towards speed.

5 Conclusion

In this work we presented a supervised learning approach for 3D shape retrieval that can make use of arbitrary local features by computing the newly introduced Class Distribution Descriptor. The CDD

	LSH	LZM	LSI
Preprocessing training set	58 min	234 min	174 min
Preprocessing test set	71 min	284 min	216 min
Preprocessing per object (average)	10.53 s	42.05 s	31.75 s
Training	45 min	46 min	50 min
CDD computation test set	25 min	26 min	27 min
CDD computation per object (average)	3.78 s	3.85 s	4.05 s
Query of one object	14.31 s	45.90 s	35.80 s

Table 6: **Timings.**

incorporates knowledge about a preclassified training set of 3D objects and is independent of the underlying feature it was derived from. Using CDDs, we were able to avoid the problem of establishing feature correspondences and we could automatically favor discriminating feature scales. Our experiments on the Princeton Shape Benchmark showed that our method is superior to state-of-the-art shape retrieval techniques.

Future work should include optimization towards speed in order to render our approach more suitable for online shape retrieval. We would also like to examine whether our approach can handle queries that represent only a certain part of an object. This would be of particular interest regarding laser range scans representing an object from one certain view.

Acknowledgments

This work was partially funded by the German Research Foundation (DFG) under grant GZ 554975(1) Oldenburg BIB 48 OLof 01-02 *PROBADO*. We would like to thank Roland Ruiters for his insightful feedback and support.

References

[1] C. Akgül, B. Sankur, Y. Yemez, and F. Schmitt. Similarity Score Fusion by Ranking Risk Minimization for 3D Object Retrieval. *Proceedings of the Eurographics Workshop on 3D Object Retrieval*, Crete, Greece, April 2008.

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

[3] T. M. Cover and J. A. Thomas. Elements of Information Theory (2nd edition), *John Wiley & Sons Inc*, 2006.

[4] T. Funkhouser and P. Shilane. Partial Matching of 3D Shapes with Priority-Driven Search *Symposium on Geometry Processing*, July 2006.

[5] R. Gal and D. Cohen-Or. Salient geometric features for partial shape matching and similarity, *ACM Transactions on Graphics*, 2006, 25,1,130-150.

[6] T. Hastie and R. Tibshirani. Classification by Pairwise Coupling, *Advances in Neural Information Processing Systems* The MIT Press, 1998, 10.

[7] S. Hou, K. Lou, and K. Ramani. SVM-based Semantic Clustering and Retrieval of A 3D Model Database, *Computer-Aided Design & Applications*, 2005, 2, 1-4.

[8] S. Hou and K. Ramani. Calligraphic Interfaces: Classifier combination for sketch-based 3D part retrieval. *Computers and Graphics*, 2007, 31,4,598-609.

[9] A. Johnson, Spin-Images: A Representation for 3-D Surface Matching, *Ph.D. Dissertation*, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA August 1997.

[10] H. Laga and M. Nakajima. A boosting approach to content-based 3D model retrieval, *GRAPHITE '07: Proceedings of the 5th international conference on Computer graphics and interactive techniques in Australia and Southeast Asia*, 2007, 227–234.

[11] Y. Liu, H. Zha and H. Qin. Shape Topics: A Compact Representation and New Algorithms for 3D Partial Shape Retrieval, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[12] D. G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints, *International Journal of Computer Vision*, 2004, 60, 2, 91-110.

[13] A. Makadia and K. Daniilidis Light Field Similarity for Model Retrieval, *SHREC 2006 3D Shape Retrieval Contest*, Technical Report,

Department of Information and Computing Sciences, Utrecht University.

[14] N. J. Mitra, L. Guibas, J. Giesen, and M. Pauly. Probabilistic Fingerprints for Shapes, *Symposium on Geometry Processing*, 2006.

[15] M. Novotni and R. Klein. 3D Zernike Descriptors for Content Based Shape Retrieval, *The 8th ACM Symposium on Solid Modeling and Applications*, June 2003.

[16] M. Novotni, G.-J. Park, R. Wessel, and R. Klein. Evaluation of Kernel Based Methods for Relevance Feedback in 3D Shape Retrieval, *The Fourth International Workshop on Content-Based Multimedia Indexing (CBMI'05)*, June 2005.

[17] M. Novotni and P. Degener and R. Klein. Correspondence Generation and Matching of 3D Shape Subparts, *Technical Report CG-2005-2, Universität Bonn*, June 2005.

[18] R. Ohbuchi, K. Osada, T. Furuya, and T. Banno. Salient local visual featuers for shape-based 3D model retrieval *Proceedings of Shape Modeling International (SMI) 2008*, June 4-6, Stony Brook, NY, USA.

[19] V. Roth and K. Tsuda. Pairwise Coupling for Machine Recognition of Hand-Printed Japanese Characters, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2001, 1, 1120-1125.

[20] B. Schölkopf and A. J. Smola. Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond, *MIT Press, Cambridge, MA, USA* 2001.

[21] P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser. The Princeton Shape Benchmark *Shape Modeling International, Genova, Italy*, June 2004.

[22] P. Shilane and T. Funkhouser. Selecting Distinctive 3D Shape Descriptors for Similarity Retrieval, *Shape Modeling International*, June 2006.

[23] D. M. J. Tax, M. van Breukelen, R. P. W. Duin, and J. Kittler. Combining multiple classifiers by averaging or by multiplying, *Pattern Recognition*, 2000, 33, 1475-1485.

[24] N. Ueda. Optimal Linear Combination of Neural Networks for Improving Classification Performance, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2000, 22, 207-215.

[25] D.V. Vranic. 3D model retrieval, *Ph.D. Dissertation*, University of Leipzig, Department of Computer Science, 2004.