Eurographics Workshop on 3D Object Retrieval (2010)
I. Pratikakis, M. Spagnuolo, T. Theoharis, and R. Veltkamp (Editors)

Learning the Compositional Structure of Man-Made Objects for 3D Shape Retrieval

R. Wessel and R. Klein

University of Bonn, Germany

Abstract

While approaches based on local features play a more and more important role for 3D shape retrieval, the problems of feature selection and similarity measurement between sets of local features still remain open tasks. Common algorithms usually measure the similarity between two such sets by either establishing feature correspondences or by using Bag-of-Features (BoF) approaches. While establishing correspondences often involves a lot of manually chosen thresholds, BoF approaches can hardly model the spatial structure of the underlying 3D object. In this paper focusing on retrieval of 3D models representing man-made objects, we try to tackle both of these problems. Exploiting the fact that man-made objects usually consist of a small set of certain shape primitives, we propose a feature selection technique that decomposes 3D point clouds into sections that can be represented by a plane, a sphere, a cylinder, a cone, or a torus. We then introduce a probabilistic framework for analyzing and learning the spatial arrangement of the detected shape primitives with respect to training objects belonging to certain categories. The knowledge acquired in this learning process allows for efficient retrieval and classification of new 3D objects. We finally evaluate our algorithm on the recently introduced 3D Architecture Shape Benchmark, which mainly consists of 3D models representing man-made objects.

Categories and Subject Descriptors (according to ACM CCS): H.3.1 [Information storage and retrieval]: Content Analysis and Indexing, I.3.m [Computer graphics]: Miscellaneous

1. Introduction

Driven by the necessity to ensure reusability of the large amount of available 3D models, 3D shape retrieval has gained more and more attention during recent years. While in the beginning the focus was merely on methods relying on global shape descriptors, approaches based on local features have become increasingly important. This is mainly due to fact that the geometric variation of certain object classes can hardly be described by only global shape properties. Apart from the question how local object features can be characterized by descriptors efficiently, there are two major ingredients for a local feature based shape retrieval algorithm, namely *feature selection* as well as computation of a *similarity measure* between two sets of local features.

Most feature selection methods are based on local geometric properties of the 3D object. The idea is to identify features as parts of the object that are *salient* in a geometric sense. Most approaches thereby focus on features that are robust to detect under object transformations like scaling, rotation, shearing, and articulation, see e.g. [NDK05, GCO06, OOFB08, HH09]. In this work, we will present a selection methods that is especially tailored to 3D models representing man-made objects. Due to common manufacturing processes, these objects mainly consist of building blocks that can be assembled from parts of certain shape primitives like planes, cylinders, spheres, cones and tori. This structure is the starting point for our feature selection. We use the algorithm presented in [SWK07] to decompose a 3D model into segments corresponding to shape primitives. For each segment, we compute a shape descriptor. Depending on the size of the underlying shape primitive, our algorithm produces features ranging from very local to rather global (see e.g. Figure 1). Our feature selection method is similar to the one presented in [FMA*09] using the mesh segmentation algorithm presented in [AFS06]. However, in contrast to this method our approach does not rely on an intact mesh connectivity which is often not available when dealing with real

DOI: 10.2312/3DOR/3DOR10/039-046

world data. Instead we only require a point cloud which can be easily obtained by densely sampling the underlying mesh. In addition to plane-, cylinder-, and sphere-like shapes which are recognized in [AFS06], the algorithm in [SWK07] supports cone- and tori-like shapes.

In contrast to shape retrieval approaches based on global descriptors where object similarity can be determined in a straight forward way by computing the distance between global descriptors, there is no such easy way for methods involving local features. Several algorithms for solving this problem have been proposed. *Bag-of-features* (BoF) based approaches map a set of local features into a single histogram, counting occurencies of certain codebook features, see e.g. [LZQ06, LGW08, OOFB08, TCF09]. Although it provides an elegant way of making local feature sets comparable, the approach faces a major drawback. In general, the spatial arrangement of local features is lost as soon as they are described by histograms, just as it happens when objects arranged in a certain order are put into a bag. In contrast, methods based on *correspondence computation* try to determine mappings between two local feature sets taking feature similarity as well as their spatial relationship into account, see e.g. [NDK05, SMS*04, FS06]. However, these approaches usually involve manual tuning of several pruning thresholds, rendering it hard to achieve good generalization results. In this work, we try to overcome the drawbacks of the above mentioned approaches. We propose a probabilitic framework for learning the compositional structure of 3D objects which is inspired by an approach to 2D image retrieval by Ommer et al. [OB07, OB09]. In contrast to common BoF-approaches it incorporates the relative position of single features as well as the spatial relationship of feature tuples. Using a supervised learning scheme, we overcome the shortcomings of methods based on correspondece computation, as we do not need to manually enforce cumbersome thresholds on descriptor similarity or spatial distances. We finally compare our new approach to the results achieved by [WBK08] on the 3D Architecture Benchmark [WBK09]. Sharing a similar supervised learning framework, this approach uses randomly selected object parts as features instead of features based on shape primitive and no spatial relationship are considered. Summarizing the key contributions of this work, they are:

- We introduce a new method for feature selection that is especially tailored to the domain of 3D models representing man-made objects.
- We propose a supervised learning framework for efficient similarity computation of local feature sets incorporating their spatial relationship.
- We evaluate our approach using the 3D Architecture Benchmark, discussing the impact of our feature localization technique as well as the proposed similarity computation algorithm.

2. Related Work

In this section, we will briefly review the related work on 3D shape retrieval. We focus on methods for feature selection as well as on methods for measuring the similarity of two 3D objects with respect to sets of local features as these two aspects are the most relevant ones for our approach.

2.1. Feature Selection

Randomly selected uniformly distributed features Probably the easiest way for feature selection is to randomly select uniformly distributed points on the object surface as feature centers. Feature radii are then usually determined according to a manually chosen value. Mitra et al. [MGGP06] locally characterize 3D shapes by probabilistic shape signatures based on spin-images [Joh97] computed at randomly selected uniformly distributed points on the object's surface. Providing good results for automatic scan alignment, the retrieval performance of this method highly depends on the chosen local spin-image scale. Wessel et al. [WBK08] also use random surface points as centers for local spin-images, Spherical Harmonics descriptors [KFR03] and Zernike moments descriptors [Nov03]. All descriptors are computed with respect to several manually defined radii. Uniformly distributed surface points also serve as a starting point for distinction-based feature selection in [SF06].

Geometry-based feature selection In [GCO06] local salient regions are detected as mesh patches providing a high curvature relative to the surrounding area. A region-growing approach is used to subsequently augment small salient patches to larger regions. Shalom et al. [SSSCO08] use the shape-diameter function for both, segmentation and part signature definition. Ohbuchi et al. [OOFB08, OF08, FO09] introduce salient local visual features extracted as SIFT-features [Low04] from rendered depth images. A generalization of the SIFT algorithm to three dimensions is presented in [NDK05]. In this work, Novotni et al. detect salient points on a 3D voxel grid as local extrema of the scale space Laplacian-of-Gaussian. For each detected salient point, they compute a local Spherical Harmonics descriptor. Further approaches based on scale spaces are presented in [LVJ05] and [ZHDQ08]. In [HH09], Hu et al. present an approach to detect local salient mesh regions using extrema in the Laplace-Beltrami spectral domain of the mesh rather than in the usual spatial domain, rendering this localization algorithm invariant to isometric mesh deformations. An approach closely related to our own is presented in [SWWK08]. Primitive shapes like planes, cylinders, etc. are detected in 3D laser range scans. In contrast to our approach, no local descriptors based on the point supports of the shapes are computed. The primitives are directly used as nodes in a graph-based algorithm searching for certain manually defined configurations of primitives, forming simple shapes like building roofs. A similar method restricted to the detection of configurations of planes is described in [VKH06].

Distinction-based feature selection Selection based on the retrieval performance of local features is introduced in [SF06] and used in [SF07, FS06]. Considering a set of preclassified training objects, a number of random surface points is sampled from all objects. For each of these points, a local Spherical Harmonics descriptor is computed characterizing the local surface geometry with respect to a certain radius. For each of these descriptors, it is determined how well they are suited for efficient object retrieval. For new unknown objects, again local Spherical Harmonics descriptors are computed around randomly sampled surface points. The knowledge acquired during the training step is used to predict the retrieval performance of these local descriptors and only the most distinctive ones are finally used for retrieval.

2.2. Similarity between two sets of local features

Establishing feature correspondences The idea behind this approach is to determine a mapping between two local feature sets taking feature similarity as well as their spatial relationship into account. While in [NDK05] object similarity is defined in terms of a thin-plate spline bending energy induced by determined pairwise feature correspondences, Funkhouser et al. [FS06] use a heuristic similarity measure involving Spherical Harmonics descriptor distances and similarity of spatial relationships. Focusing on recognition of small vehicles in point clouds from laser range scans, RANSAC based approach for the detection of small compatible feature sets are presented in [SMS*04] and [JH99]. Despite their abibility to include spatial relationships of local features into the object similarity measure, the methods mentioned so far require to manually define a lot of pruning thresholds on descriptor similarity and spatial consistency, rendering it hard to achieve good generalization results. Methods based on geometric hashing [WL88] are extremely popular in computer vision but have also been applied to 3D shape retrieval [GCO06]. Although this approach takes spatial relationships of features into account, it faces two major drawbacks. First, the memory consumption for storing the hash tables is rather high. Second, the degree of discretization of the transformation space and the Euclidean space at which high quality retrieval results can be achieved is rather hard to determine.

Bag-of-Features Methods based on the BoF paradigm have recently gained increasing attention in the 3D shape retrieval community [LZQ06, LGW08, OOFB08, OF08, FO09, TCF09]. The idea behind this approach is inspired by the common Bag-of-Words approach which is used for text retrieval and classification. First, a codebook of local features is selected with respect to a set of training objects. New objects are then characterized by describing their local feature occurencies with respect to the before established codebook. By that, local features are mapped into a single histogram, allowing for easy comparison of two 3D objects. As mentioned before, BoF based approaches lack the abilitiy to represent the spatial relationship of local features. In [LGW08], Li et al. try to alleviate this shortcoming by additionally taking the distance between the object center and the local feature into account. However, the exact spatial relationship between tuples of features cannot be represented appropriately by a BoF approach.

Meta descriptors The approach presented in [WBK08] is loosely related to BoF methods, as the idea is to map a set of local features into one single description. For each local feature it is first determined how characteristic it is for a set of certain object classes with respect to knowledge that was acquired in a training step. This information is aggregated in a discrete probability distribution (the meta descriptor). Finally, the probability distributions of all local features are combined into one single distribution allowing for easy object comparison. Like with the BoF methods, the spatial relationship of features is not taken into account. Using this method as a starting point for our own approach, we will show how to overcome this drawback in Section 4.

3. Feature Selection and Descriptor Computation

In this section, we will first describe how features of 3D models representing man-made objects can be selected using shape primitives like planes, cylinders, etc. After that we will show how the supporting regions of shape primitives can be represented by descriptors.

3.1. Feature Selection

As a starting point for the detection of primitive shapes we use an unstructured 3D point cloud which can be obtained by randomly sampling from a 3D mesh. We employ the algorithm presented in [SWK07] which recognizes planes, spheres, cylinders, cones and tori in the point cloud. In the evaluation conducted in [LSSK09], the segmentation provided by this approach showed increased robustness compared to the method presented in [WK05]. In contrast to [AFS06], the point clound-based approach does not require an intact mesh connectivity and it is not restricted to planes, cylinders, and spheres. In this section we will only give a very brief outline of the shape detection technique and the interested reader is referred to the original paper.

The data is decomposed into disjoint sets of points, each corresponding to a detected shape proxy respectively, and a set of remaining points that consists of outliers as well as areas of more complex geometry for which primitive shapes would give an inappropriate representation. For further processing, all remaining points are ignored. Points that are represented by a shape primitive are also called the *support* of a shape. Thus, given the point-cloud $P = p_1, \ldots, p_N$, the output of the shape detection is the following:

$$P = S_{\phi_1} \cup \ldots \cup S_{\phi_A} \cup R, \qquad (1)$$

where each subset (the support) S_{ϕ_i} is associated with a shape primitive ϕ_i. All points in S_{ϕ_i} constitute a connected component and fulfill the condition

$$s \in S_{\phi_i} \Rightarrow d(s,\phi_i) < \varepsilon \wedge \angle(n_s, n(\phi_i, s)) < \alpha, \qquad (2)$$

where n_s is the normal of point s, $n(\phi_i, s)$ denotes the normal of the primitive ϕ_i at the point closest to s and $d(s,\phi_i)$ denotes the Euclidean distance between s and ϕ_i. The normals n_s are thereby estimated on the point cloud. The parameters ε and α are chosen by the user according to the sampling distance. The set R contains all remaining, unassigned points.

Examples for the decomposition of several objects from the 3D Architecture Shape Benchmark [WBK09] can be found in Figure 1. For the choice of parameters concerning the primitive shape detection, we refer to Section 5.

3.2. Descriptor Computation

Theoretically, it would be possible to use the primitive shape type together with certain properties (e.g. radius and height for a cylinder primitive) as a shape descriptor. However, there are two reasons rendering this approach inefficient. First, the primitive shape detection is not robust with respect to the type of the detected primitive. For example, a set of points originating from a pipe might either be identified as part of a cylinder primitive or as part of a torus primitive with a very large radius (see e.g. the legs of the bench in Figure 1f). Second, such a descriptor would not incorporate the fact that the underlying support points might only represent a part of a primitive (e.g. only a hemisphere instead of a whole sphere). We therefore do not characterize the local object part by the primitive itsself but rather by its support. Once the primitive shape is detected, we compute a spin image [Joh97] representing the support points. By that, the discrete shape type is described in a more continuous way.

We align the spin image axis according to the Z-axis of the underlying object. Note that this representation is not only invariant under rotations around the Z-axis of the object. However, 3D models representing man-made objects are mostly modeled in a way that their Z-axis is chosen according to the world's up-direction. Therefore, this choice does not put a severe restriction to our algorithm, see e.g. the models in the 3D Architecture Shape Benchmark [WBK09] that we use for evaluation. Note that our framework for learning the compositional structure of 3D objects (see Section 4) is not restricted to the usage of spin images. It would also be possible to use e.g. Spherical Harmonics descriptors.

4. Learning the Compositional Structure of 3D Objects

As the approach presented in [WBK08] serves as a starting point for our method, we will first briefly explain it and will then present our extension towards the learning of compositional 3D object structures. The method for similarity computation between two sets of local features presented in [WBK08] relies on a supervised learning approach. The idea is to transform an arbitrary descriptor $d_i \in \mathbb{R}^k$ into a *class distribution descriptor (CDD)* $cdd(d_i)$ that states how characterstic d_i is for a set of certain object categories $\mathcal{C} = \{c_1, ..., c_n\}$. In terms of conditional class probabilities, this meta descriptor reads:

$$cdd(d_i) = \begin{pmatrix} p(c_1|d_i) \\ \vdots \\ p(c_n|d_i) \end{pmatrix}. \qquad (3)$$

The supervised learning approach consists of two steps. In the first step, conditional class probabilities are learned using nonlinear kernel discriminant analysis (NKDA) [RT01] with respect to a set of preclassified training features derived from 3D objects. In the second step, the acquired knowledge is used to predict conditional class probabilities for new local features. By that, for an object consisting of l local features $d_1, ..., d_l$, a set of l CDDs $cdd(d_1), ..., cdd(d_l)$ is computed. As these descriptors only contain conditional class probabilities which are uncoupled from the underlying geometric descriptor, similarity computations between two sets of local features can be easily realized. Adopting the Bayesian point of view, the according CDDs can be combined using the product rule:

$$\begin{aligned} cdd(d_1, ..., d_l) &= \begin{pmatrix} p(c_1|d_1, ..., d_l) \\ \vdots \\ p(c_n|d_1, ..., d_l) \end{pmatrix} \\ &= \frac{\prod_{i=1}^{l} cdd(d_i)}{\sum_{j=1}^{n} \prod_{i=1}^{l} cdd(d_i)}, \end{aligned} \qquad (4)$$

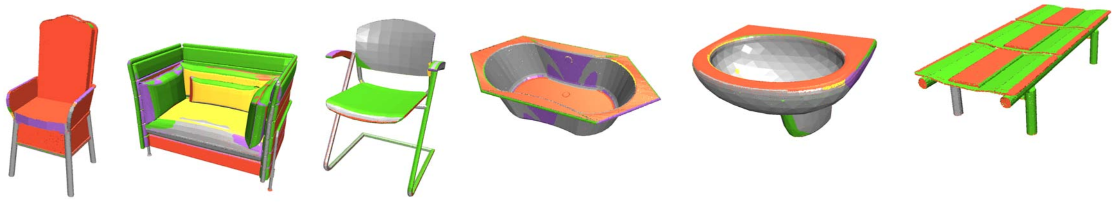
where $\prod$ denotes a pointwise product. Now consider two 3D objects o_1 and o_2 as well as a distance measure Δ. Object similarity $S(o_1, o_2)$ can then be written in terms of distance between according CDDs:

$$S(o_1, o_2) = \Delta(cdd(d_1^{o1}, ..., d_{l1}^{o1}), cdd(d_1^{o2}, ..., d_{l2}^{o2})). \qquad (5)$$

For further insights into how the conditional class probabilities are exactly computed, we refer to the original paper by Wessel et al. [WBK08]. Although this approach allows for incorporation of learned knowledge about object categories and avoids the cumbersome process of establishing feature correspondences, it does not take spatial relationships between features into account.

4.1. Integrating Feature Locations

In a first step, we add the relative position of single local features d_i with respect to the center of gravity m_o of the underlying object. Let now $\Phi(m_o, d_i)$ denote the spatial relationship between the feature d_i and the object center m_o.There are several possibilities how to choose $\Phi(m_o, d_i)$. In a setting where the underlying object can be rotated in an arbitrary way, the natural choice for $\Phi(m_o, d_i)$ would be $\Phi(m_o, d_i) := ||m_o - m_{di}||$, where m_{di} is the center of gravity of the support points of feature d_i. However, as in



(a) armchair (b) easy chair (c) side chair (d) bathtub (e) sink (f) bench

Figure 1: Detection of primitive shapes (for better understanding please see the color pages). Colors are chosen with respect to the partially detected primitive types plane (red), cylinder (green), torus (grey), cone (purple), and sphere (yellow). Figure f) shows an example for the instability of the shape detection. Two legs are identified as a cylinder, but one is identified as part of a torus. All examples are taken from the 3D Architecture Shape Benchmark.

our setting the Z-axes of the objects are consistently oriented, we follow another approach allowing us to integrate more precise information about the spatial relationship. Setting $\Phi(m_o, d_i) := (\delta_z(m_o, m_{di}), \delta_r(m_o, m_{di}))$, we take into account the signed distance $\delta_z(m_o, d_i) =: \delta_{zi}$ along the Z-axis as well as the unsigned distance $\delta_r(m_o, m_{di}) =: \delta_{ri}$ measured in the plane perpendicular to the Z-axis.

So far, the size of a local feature is not incorporated. We therefore introduce an additional parameter γ_i describing the number of support points of feature d_i with respect to the total number of points in the object. By that, the modified CDD reads:

$$cdd(d_i) = \begin{pmatrix} p(c_1|d_i, \delta_{zi}, \delta_{ri}, \gamma_i) \\ \vdots \\ p(c_n|d_i, \delta_{zi}, \delta_{ri}, \gamma_i) \end{pmatrix}. \qquad (6)$$

4.2. Spatial Relationship between Features

In the second step, we will additionally consider the spatial relationship between feature tuples d_{ij} consisting of two features d_i and d_j from the same object. As the positions of m_{di}, m_{dj} around the object center m_o are fixed by δ_z and δ_r except for rotation, we only need to additionally incorporate the distance $\delta_p(m_{di}, m_{dj}) =: \delta_{pij}$ between d_i and d_j into our framework. The according CDD is then given by:

$$cdd(d_{ij}) = \begin{pmatrix} p(c_1|d_{ij}, \delta_{zi}, \delta_{ri}, \gamma_i, \delta_{zj}, \delta_{rj}, \gamma_j, \delta_{pij}) \\ \vdots \\ p(c_n|d_{ij}, \delta_{zi}, \delta_{ri}, \gamma_i, \delta_{zj}, \delta_{rj}, \gamma_j, \delta_{pij}) \end{pmatrix}. \quad (7)$$

Intuitively speaking, it describes how likely it is that the currently considered object belongs to a certain object category, given the co-occurrence of features d_i and d_j in a certain spatial arrangement.

4.3. NKDA Kernel Function

The kernel function determines how similarity between features is computed during training as well as during the process of predicting the conditional probabilities for the CDD. A common choice for vector-valued features is a Gaussian kernel, which is also used in [WBK08]:

$$k(d_i, d_j) = \exp\left(-\frac{|d_i - d_j|^2}{2\sigma^2}\right), \qquad (8)$$

where σ denotes the kernel width. As described in Section 3.2 we use spin image coefficients $d_i = (d_{i1}, ..., d_{ik})$ as a descriptor for the extracted shape primitives. Considering single features, this descriptor has to be combined with the additional information about feature location and size given by the parameters δ_{zi},δ_{ri}, and γ_i. Note that simply defining

$$d_i' = (d_{i1}, ..., d_{ik}, \delta_{zi}, \delta_{ri}, \gamma_i)^t \qquad (9)$$

and evaluating $k(d_i', d_j')$ would lead to instabilities as the coefficients for spatial relationship and relative feature size have a completely different meaning and scale compared to the spin image coefficients. Although kernel-based discriminant functions are known to be able to implicitely weight certain feature entries, stability can be increased by introducing weighting factors when considering feature entries that are measured on different scales. Therefore, we modify the kernel function by introducing weights to properly balance all coefficients in d_i'. For single features, we define

$$k_s(d_i', d_j') := \exp\left(-\frac{(d_i' - d_j')^t W_s (d_i' - d_j')}{2\sigma^2}\right), \qquad (10)$$

where W_s is a diagonal matrix of size $(k+3) \times (k+3)$ containing weighting factors for δ_{zi},δ_{ri}, and γ_i. The diagonal $D(W_s)$ reads:

$$D(W_s) := (1, \cdots, 1, \alpha_\delta, \alpha_\delta, \alpha_\gamma). \qquad (11)$$

Considering a feature pair d_{ij}, the underlying shape descriptors d_i and d_j must be combined into one common vector. In this vector, we order d_i and d_j according to the size of their point support. Let d_i be the local feature with the larger point support. Then, incorporation of the additional information about feature location, size and spatial relationship leads to the following kernel input vector d'_{ij}:

$$\begin{aligned} d'_{ij} \;=\; & (d_{i1},\ldots,d_{ik},d_{j1},\ldots,d_{jk}, && (12)\\ & \delta_{zi},\delta_{ri},\gamma_i,\delta_{zj},\delta_{rj},\gamma_j,\delta_{pij})^t && (13) \end{aligned}$$

The according kernel function reads

$$k_t(d'_{ij},d'_{qr}) := \exp\left(-\frac{(d'_{ij}-d'_{qr})^t W_t (d'_{ij}-d'_{qr})}{2\sigma^2}\right), \quad (14)$$

where W_t is a $(2k+7)\times(2k+7)$ diagonal matrix containing the weighting factors such that the diagonal $D(W_t)$ reads:

$$D(W_t) := (1,\cdots,1,\alpha_\delta,\alpha_\delta,\alpha_\gamma,\alpha_\delta,\alpha_\delta,\alpha_\gamma,\alpha_\delta). \quad (15)$$

We determine all weighting factors as well as the kernel width σ completely automatically using crossvalidation. During the training process, a discriminant function for each pair of the n object classes is computed. This leads to different α_s and σ for each discriminant function, taking into account that feature size and relative position are of varying importance depending on the considered object categories.

4.4. Combining CDDs

In order to finally compare the CDDs derived from different objects, the CDDs of each single feature and of each feature pair must be combined into one single CDD. Following the combination technique presented in Equation 4, this descriptor can be determined by multiplying and renormalizing the CDDs computed so far:

$$cdd(\{d_i\},\{d_{ij}\}) \;=\; \frac{\prod_{i=1}^{l} cdd(d_i)\prod_i\prod_{j\neq i} cdd(d_{ij})}{\sum_{c=1}^{n}\prod_{i=1}^{l} cdd(d_i)\prod_i\prod_{j\neq i} cdd(d_{ij})},$$

where $\{d_i\},\{d_{ij}\}$ denote the sets of single features and feature pairs, respectively. We can now compute the similarity S between two objects o_1 and o_2 according to the underlying CDDs by evaluating

$$S(o_1,o_2) = \Delta(cdd_{o1}(\{d_i\},\{d_{ij}\}),cdd_{o2}(\{d_i\},\{d_{ij}\})$$

with respect to a similarity measure Δ.

5. Results

For our experiments, we use the 3D Architecture Shape Benchmark [WBK09] which contains 2257 models of man-made objects from the architectural domain like building elements or furnishing. In the work introducing this benchmark, several shape retrieval methods were tested on this new dataset. The results were compared to those achieved on the Princeton Shape Benchmark (PSB) [SMKF04] using the exact same methods. The best performing method was the approach presented in [WBK08] which we briefly described in the beginning of Section 4. As descriptors, 64 local spin images centered at randomly selected uniformly distributed points were used. We compare our own approach to this method in terms of retrieval performance.

5.1. Experimental Setup

Dataset The original 3D Architecture Shape Benchmark consists of 2257 models arranged in 180 and 183 classes, respectively. To ensure appropriate generalization performance of our supervised learning framework, we use a subset of this benchmark, selecting all classes that contain at least 20 objects.We divide the resulting 1817 objects belonging to 25 classes into a training set and a test set. For the training set, we randomly select 16 objects of each class.The remaining 1417 objects are put into the test set.

Preprocessing and Shape Detection A point cloud representation is the prerequisite for computing shape primitive features as well as spin image descriptors. We therefore normalize all meshes to the $[-1,-1,-1]\times[1,1,1]$ bounding box and randomly sample 50000 points per unit area on the surface from the underlying triangles. For the shape detection described in Section 3.1, we set $\alpha = 0.9$ and $\varepsilon = 0.002$. Note that the same parameter setting is used for the whole dataset. Depending on the complexity of the underlying model, the number of detected shapes varies between 10 to 200. For further descriptor computation, we select those 32 shapes providing the largest point support. If less than 32 shapes are detected, all of them are used.

Descriptors To evaluate our approach, we compute spin image descriptors describing the point support of every selected shape primitive. The spin images are positioned at the center of gravity of the support points and oriented according to the Z-axis of the object. The radius is chosen with respect to the support point farthest from the center of gravity. For the comparison to the approach presented in [WBK08], we randomly select 64 uniformly distributed surface points as spin image centers. In this setting, spin images are oriented according to the surface normal. For both settings, the spin image resolution is set to 16 × 16 bins.

Feature Tuples For an object with n detected shape features, we select those $n/2$ features providing the largest support to generate $\binom{n/2}{2}$ tuples. If the maximum number of 32 is detected, this leads to 120 feature tuples.

5.2. Evaluation

As a distance measure Δ (see Section 4.4) between CDDs we use the χ^2 metric. The performance of our algorithm is shown in Figure 2 and in Table 1. As can be seen in the precision-recall plot, our method *(Shapes and Spatial Relationships)* outperforms the approach based on spin images

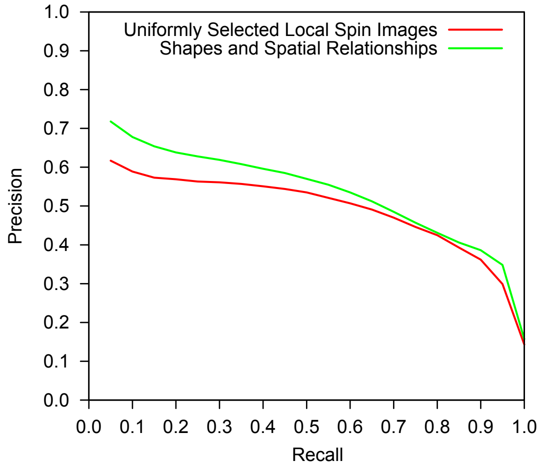


Figure 2

Method	1-NN	1-Tier	2-Tier	DCG
Uniformly Selected Local SI	0.642	0.486	0.676	0.782
Shapes + SR	0.748	0.531	0.683	0.809

Table 1: Comparison of our approaches to random feature selection. Our algorithm including shapes and spatial relationships shows superior quality to random feature selection.

centered at randomly selected uniformly distributed surface points *(Uniformly Selected Local Spin Images)*. Table 1 shows the performance of both 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, $1-Tier = n-1$ and $2-Tier = 2(n-1)$. Average discounted cumulative gain (DCG) gives an impression of how the overall retrieval would be viewed by a human. As can be seen, our proposed method involving spatial relationships and feature selection according to shape primitives achieves a higher retrieval performance.

Regarding the overall performance of both approaches, the Architecture Benchmark still remains a hard task for shape retrieval. In contrast to the PSB, on which the approach presented in [WBK08] showed very encouraging results (see [WBK09]), the Architecture Benchmark is restricted to a single object domain and the classification schemes provide very fine granularity (e.g. there are 11 different classes for chairs in the form-based scheme). Both of these properties cause a smaller inter-class variation in this benchmark, rendering shape retrieval difficult.

	Uniformly Selected Local SI	Shapes and SR
Preprocessing	1 h 15 min	15 h
CDD computation	55 min	2 h 13 min
Training Shapes	44 min	9 h 24 min

Table 2: Timings. Preprocessing times are with respect to the whole dataset including 1817 object. CDD computation times are with respect to the test set including 1417 objects.

5.3. Timings

In Table 2, we provide information about the time consumption of our approach. All experiments were run on an Intel®Core$^{\text{TM}}$2 Quad with 2.33 GHz and 4 GB RAM. Shape detection, training and CDD computation were parallelized using OpenMP. Training and CDD computation were additionally accelerated using a NVIDIA® GeForce®8800. Preprocessing timings include point cloud generation, shape detection and spin image descriptor computation for the shape and spatial relationships based approach and point cloud generation and spin image descriptor computation for the random feature selection based approach, respectively. Training and CDD computation take longer if spatial relationships are taken into account which is due to two reasons: First, feature tuples lead to an additional amount of training vectors. Second, cross validation must be performed to determine the α_s weighting factors.

6. Conclusion and Future Work

In this work, we introduced a supervised learning framework for 3D models representing man-made objects. It consists of a feature selection technique relying on detecting the shape primitives plane, cylinder, torus, sphere, and cone. As such shape primitives are building blocks of many man-made objects, our selection method is not only based on geometric saliency like most other approaches, but it also reflects the structural character of the underlying object domain. Additionally, our framework incorporates the spatial relationship of detected features. In our evaluation using the Architecture Benchmark, we show that our combination of supervised learning, feature selection and incorporation of spatial arrangement is superior to supervised learning and randomly selected uniformly distributed features. A drawback of our approach is the increased time consumption.

Future work should include the evaluation of other feature selection techniques which can be easily plugged into our framework. Considering the problems caused by the fine granularity of the benchmark, hierarchical approaches involving coarser classification schemes should be examined. When using supervised learning methods, classifiers could be specialized on certain hierarchy levels of the model taxonomy which might lead to improved retrieval performance.

7. Acknowledgements

We would like to thank Ruwen Schnabel for his help with the shape detection. We gratefully acknowledge support from the German Research Foundation DFG. The PROBADO project started in February 2006 with a tentative duration of five years. Partners are the University of Bonn, Technische Universität Darmstadt, Graz University of Technology, the German National Library of Science and Technology in Hannover, and the Bavarian State Library in Munich. For further information, please refer to the PROBADO website at http://www.probado.de. This work was partially supported under grant INST 3299/1-1.

References

[AFS06] ATTENE M., FALCIDIENO B., SPAGNUOLO M.: Hierarchical mesh segmentation based on fitting primitives. *Visual Computer 22*, 3 (2006), 181–193. 1, 2, 3

[FMA*09] FERREIRA A., MARINI S., ATTENE M., FONSECA M. J., SPAGNUOLO M., JORGE J. A., FALCIDIENO B.: Thesaurus-based 3d object retrieval with part-in-whole matching. *International Journal of Computer Vision* (June 2009). 1

[FO09] FURUYA T., OHBUCHI R.: Dense sampling and fast encoding for 3d model retrieval using bag-of-visual features. In *CIVR* (2009), pp. 1–8. 2, 3

[FS06] FUNKHOUSER T., SHILANE P.: Partial matching of 3d shapes with priority-driven search. In *Symposium on Geometry Processing* (2006), pp. 131–142. 2, 3

[GCO06] GAL R., COHEN-OR D.: Salient geometric features for partial shape matching and similarity. *ACM TOG 25*, 1 (2006), 130–150. 1, 2, 3

[HH09] HU J., HUA J.: Salient spectral geometric features for shape matching and retrieval. *The Visual Computer 25*, 5-7 (2009), 667–675. 1, 2

[JH99] JOHNSON A. E., HEBERT M.: Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Trans. Pattern Anal. Mach. Intell. 21*, 5 (1999), 433–449. 3

[Joh97] JOHNSON A.: *Spin-Images: A Representation for 3-D Surface Matching*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, August 1997. 2, 4

[KFR03] KAZHDAN M., FUNKHOUSER T., RUSINKIEWICZ S.: Rotation invariant spherical harmonic representation of 3d shape descriptors. In *SGP* (2003), pp. 156–164. 2

[LGW08] LI X., GODIL A., WAGAN A.: Spatially enhanced bags of words for 3d shape retrieval. In *ISVC* (2008), pp. 349–358. 2, 3

[Low04] LOWE D. G.: Distinctive image features from scale-invariant keypoints. *IJCV 60*, 2 (2004), 91–110. 2

[LSSK09] LI B., SCHNABEL R., SHIYAO J., KLEIN R.: Variational surface approximation and model selection. *Computer Graphics Forum 28*, 7 (oct 2009). 3

[LVJ05] LEE C. H., VARSHNEY A., JACOBS D. W.: Mesh saliency. *ACM TOG 24*, 3 (2005), 659–666. 2

[LZQ06] LIU Y., ZHA H., QIN H.: Shape topics: A compact representation and new algorithms for 3d partial shape retrieval. *CVPR 2* (2006), 2025–2032. 2, 3

[MGGP06] MITRA N. J., GUIBAS L., GIESEN J., PAULY M.: Probabilistic fingerprints for shapes. In *Symposium on Geometry Processing* (2006), pp. 121–130. 2

[NDK05] NOVOTNI M., DEGENER P., KLEIN R.: *Correspondence Generation and Matching of 3D Shape Subparts*. Tech. Rep. CG-2005-2, Universität Bonn, June 2005. 1, 2, 3

[Nov03] NOVOTNI M.: 3d zernike descriptors for content based shape retrieval. In *In The 8th ACM Symposium on Solid Modeling and Applications* (2003), ACM Press, pp. 216–225. 2

[OB07] OMMER B., BUHMANN J.: Learning the compositional nature of visual objects. In *CVPR* (June 2007), pp. 1–8. 2

[OB09] OMMER B., BUHMANN J. M.: Learning the compositional nature of visual object categories for recognition. *IEEE TPAMI 32* (2009), 501–516. 2

[OF08] OHBUCHI R., FURUYA T.: Accelerating bag-of-features sift algorithm for 3d model retrieval. In *Proceedings of the SAMT Workshop on Semantic 3D Media* (2008), pp. 28–30. 2, 3

[OOFB08] OHBUCHI R., OSADA K., FURUYA T., BANNO T.: Salient local visual features for shape-based 3d model retrieval. In *Shape Modeling International* (2008), pp. 93–102. 1, 2, 3

[RT01] ROTH V., TSUDA K.: Pairwise coupling for machine recognition of hand-printed japanese characters. In *CVPR* (2001), pp. I:1120–1125. 4

[SF06] SHILANE P., FUNKHOUSER T.: Selecting distinctive 3D shape descriptors for similarity retrieval. In *Shape Modeling International* (June 2006). 2, 3

[SF07] SHILANE P., FUNKHOUSER T.: Distinctive regions of 3D surfaces. *ACM TOG 26*, 2 (June 2007), Article 7. 3

[SMKF04] SHILANE P., MIN P., KAZHDAN M., FUNKHOUSER T.: The princeton shape benchmark. In *Shape Modeling International* (June 2004). 6

[SMS*04] SHAN Y., MATEI B., SAWHNEY H. S., KUMAR R., HUBER D., HEBERT M.: Linear model hashing and batch ransac for rapid and accurate object recognition. In *CVPR* (2004). 2, 3

[SSSCO08] SHALOM S., SHAPIRA L., SHAMIR A., COHEN-OR D.: Part analogies in sets of objects. In *3DOR* (2008), Eurographics Association, pp. 33–40. 2

[SWK07] SCHNABEL R., WAHL R., KLEIN R.: Efficient ransac for point-cloud shape detection. *Computer Graphics Forum 26*, 2 (June 2007), 214–226. 1, 2, 3

[SWWK08] SCHNABEL R., WESSEL R., WAHL R., KLEIN R.: Shape recognition in 3d point-clouds. In *WSCG* (Feb. 2008), Skala V., (Ed.), UNION Agency-Science Press. 2

[TCF09] TOLDO R., CASTELLANI U., FUSIELLO A.: Visual vocabulary signature for 3d object retrieval and partial matching. In *3DOR* (2009), pp. 21–28. 2, 3

[VKH06] VERMA V., KUMAR R., HSU S.: 3d building detection and modeling from aerial lidar data. In *CVPR* (2006), pp. 2213–2220. 2

[WBK08] WESSEL R., BARANOWSKI R., KLEIN R.: Learning distinctive local object characteristics for 3d shape retrieval. In *VMV* (2008), pp. 167–178. 2, 3, 4, 5, 6, 7

[WBK09] WESSEL R., BLÜMEL I., KLEIN R.: A 3d shape benchmark for retrieval and automatic classification of architectural data. In *3DOR* (Mar. 2009), pp. 53–56. 2, 4, 6, 7

[WK05] WU J., KOBBELT L.: Structure recovery via hybrid variational surface approximation. *Computer Graphics Forum 24*, 3 (2005), 277–284. 3

[WL88] WOLFSON H., LAMDAN Y.: Geometric hashing: A general and efficient model-based recognition scheme. In *ICCV88* (1988), pp. 238–249. 3

[ZHDQ08] ZOU G., HUA J., DONG M., QIN H.: Surface matching with salient keypoints in geodesic scale space. *Computer Animation and Virtual Worlds 19*, 3-4 (2008), 399–410. 2